

AC784xx_DFP HSM

11.1.0

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	6
2.1	File List	6
3	Class Documentation	8
3.1	aead_testvec Struct Reference	8
3.1.1	Detailed Description	8
3.1.2	Member Data Documentation	8
3.1.2.1	alen	8
3.1.2.2	assoc	9
3.1.2.3	clen	9
3.1.2.4	crypt_error	9
3.1.2.5	ctxt	9
3.1.2.6	iv	9
3.1.2.7	ivlen	9
3.1.2.8	key	10
3.1.2.9	klen	10
3.1.2.10	novrfy	10
3.1.2.11	plen	10
3.1.2.12	ptext	10
3.1.2.13	setauthsize_error	10
3.1.2.14	setkey_error	11
3.1.2.15	wk	11

3.2	akcipher_testvec Struct Reference	11
3.2.1	Detailed Description	11
3.2.2	Member Data Documentation	12
3.2.2.1	c	12
3.2.2.2	c_size	12
3.2.2.3	hash_alg	12
3.2.2.4	key	12
3.2.2.5	key_len	12
3.2.2.6	m	12
3.2.2.7	m_size	13
3.2.2.8	param_len	13
3.2.2.9	params	13
3.2.2.10	public_key_vec	13
3.2.2.11	siggen_sigver_test	13
3.3	bitmap Struct Reference	13
3.3.1	Detailed Description	14
3.3.2	Member Data Documentation	14
3.3.2.1	bit_bytes	14
3.3.2.2	bit_map	14
3.3.2.3	bit_max	14
3.4	cipher_session_st Struct Reference	14
3.4.1	Detailed Description	15
3.4.2	Member Data Documentation	15
3.4.2.1	aead_data	15
3.4.2.2	algorithm	15
3.4.2.3	cipher_mode	16
3.4.2.4	cmd_id	16
3.4.2.5	ctx	16
3.4.2.6	ctx_block_mgr	16
3.4.2.7	direction	16
3.4.2.8	is_hmac	16

3.4.2.9	job_id	17
3.4.2.10	key_auth_addr	17
3.4.2.11	key_auth_size	17
3.4.2.12	key_handle	17
3.4.2.13	key_type	17
3.4.2.14	mac_bytes	17
3.4.2.15	padding	18
3.4.2.16	rsa_crt_mode	18
3.4.2.17	signature_addr	18
3.4.2.18	signature_size	18
3.4.2.19	status	18
3.4.2.20	time_stamp	18
3.4.2.21	utc_time	19
3.5	cipher_testvec Struct Reference	19
3.5.1	Detailed Description	19
3.5.2	Member Data Documentation	19
3.5.2.1	clen	19
3.5.2.2	crypt_error	20
3.5.2.3	ctxt	20
3.5.2.4	fips_skip	20
3.5.2.5	iv	20
3.5.2.6	iv_out	20
3.5.2.7	key	20
3.5.2.8	klen	21
3.5.2.9	len	21
3.5.2.10	ptext	21
3.5.2.11	setkey_error	21
3.5.2.12	wk	21
3.6	counter_value_64_t Struct Reference	21
3.6.1	Detailed Description	22
3.6.2	Member Data Documentation	22

3.6.2.1	high_word	22
3.6.2.2	low_word	22
3.7	Crypto_AlgorithmInfoType Struct Reference	22
3.7.1	Detailed Description	22
3.7.2	Member Data Documentation	23
3.7.2.1	family	23
3.7.2.2	keyLength	23
3.7.2.3	mode	23
3.7.2.4	secondaryFamily	23
3.8	crypto_copy_key_dh_key_info Struct Reference	23
3.8.1	Detailed Description	24
3.8.2	Member Data Documentation	24
3.8.2.1	g	24
3.8.2.2	g_size	24
3.8.2.3	p	24
3.8.2.4	p_size	24
3.8.2.5	q	24
3.8.2.6	q_size	25
3.9	crypto_copy_key_info Struct Reference	25
3.9.1	Detailed Description	25
3.9.2	Member Data Documentation	25
3.9.2.1	auth_size	25
3.9.2.2	auth_value	25
3.9.2.3	element_data	26
3.9.2.4	element_size	26
3.9.2.5	key_handle	26
3.10	crypto_create_evita_key_info Struct Reference	26
3.10.1	Detailed Description	26
3.10.2	Member Data Documentation	26
3.10.2.1	element_data	27
3.10.2.2	element_size	27

3.10.2.3	key_size	27
3.10.2.4	type	27
3.10.2.5	valid_until	27
3.11	crypto_evita_key_info Struct Reference	27
3.11.1	Detailed Description	28
3.11.2	Member Data Documentation	28
3.11.2.1	auth_size	28
3.11.2.2	auth_value	28
3.11.2.3	key_handle	28
3.12	crypto_exported_key Struct Reference	28
3.12.1	Detailed Description	29
3.12.2	Member Data Documentation	29
3.12.2.1	encrypted_key	29
3.12.2.2	encrypted_key_buffer_size	29
3.12.2.3	encrypted_key_size	29
3.12.2.4	key_auth_code	29
3.12.2.5	key_auth_code_buffer_size	30
3.12.2.6	key_auth_code_size	30
3.13	crypto_import_evita_key_info Struct Reference	30
3.13.1	Detailed Description	30
3.13.2	Member Data Documentation	30
3.13.2.1	authenticity_key_authorization	31
3.13.2.2	authenticity_key_authorization_size	31
3.13.2.3	authenticity_key_handle	31
3.13.2.4	encrypted_key	31
3.13.2.5	encrypted_key_size	31
3.13.2.6	key_authenticity_code	31
3.13.2.7	key_authenticity_code_size	32
3.13.2.8	transport_key_authorization	32
3.13.2.9	transport_key_authorization_size	32
3.13.2.10	transport_key_handle	32

3.13.2.11 type	32
3.14 Crypto_JobInfoType Struct Reference	32
3.14.1 Detailed Description	33
3.14.2 Member Data Documentation	33
3.14.2.1 jobId	33
3.14.2.2 jobPriority	33
3.15 Crypto_JobPrimitiveInfoType Struct Reference	33
3.15.1 Detailed Description	33
3.15.2 Member Data Documentation	34
3.15.2.1 callbackId	34
3.15.2.2 callbackUpdateNotification	34
3.15.2.3 crylfKeyId	34
3.15.2.4 primitiveInfo	34
3.15.2.5 processingType	34
3.16 Crypto_JobPrimitiveInputOutputType Struct Reference	34
3.16.1 Detailed Description	35
3.16.2 Member Data Documentation	35
3.16.2.1 crylfKeyId	35
3.16.2.2 input64	35
3.16.2.3 inputLength	35
3.16.2.4 inputPtr	36
3.16.2.5 mode	36
3.16.2.6 output64Ptr	36
3.16.2.7 outputLengthPtr	36
3.16.2.8 outputPtr	36
3.16.2.9 secondaryInputLength	36
3.16.2.10 secondaryInputPtr	37
3.16.2.11 secondaryOutputLengthPtr	37
3.16.2.12 secondaryOutputPtr	37
3.16.2.13 targetCrylfKeyId	37
3.16.2.14 tertiaryInputLength	37

3.16.2.15 tertiaryInputPtr	37
3.16.2.16 verifyPtr	38
3.17 Crypto_JobRedirectionInfoType Struct Reference	38
3.17.1 Detailed Description	38
3.17.2 Member Data Documentation	38
3.17.2.1 inputKeyElementId	38
3.17.2.2 inputKeyId	39
3.17.2.3 outputKeyElementId	39
3.17.2.4 outputKeyId	39
3.17.2.5 redirectionConfig	39
3.17.2.6 secondaryInputKeyElementId	39
3.17.2.7 secondaryInputKeyId	39
3.17.2.8 secondaryOutputKeyElementId	40
3.17.2.9 secondaryOutputKeyId	40
3.17.2.10 tertiaryInputKeyElementId	40
3.17.2.11 tertiaryInputKeyId	40
3.18 Crypto_JobType Struct Reference	40
3.18.1 Detailed Description	41
3.18.2 Member Data Documentation	41
3.18.2.1 cryptoKeyId	41
3.18.2.2 jobId	41
3.18.2.3 jobInfo	41
3.18.2.4 jobPrimitiveInfo	41
3.18.2.5 jobPrimitiveInputOutput	41
3.18.2.6 jobRedirectionInfoRef	42
3.18.2.7 jobState	42
3.19 crypto_key_derive_info Struct Reference	42
3.19.1 Detailed Description	42
3.19.2 Member Data Documentation	42
3.19.2.1 derive_type	42
3.19.2.2 itera_times	43

3.19.2.3	key_info	43
3.19.2.4	passwd	43
3.19.2.5	passwd_size	43
3.20	crypto_key_element_type_info_st Struct Reference	43
3.20.1	Detailed Description	44
3.20.2	Member Data Documentation	44
3.20.2.1	CryptoKeyElementAllowPartialAccess	44
3.20.2.2	CryptoKeyElementId	44
3.20.2.3	CryptoKeyElementMaxSize	44
3.20.2.4	CryptoKeyElementPersist	44
3.20.2.5	CryptoKeyElementReadAccess	44
3.20.2.6	CryptoKeyElementWriteAccess	45
3.20.2.7	CryptoKeyFormat	45
3.21	crypto_key_export_info Struct Reference	45
3.21.1	Detailed Description	45
3.21.2	Member Data Documentation	45
3.21.2.1	authenticity_key_authorization	45
3.21.2.2	authenticity_key_authorization_size	46
3.21.2.3	authenticity_key_handle	46
3.21.2.4	key_handle	46
3.21.2.5	transport_key_authorization	46
3.21.2.6	transport_key_authorization_size	46
3.21.2.7	transport_key_handle	46
3.21.2.8	use_flags	47
3.22	crypto_key_status_info Struct Reference	47
3.22.1	Detailed Description	47
3.22.2	Member Data Documentation	47
3.22.2.1	certification_key_auth_size	47
3.22.2.2	certification_key_auth_value	47
3.22.2.3	certification_key_handle	48
3.22.2.4	key_handle	48

3.23 crypto_object Struct Reference	48
3.23.1 Detailed Description	48
3.23.2 Member Data Documentation	48
3.23.2.1 cmd_limit	48
3.23.2.2 cmd_list	49
3.23.2.3 cmd_sent	49
3.23.2.4 cmd_sent_num	49
3.23.2.5 name	49
3.23.2.6 queue_capacity	49
3.23.2.7 state	49
3.23.2.8 type	50
3.24 Crypto_PrimitiveInfoType Struct Reference	50
3.24.1 Detailed Description	50
3.24.2 Member Data Documentation	50
3.24.2.1 algorithm	50
3.24.2.2 resultLength	50
3.24.2.3 service	51
3.25 crypto_she_key Struct Reference	51
3.25.1 Detailed Description	51
3.25.2 Member Data Documentation	51
3.25.2.1 m1	51
3.25.2.2 m2	51
3.25.2.3 m3	52
3.25.2.4 m4	52
3.25.2.5 m5	52
3.25.2.6 she_ext_flag	52
3.26 CryptoDriverObject Struct Reference	52
3.26.1 Detailed Description	52
3.26.2 Member Data Documentation	53
3.26.2.1 CryptoDriverObjectEcucPartitionRef	53
3.26.2.2 CryptoDriverObjectId	53

3.26.2.3	CryptoPrimitiveNum	53
3.26.2.4	CryptoPrimitiveRef	53
3.26.2.5	CryptoQueueSize	53
3.27	CryptoKey Struct Reference	53
3.27.1	Detailed Description	54
3.27.2	Member Data Documentation	54
3.27.2.1	KeyType	54
3.27.2.2	Typeld	54
3.28	CryptoKeyElementType Struct Reference	54
3.28.1	Detailed Description	55
3.28.2	Member Data Documentation	55
3.28.2.1	CryptoElementActualSize	55
3.28.2.2	CryptoElementArray	55
3.28.2.3	CryptoKeyElementAllowPartialAccess	55
3.28.2.4	CryptoKeyElementId	55
3.28.2.5	CryptoKeyElementMaxSize	55
3.28.2.6	CryptoKeyElementPersist	56
3.28.2.7	CryptoKeyElementReadAccess	56
3.28.2.8	CryptoKeyElementWriteAccess	56
3.28.2.9	CryptoKeyFormat	56
3.29	CryptoKeyType Struct Reference	56
3.29.1	Detailed Description	56
3.29.2	Member Data Documentation	57
3.29.2.1	element_info	57
3.29.2.2	keyelement_arr	57
3.29.2.3	keyelement_num	57
3.30	CryptoPrimitive Struct Reference	57
3.30.1	Detailed Description	57
3.30.2	Member Data Documentation	58
3.30.2.1	CryptoPrimitiveAlgorithmFamily	58
3.30.2.2	CryptoPrimitiveAlgorithmMode	58

3.30.2.3	CryptoPrimitiveAlgorithmSecondaryFamily	58
3.30.2.4	CryptoPrimitiveService	58
3.31	dlist_head Struct Reference	58
3.31.1	Detailed Description	59
3.31.2	Member Data Documentation	59
3.31.2.1	next	59
3.31.2.2	prev	59
3.32	ecies_testvec Struct Reference	59
3.32.1	Detailed Description	60
3.32.2	Member Data Documentation	60
3.32.2.1	cipher_part1	60
3.32.2.2	cipher_part2_part3_with_s1	60
3.32.2.3	cipher_part2_part3_with_s1_s2	60
3.32.2.4	cipher_part2_part3_with_s2	60
3.32.2.5	cipher_part2_part3_without_s1_s2	60
3.32.2.6	curve_id	61
3.32.2.7	kdf_hash_alg	61
3.32.2.8	mac_hash_alg	61
3.32.2.9	mac_k_bytes	61
3.32.2.10	msg	61
3.32.2.11	msg_bytes	61
3.32.2.12	point_form	62
3.32.2.13	receiver_pri_key	62
3.32.2.14	receiver_pri_key_sz	62
3.32.2.15	receiver_pub_key	62
3.32.2.16	receiver_pub_key_sz	62
3.32.2.17	sender_tmp_pri_key	62
3.32.2.18	sender_tmp_pri_key_sz	63
3.32.2.19	shared_info1	63
3.32.2.20	shared_info1_bytes	63
3.32.2.21	shared_info2	63

3.32.2.22 shared_info2_bytes	63
3.33 ehsm_aead_data_ptr Struct Reference	63
3.33.1 Detailed Description	64
3.33.2 Member Data Documentation	64
3.33.2.1 input_data	64
3.33.2.2 output_data	64
3.34 ehsm_certificate_verify_st Struct Reference	64
3.34.1 Detailed Description	64
3.34.2 Member Data Documentation	65
3.34.2.1 certificate_info	65
3.34.2.2 verify	65
3.35 ehsm_change_control_field_cmd Struct Reference	65
3.35.1 Detailed Description	65
3.35.2 Member Data Documentation	65
3.35.2.1 reserved	65
3.35.2.2 size	66
3.35.2.3 type	66
3.35.2.4 value_addr	66
3.36 ehsm_change_control_field_st Struct Reference	66
3.36.1 Detailed Description	66
3.36.2 Member Data Documentation	66
3.36.2.1 rev	67
3.36.2.2 size	67
3.36.2.3 type	67
3.36.2.4 value	67
3.37 ehsm_change_lifecycle_cmd Struct Reference	67
3.37.1 Detailed Description	67
3.37.2 Member Data Documentation	68
3.37.2.1 type	68
3.38 ehsm_close_debug_cmd Struct Reference	68
3.38.1 Detailed Description	68

3.38.2	Member Data Documentation	68
3.38.2.1	type	68
3.39	ehsm_cmd Struct Reference	68
3.39.1	Detailed Description	69
3.39.2	Member Data Documentation	69
3.39.2.1	output_size	69
3.39.2.2	req_cipher	69
3.40	ehsm_cmd_aead_ptr_st Struct Reference	69
3.40.1	Detailed Description	69
3.40.2	Member Data Documentation	70
3.40.2.1	aad_ptr	70
3.40.2.2	data_ptr	70
3.40.2.3	tag_ptr	70
3.41	ehsm_cmd_cipher_st Struct Reference	70
3.41.1	Detailed Description	71
3.41.2	Member Data Documentation	71
3.41.2.1	cmd_id	71
3.41.2.2	context_addr	71
3.41.2.3	context_size	71
3.41.2.4	input_addr	71
3.41.2.5	input_size	71
3.41.2.6	key_addr	72
3.41.2.7	key_handle	72
3.41.2.8	key_size	72
3.41.2.9	output_addr	72
3.41.2.10	output_size	72
3.41.2.11	rev1	72
3.41.2.12	sec_input_addr	73
3.41.2.13	sec_input_size	73
3.41.2.14	u_hdr	73
3.42	ehsm_cmd_hdr_eccp_keygen_st Struct Reference	73

3.42.1 Detailed Description	73
3.42.2 Member Data Documentation	73
3.42.2.1 curve_id	74
3.42.2.2 hdr_rev1	74
3.42.2.3 type	74
3.43 ehsm_cmd_hdr_ecise_st Struct Reference	74
3.43.1 Detailed Description	74
3.43.2 Member Data Documentation	75
3.43.2.1 cipher_alg	75
3.43.2.2 curve_id	75
3.43.2.3 direction	75
3.43.2.4 hdr_rev1	75
3.43.2.5 hdr_rev2	75
3.43.2.6 kdf_alg	75
3.43.2.7 mac_alg	76
3.43.2.8 mac_k_byte	76
3.44 ehsm_cmd_hdr_pke_st Struct Reference	76
3.44.1 Detailed Description	76
3.44.2 Member Data Documentation	76
3.44.2.1 algorithm	76
3.44.2.2 direction	77
3.44.2.3 hdr_rev1	77
3.44.2.4 padding	77
3.44.2.5 process_mode	77
3.44.2.6 rsa crt_mode	77
3.44.2.7 time_stamp	77
3.45 ehsm_cmd_hdr_rng_st Struct Reference	78
3.45.1 Detailed Description	78
3.45.2 Member Data Documentation	78
3.45.2.1 algorithm	78
3.45.2.2 hdr_rev1	78

3.45.2.3	hdr_rev2	78
3.46	ehsm_cmd_hdr_rsa_keygen_st Struct Reference	79
3.46.1	Detailed Description	79
3.46.2	Member Data Documentation	79
3.46.2.1	e_bit_size	79
3.46.2.2	hdr_rev1	79
3.46.2.3	is_crt	79
3.46.2.4	n_bit_size	80
3.46.2.5	type	80
3.47	ehsm_cmd_hdr_ske_st Struct Reference	80
3.47.1	Detailed Description	80
3.47.2	Member Data Documentation	80
3.47.2.1	algorithm	81
3.47.2.2	cipher_mode	81
3.47.2.3	direction	81
3.47.2.4	key_type	81
3.47.2.5	padding	81
3.47.2.6	process_mode	81
3.47.2.7	tag_size	82
3.47.2.8	time_stamp	82
3.48	ehsm_cmd_hdr_sm9_st Struct Reference	82
3.48.1	Detailed Description	82
3.48.2	Member Data Documentation	82
3.48.2.1	direction	82
3.48.2.2	enc_type	83
3.48.2.3	hdr_rev1	83
3.48.2.4	hdr_rev2	83
3.48.2.5	hid	83
3.48.2.6	key2_size	83
3.48.2.7	padding	83
3.49	ehsm_cmd_req Struct Reference	84

3.49.1 Detailed Description	84
3.49.2 Member Data Documentation	84
3.49.2.1 api_type	84
3.49.2.2 channel	84
3.49.2.3 cmd_data	85
3.49.2.4 cmd_id	85
3.49.2.5 cmd_size	85
3.49.2.6 cmd_state	85
3.49.2.7 error_code	85
3.49.2.8 list	85
3.49.2.9 object_type	86
3.49.2.10 priority	86
3.49.2.11 release_cb	86
3.49.2.12 req_cb	86
3.49.2.13 req_ctx	86
3.49.2.14 req_type	86
3.49.2.15 rps_data	87
3.49.2.16 timeout	87
3.50 ehsm_cmd_req_buffer Struct Reference	87
3.50.1 Detailed Description	87
3.50.2 Member Data Documentation	87
3.50.2.1 bitmap	87
3.50.2.2 cmd_req	87
3.51 ehsm_cmd_sm9_sig_vry_output_ptr_st Struct Reference	88
3.51.1 Detailed Description	88
3.51.2 Member Data Documentation	88
3.51.2.1 h	88
3.51.2.2 Sig	88
3.52 ehsm_copy_key_cmd Struct Reference	88
3.52.1 Detailed Description	89
3.52.2 Member Data Documentation	89

3.52.2.1	key_auth_size	89
3.52.2.2	key_auth_value	89
3.52.2.3	key_handle	89
3.52.2.4	key_usage	89
3.52.2.5	key_usage_size	90
3.52.2.6	reserved1	90
3.53	ehsm_create_dh_key_cmd Struct Reference	90
3.53.1	Detailed Description	90
3.53.2	Member Data Documentation	90
3.53.2.1	algorithm	91
3.53.2.2	dh_mode	91
3.53.2.3	key_size	91
3.53.2.4	key_usage	91
3.53.2.5	key_usage_size	91
3.53.2.6	local_key_auth_size	91
3.53.2.7	local_key_auth_value	92
3.53.2.8	local_key_handle	92
3.53.2.9	parent_alg	92
3.53.2.10	remote_key_auth_size	92
3.53.2.11	remote_key_auth_value	92
3.53.2.12	remote_key_handle	92
3.53.2.13	reserved2	93
3.53.2.14	reserved3	93
3.53.2.15	sm2_ext_param	93
3.53.2.16	ss_addr	93
3.53.2.17	type	93
3.53.2.18	valid_until	93
3.54	ehsm_create_dh_key_param Struct Reference	94
3.54.1	Detailed Description	94
3.54.2	Member Data Documentation	94
3.54.2.1	dh_mode	94

3.54.2.2	key_element_data	94
3.54.2.3	key_element_size	95
3.54.2.4	key_handle	95
3.54.2.5	key_size	95
3.54.2.6	local_key_auth_size	95
3.54.2.7	local_key_auth_value	95
3.54.2.8	local_key_handle	95
3.54.2.9	parent_alg	96
3.54.2.10	remote_key_auth_or_pub_key_size	96
3.54.2.11	remote_key_auth_value_or_pub_key	96
3.54.2.12	remote_key_handle	96
3.54.2.13	sm2_ext_para	96
3.54.2.14	ss_addr	96
3.54.2.15	target_algorithm_identifier	97
3.54.2.16	type	97
3.55	ehsm_create_dh_sm2_ext_param Struct Reference	97
3.55.1	Detailed Description	97
3.55.2	Member Data Documentation	97
3.55.2.1	local_tmp_key_auth_size	97
3.55.2.2	local_tmp_key_auth_value	98
3.55.2.3	local_tmp_key_handle	98
3.55.2.4	peer_temp_pubkey	98
3.55.2.5	reserved2	98
3.55.2.6	s1_s2_value	98
3.55.2.7	sa_sb_value	98
3.55.2.8	sm2_role	99
3.56	ehsm_create_evita_key_param Struct Reference	99
3.56.1	Detailed Description	99
3.56.2	Member Data Documentation	99
3.56.2.1	gen_key_param	99
3.56.2.2	key_element_data	99

3.56.2.3	key_element_size	100
3.56.2.4	key_handle	100
3.56.2.5	key_size	100
3.56.2.6	type	100
3.57	ehsm_create_random_key_param Struct Reference	100
3.57.1	Detailed Description	101
3.57.2	Member Data Documentation	101
3.57.2.1	g	101
3.57.2.2	g_size	101
3.57.2.3	key_element_data	101
3.57.2.4	key_element_size	101
3.57.2.5	key_handle	101
3.57.2.6	key_size	102
3.57.2.7	p	102
3.57.2.8	p_size	102
3.57.2.9	q	102
3.57.2.10	q_size	102
3.57.2.11	target_algorithm_identifier	102
3.57.2.12	type	103
3.58	ehsm_crypto_key Struct Reference	103
3.58.1	Detailed Description	103
3.58.2	Member Data Documentation	103
3.58.2.1	crypto_key	103
3.58.2.2	ehsm_key_type	103
3.58.2.3	valid	104
3.59	ehsm_crypto_randomgenerate_param Struct Reference	104
3.59.1	Detailed Description	104
3.59.2	Member Data Documentation	104
3.59.2.1	algorithm	104
3.59.2.2	random_data_addr	104
3.59.2.3	request_size	105

3.60 ehsm_ctx_block_mgr Struct Reference	105
3.60.1 Detailed Description	105
3.60.2 Member Data Documentation	105
3.60.2.1 block_buf	105
3.60.2.2 block_sz	105
3.60.2.3 remain_data_sz	106
3.61 ehsm_ctx_session_st Struct Reference	106
3.61.1 Detailed Description	106
3.61.2 Member Data Documentation	106
3.61.2.1 block_sz	106
3.61.2.2 max_chunk_size	106
3.61.2.3 session_id	107
3.62 ehsm_debug_auth_st Struct Reference	107
3.62.1 Detailed Description	107
3.62.2 Member Data Documentation	107
3.62.2.1 alg	107
3.62.2.2 public_key	107
3.62.2.3 public_key_size	108
3.62.2.4 signature	108
3.62.2.5 signature_size	108
3.62.2.6 type	108
3.63 ehsm_debug_authentication_cmd Struct Reference	108
3.63.1 Detailed Description	109
3.63.2 Member Data Documentation	109
3.63.2.1 algrithm	109
3.63.2.2 pub_addr	109
3.63.2.3 pub_size	109
3.63.2.4 rev1	109
3.63.2.5 rev2	109
3.63.2.6 sign_addr	110
3.63.2.7 sign_size	110

3.63.2.8	type	110
3.64	ehsm_derive_key_cmd Struct Reference	110
3.64.1	Detailed Description	111
3.64.2	Member Data Documentation	111
3.64.2.1	derive_type	111
3.64.2.2	itera_times	111
3.64.2.3	key_deriv_func	111
3.64.2.4	key_size	111
3.64.2.5	key_usage	111
3.64.2.6	key_usage_size	112
3.64.2.7	parent_key_auth_size	112
3.64.2.8	parent_key_auth_value	112
3.64.2.9	parent_key_handle	112
3.64.2.10	pw_data	112
3.64.2.11	pw_size	112
3.64.2.12	reserved1	113
3.64.2.13	reserved2	113
3.64.2.14	salt_data	113
3.64.2.15	salt_size	113
3.64.2.16	type	113
3.64.2.17	valid_until	113
3.65	ehsm_dh_param Struct Reference	114
3.65.1	Detailed Description	114
3.65.2	Member Data Documentation	114
3.65.2.1	g	114
3.65.2.2	p	114
3.65.2.3	q	114
3.66	ehsm_dh_param_size_info Struct Reference	114
3.66.1	Detailed Description	115
3.66.2	Member Data Documentation	115
3.66.2.1	g_size	115

3.66.2.2	p_size	115
3.66.2.3	q_size	115
3.67	ehsm_dh_prikey Struct Reference	115
3.67.1	Detailed Description	116
3.67.2	Member Data Documentation	116
3.67.2.1	priv	116
3.68	ehsm_dh_pubkey Struct Reference	116
3.68.1	Detailed Description	116
3.68.2	Member Data Documentation	116
3.68.2.1	dh_param	116
3.68.2.2	pub	117
3.69	ehsm_ecc_key_size_ Struct Reference	117
3.69.1	Detailed Description	117
3.69.2	Member Data Documentation	117
3.69.2.1	priv_key_size	117
3.69.2.2	pub_key_size	117
3.70	ehsm_ecc_pubkey Struct Reference	118
3.70.1	Detailed Description	118
3.70.2	Member Data Documentation	118
3.70.2.1	p	118
3.71	ehsm_emu_status_st_ Struct Reference	118
3.71.1	Detailed Description	118
3.71.2	Member Data Documentation	118
3.71.2.1	o_hsm_err_fw	119
3.71.2.2	o_hsm_err_hw	119
3.71.2.3	o_hsm_err_sensor	119
3.71.2.4	o_hsm_status	119
3.72	ehsm_evita_key_export Struct Reference	119
3.72.1	Detailed Description	120
3.72.2	Member Data Documentation	120
3.72.2.1	authenticity_key_author_size	120

3.72.2.2	authenticity_key_author_value	120
3.72.2.3	authenticity_key_handle	120
3.72.2.4	encrypted_key	120
3.72.2.5	encrypted_key_size	120
3.72.2.6	key_auth_code	121
3.72.2.7	key_auth_code_size	121
3.72.2.8	key_handle	121
3.72.2.9	transport_key_author_size	121
3.72.2.10	transport_key_author_value	121
3.72.2.11	transport_key_handle	121
3.72.2.12	use_flags	122
3.73	ehsm_evita_key_import_st Struct Reference	122
3.73.1	Detailed Description	122
3.73.2	Member Data Documentation	122
3.73.2.1	authenticity_key_author_size	122
3.73.2.2	authenticity_key_author_value	123
3.73.2.3	authenticity_key_handle	123
3.73.2.4	encrypted_key	123
3.73.2.5	encrypted_key_size	123
3.73.2.6	key_auth_code	123
3.73.2.7	key_auth_code_size	123
3.73.2.8	key_handle	124
3.73.2.9	transport_key_author_size	124
3.73.2.10	transport_key_author_value	124
3.73.2.11	transport_key_handle	124
3.73.2.12	type	124
3.74	ehsm_evita_memory_info_st Struct Reference	124
3.74.1	Detailed Description	125
3.74.2	Member Data Documentation	125
3.74.2.1	nvm_free_size	125
3.74.2.2	nvm_total_size	125

3.74.2.3	ram_free_size	125
3.74.2.4	ram_total_size	125
3.75	ehsm_exchange_sm9_key_param Struct Reference	126
3.75.1	Detailed Description	126
3.75.2	Member Data Documentation	126
3.75.2.1	fp12g	126
3.75.2.2	key_handle	126
3.75.2.3	key_size	127
3.75.2.4	kgc_pub_key	127
3.75.2.5	peer_id	127
3.75.2.6	peer_id_size	127
3.75.2.7	peer_tmp_pub	127
3.75.2.8	role	127
3.75.2.9	s1_s2	128
3.75.2.10	sa_sb	128
3.75.2.11	self_id	128
3.75.2.12	self_id_size	128
3.75.2.13	type	128
3.75.2.14	user_priv_key_handle	128
3.75.2.15	user_tmp_key_handle	129
3.76	ehsm_export_key_cmd Struct Reference	129
3.76.1	Detailed Description	129
3.76.2	Member Data Documentation	129
3.76.2.1	authenticity_key_auth_size	129
3.76.2.2	authenticity_key_auth_value	130
3.76.2.3	authenticity_key_handle	130
3.76.2.4	encrypted_key	130
3.76.2.5	encrypted_key_size	130
3.76.2.6	key_auth_size	130
3.76.2.7	key_auth_value	130
3.76.2.8	key_handle	131

3.76.2.9	transport_key_auth_size	131
3.76.2.10	transport_key_auth_value	131
3.76.2.11	transport_key_handle	131
3.76.2.12	use_flags	131
3.77	ehsm_export_pub_key_Struct Reference	131
3.77.1	Detailed Description	132
3.77.2	Member Data Documentation	132
3.77.2.1	algo_id	132
3.77.2.2	dh_pubkey_bytes_size	132
3.77.2.3	key	132
3.77.2.4	rsa_e_bytes_size	132
3.77.2.5	size_info	133
3.78	ehsm_external_key_Struct Reference	133
3.78.1	Detailed Description	133
3.78.2	Member Data Documentation	133
3.78.2.1	auth_sign_data	133
3.78.2.2	evita_internal_key	133
3.79	ehsm_fast_cmac_st Struct Reference	133
3.79.1	Detailed Description	134
3.79.2	Member Data Documentation	134
3.79.2.1	algorithm	134
3.79.2.2	direction	134
3.79.2.3	in	134
3.79.2.4	key_auth_size	135
3.79.2.5	key_auth_value	135
3.79.2.6	key_handle	135
3.79.2.7	key_type	135
3.79.2.8	mac	135
3.79.2.9	size	135
3.80	ehsm_fw_encrypt_key Struct Reference	136
3.80.1	Detailed Description	136

3.80.2	Member Data Documentation	136
3.80.2.1	key_data	136
3.80.2.2	key_size	136
3.80.2.3	key_slot	136
3.80.2.4	key_type	137
3.81	ehsm_fw_encrypt_key_cmd Struct Reference	137
3.81.1	Detailed Description	137
3.81.2	Member Data Documentation	137
3.81.2.1	input_addr	137
3.81.2.2	input_size	137
3.81.2.3	key_slot	138
3.81.2.4	key_type	138
3.81.2.5	rev1	138
3.81.2.6	rev2	138
3.81.2.7	rev3	138
3.82	ehsm_fw_get_random_key_cmd Struct Reference	138
3.82.1	Detailed Description	139
3.82.2	Member Data Documentation	139
3.82.2.1	key_slot	139
3.82.2.2	key_type	139
3.82.2.3	reserved1	139
3.83	ehsm_fw_random_key Struct Reference	139
3.83.1	Detailed Description	140
3.83.2	Member Data Documentation	140
3.83.2.1	key_slot	140
3.83.2.2	key_type	140
3.84	ehsm_gen_dh_key_param_st Struct Reference	140
3.84.1	Detailed Description	140
3.84.2	Member Data Documentation	140
3.84.2.1	g	141
3.84.2.2	g_size	141

3.84.2.3	p	141
3.84.2.4	p_size	141
3.84.2.5	q	141
3.84.2.6	q_size	141
3.85	ehsm_gen_key_cmd Struct Reference	142
3.85.1	Detailed Description	142
3.85.2	Member Data Documentation	142
3.85.2.1	alg_or_crt	142
3.85.2.2	e_size	142
3.85.2.3	g	143
3.85.2.4	g_size	143
3.85.2.5	key_usage	143
3.85.2.6	key_usage_size	143
3.85.2.7	n_size	143
3.85.2.8	p	143
3.85.2.9	p_size	144
3.85.2.10	q	144
3.85.2.11	q_size	144
3.85.2.12	reserved1	144
3.85.2.13	reserved2	144
3.85.2.14	reserved3	144
3.85.2.15	type	145
3.85.2.16	valid_until	145
3.86	ehsm_gen_key_param_st Struct Reference	145
3.86.1	Detailed Description	145
3.86.2	Member Data Documentation	145
3.86.2.1	algo_id	145
3.86.2.2	dh_key_param	146
3.86.2.3	param	146
3.86.2.4	rsa_e_bit_size	146
3.87	ehsm_gen_sm9_key_param Struct Reference	146

3.87.1 Detailed Description	146
3.87.2 Member Data Documentation	146
3.87.2.1 hid	147
3.87.2.2 key_handle	147
3.87.2.3 key_param	147
3.87.2.4 master_key	147
3.87.2.5 priv_key	147
3.87.2.6 rev	147
3.87.2.7 sm9_key_type	148
3.88 ehsm_gen_sm9_master_key_param Struct Reference	148
3.88.1 Detailed Description	148
3.88.2 Member Data Documentation	148
3.88.2.1 master_key_type	148
3.89 ehsm_gen_sm9_userpriv_key_cmd Struct Reference	148
3.89.1 Detailed Description	149
3.89.2 Member Data Documentation	149
3.89.2.1 id_addr	149
3.89.2.2 id_size	149
3.89.2.3 rev1	149
3.89.2.4 rev_key_auth	149
3.89.2.5 rev_key_auth_size	150
3.89.2.6 rev_key_handle	150
3.89.2.7 type	150
3.90 ehsm_gen_sm9_userpriv_key_param Struct Reference	150
3.90.1 Detailed Description	150
3.90.2 Member Data Documentation	150
3.90.2.1 kgc_pubkey	151
3.90.2.2 priv_key_type	151
3.90.2.3 type	151
3.90.2.4 user_id_size	151
3.90.2.5 user_id_value	151

3.90.2.6	with_pubkey	151
3.91	ehsm_get_challenge_cmd Struct Reference	152
3.91.1	Detailed Description	152
3.91.2	Member Data Documentation	152
3.91.2.1	output_addr	152
3.91.2.2	reserved1	152
3.91.2.3	type	152
3.92	ehsm_get_challenge_st Struct Reference	152
3.92.1	Detailed Description	153
3.92.2	Member Data Documentation	153
3.92.2.1	buf	153
3.92.2.2	size	153
3.92.2.3	type	153
3.93	ehsm_get_emu_cmd Struct Reference	153
3.93.1	Detailed Description	154
3.93.2	Member Data Documentation	154
3.93.2.1	emu_addr	154
3.93.2.2	emu_size	154
3.93.2.3	rev1	154
3.93.2.4	rev2	154
3.93.2.5	rev3	155
3.93.2.6	rev_key_auth_addr	155
3.93.2.7	rev_key_auth_size	155
3.93.2.8	rev_key_handle	155
3.94	ehsm_get_emu_status_param_st Struct Reference	155
3.94.1	Detailed Description	155
3.94.2	Member Data Documentation	156
3.94.2.1	emu_addr	156
3.94.2.2	emu_size	156
3.95	ehsm_get_pub_from_priv_cmd Struct Reference	156
3.95.1	Detailed Description	156

3.95.2	Member Data Documentation	156
3.95.2.1	key_alg_id	157
3.95.2.2	key_auth_size	157
3.95.2.3	key_auth_value	157
3.95.2.4	key_handle	157
3.95.2.5	public_key_addr	157
3.95.2.6	public_key_buffer_size	157
3.95.2.7	reserved1	158
3.95.2.8	reserved2	158
3.96	ehsm_get_pub_from_priv_param Struct Reference	158
3.96.1	Detailed Description	158
3.96.2	Member Data Documentation	158
3.96.2.1	key_alg_id	158
3.96.2.2	key_auth_size	159
3.96.2.3	key_auth_value	159
3.96.2.4	key_handle	159
3.96.2.5	public_key_addr	159
3.96.2.6	public_key_buffer_size	159
3.96.2.7	public_key_size	159
3.97	ehsm_get_she_id_cmd Struct Reference	160
3.97.1	Detailed Description	160
3.97.2	Member Data Documentation	160
3.97.2.1	challenge_addr	160
3.97.2.2	challenge_size	160
3.97.2.3	reserved	160
3.97.2.4	signatrue_addr	161
3.97.2.5	signatrue_size	161
3.97.2.6	status_addr	161
3.97.2.7	status_size	161
3.98	ehsm_image Struct Reference	161
3.98.1	Detailed Description	162

3.98.2	Member Data Documentation	162
3.98.2.1	ctx	162
3.98.2.2	ctx_size	162
3.98.2.3	image	162
3.98.2.4	image_size	162
3.98.2.5	process_mode	162
3.98.2.6	storage	163
3.98.2.7	storage_size	163
3.99	ehsm_image_upgrade_cmd Struct Reference	163
3.99.1	Detailed Description	163
3.99.2	Member Data Documentation	163
3.99.2.1	ctx_addr	164
3.99.2.2	ctx_size	164
3.99.2.3	image_addr	164
3.99.2.4	image_size	164
3.99.2.5	process_mode	164
3.99.2.6	rev1	164
3.99.2.7	rev2	165
3.99.2.8	rev3	165
3.99.2.9	rev4	165
3.99.2.10	rev_key_auth	165
3.99.2.11	rev_key_auth_size	165
3.99.2.12	rev_key_handle	165
3.99.2.13	storage_addr	166
3.100	ehsm_image_verfiy_cmd Struct Reference	166
3.100.1	Detailed Description	166
3.100.2	Member Data Documentation	166
3.100.2.1	ctx_addr	166
3.100.2.2	ctx_size	167
3.100.2.3	image_addr	167
3.100.2.4	image_size	167

3.100.2.5 process_mode	167
3.100.2.6 rev1	167
3.100.2.7 rev2	167
3.100.2.8 rev3	168
3.100.2.9 rev4	168
3.100.2.10rev5	168
3.100.2.11rev_key_auth	168
3.100.2.12rev_key_auth_size	168
3.100.2.13rev_key_handle	168
3.100.2.14type	169
3.101ehsm_image_verify_st Struct Reference	169
3.101.1 Detailed Description	169
3.101.2 Member Data Documentation	169
3.101.2.1 ctx	169
3.101.2.2 ctx_size	169
3.101.2.3 image	170
3.101.2.4 image_size	170
3.101.2.5 process_mode	170
3.101.2.6 storage	170
3.101.2.7 storage_size	170
3.101.2.8 type	170
3.102ehsm_import_key_cmd Struct Reference	171
3.102.1 Detailed Description	171
3.102.2 Member Data Documentation	171
3.102.2.1 authenticity_key_auth_size	171
3.102.2.2 authenticity_key_auth_value	171
3.102.2.3 authenticity_key_handle	171
3.102.2.4 encrypted_key	172
3.102.2.5 encrypted_key_size	172
3.102.2.6 key_auth_size	172
3.102.2.7 key_auth_value	172

3.102.2.8 key_type	172
3.102.2.9 rev1	172
3.102.2.10transport_key_auth_size	173
3.102.2.11transport_key_auth_value	173
3.102.2.12transport_key_handle	173
3.103ehsm_internal_key_ Struct Reference	173
3.103.1 Detailed Description	173
3.103.2 Member Data Documentation	173
3.103.2.1 attr	174
3.103.2.2 key_signatrue	174
3.103.2.3 key_usage	174
3.103.2.4 prikey	174
3.103.2.5 prikey_enc_size	174
3.103.2.6 pubkey	174
3.104ehsm_key_attr_data_ Struct Reference	175
3.104.1 Detailed Description	175
3.104.2 Member Data Documentation	175
3.104.2.1 algo_id	175
3.104.2.2 key_identifier	175
3.104.2.3 key_info	175
3.104.2.4 key_signatrue_off	176
3.104.2.5 key_usage_size	176
3.104.2.6 valid_util	176
3.105ehsm_key_copy_param Struct Reference	176
3.105.1 Detailed Description	176
3.105.2 Member Data Documentation	176
3.105.2.1 key_auth_size	177
3.105.2.2 key_auth_value	177
3.105.2.3 key_element_data	177
3.105.2.4 key_element_size	177
3.105.2.5 parent_key_handle	177

3.105.2.6 target_key_handle	177
3.106ehsm_key_derived_param Struct Reference	178
3.106.1 Detailed Description	178
3.106.2 Member Data Documentation	178
3.106.2.1 derive_type	178
3.106.2.2 itera_times	178
3.106.2.3 key_deriv_func	179
3.106.2.4 key_element_data	179
3.106.2.5 key_element_size	179
3.106.2.6 key_handle	179
3.106.2.7 key_size	179
3.106.2.8 parent_key_author_size	179
3.106.2.9 parent_key_author_value	180
3.106.2.10parent_key_handle	180
3.106.2.11passwd	180
3.106.2.12passwd_size	180
3.106.2.13salt_data	180
3.106.2.14salt_size	180
3.106.2.15type	181
3.107ehsm_key_flags_element_st Struct Reference	181
3.107.1 Detailed Description	181
3.107.2 Member Data Documentation	181
3.107.2.1 auth_flag	181
3.107.2.2 auth_size	181
3.107.2.3 auth_value	182
3.107.2.4 auth_value_exist_flags	182
3.107.2.5 trnsp_flags	182
3.107.2.6 use_flags	182
3.108ehsm_key_remove_cmd Struct Reference	182
3.108.1 Detailed Description	182
3.108.2 Member Data Documentation	183

3.108.2.1 key_auth_size	183
3.108.2.2 key_auth_value	183
3.108.2.3 key_handle	183
3.108.2.4 reserved1	183
3.109ehsm_key_remove_param Struct Reference	183
3.109.1 Detailed Description	184
3.109.2 Member Data Documentation	184
3.109.2.1 key_auth_size	184
3.109.2.2 key_auth_value	184
3.109.2.3 key_handle	184
3.110ehsm_key_signature_ Struct Reference	184
3.110.1 Detailed Description	185
3.110.2 Member Data Documentation	185
3.110.2.1 sign_id	185
3.110.2.2 sign_info	185
3.110.2.3 sign_key_id	185
3.110.2.4 signatrue	185
3.110.2.5 target_key_id	185
3.111ehsm_key_status_ Struct Reference	186
3.111.1 Detailed Description	186
3.111.2 Member Data Documentation	186
3.111.2.1 activeUseFlag	186
3.111.2.2 algo_id	186
3.111.2.3 cert_data	186
3.111.2.4 cert_size	187
3.111.2.5 key_sign_data	187
3.111.2.6 keyId	187
3.111.2.7 keyIdSize	187
3.111.2.8 mem_location	187
3.111.2.9 pubkey	187
3.111.2.10valid_util	188

3.112ehsm_key_status_cmd Struct Reference	188
3.112.1 Detailed Description	188
3.112.2 Member Data Documentation	188
3.112.2.1 cert_key_auth_size	188
3.112.2.2 cert_key_auth_value	188
3.112.2.3 cert_key_handle	189
3.112.2.4 key_handle	189
3.112.2.5 key_status	189
3.112.2.6 key_status_size	189
3.112.2.7 reserved1	189
3.112.2.8 reserved2	189
3.113ehsm_key_status_param Struct Reference	190
3.113.1 Detailed Description	190
3.113.2 Member Data Documentation	190
3.113.2.1 certification_key_auth_size	190
3.113.2.2 certification_key_auth_value	190
3.113.2.3 certification_key_handle	190
3.113.2.4 key_handle	191
3.113.2.5 key_status	191
3.113.2.6 key_status_buffer_size	191
3.113.2.7 key_status_size	191
3.114ehsm_key_usages_st Struct Reference	191
3.114.1 Detailed Description	192
3.114.2 Member Data Documentation	192
3.114.2.1 decrypt	192
3.114.2.2 dhkey	192
3.114.2.3 encrypt	192
3.114.2.4 remove	192
3.114.2.5 secureboot	192
3.114.2.6 securestorage	193
3.114.2.7 sign	193

3.114.2.8 timestamp	193
3.114.2.9 transport	193
3.114.2.10utcsync	193
3.114.2.11verify	193
3.115ehsm_keyexchange_key_info Struct Reference	194
3.115.1 Detailed Description	194
3.115.2 Member Data Documentation	194
3.115.2.1 key_auth_size	194
3.115.2.2 key_auth_value	194
3.115.2.3 key_handle	194
3.116ehsm_low_power_cmd Struct Reference	194
3.116.1 Detailed Description	195
3.116.2 Member Data Documentation	195
3.116.2.1 power_mode	195
3.116.2.2 reserved	195
3.117ehsm_mailbox_req Struct Reference	195
3.117.1 Detailed Description	196
3.117.2 Member Data Documentation	196
3.117.2.1 api_type	197
3.117.2.2 close_debug	197
3.117.2.3 cmd_id	197
3.117.2.4 copy_key	197
3.117.2.5 create_dh_key	197
3.117.2.6 debug_authentication	197
3.117.2.7 derive_key	198
3.117.2.8 ehsm_cmd	198
3.117.2.9 ehsm_gen_key	198
3.117.2.10ehsm_gen_sm9_userpriv_key	198
3.117.2.11export_key	198
3.117.2.12fw_encrypt_key	198
3.117.2.13fw_random_key	199

3.117.2.14	gen_usertmp	199
3.117.2.15	get_challenge	199
3.117.2.16	get_emu	199
3.117.2.17	get_mast_pubkey	199
3.117.2.18	get_pub_from_priv	199
3.117.2.19	get_she_id	200
3.117.2.20	get_tmp_pubkey	200
3.117.2.21	image_upgrade	200
3.117.2.22	image_verify	200
3.117.2.23	import_key	200
3.117.2.24	key_remove	200
3.117.2.25	key_status	201
3.117.2.26	module_status	201
3.117.2.27	otp_read	201
3.117.2.28	otp_write	201
3.117.2.29	rev	201
3.117.2.30	she_load_export_key	201
3.117.2.31	she_load_plain_key	202
3.117.2.32	sm9_exchg_key	202
3.117.2.33	sm9_export_key	202
3.117.2.34	sm9_import_key	202
3.117.2.35	sm9_remove_key	202
3.117.2.36	sm9_unwrap_key	202
3.117.2.37	sm9_wrap_key	203
3.117.2.38	soc_boot	203
3.117.2.39	soc_image_verify	203
3.117.2.40	uart_cmd	203
3.118	ehsm_mbox_cancel_channel_req Struct Reference	203
3.118.1	Detailed Description	203
3.118.2	Member Data Documentation	204
3.118.2.1	api_type	204

3.118.2.2 cancel_type	204
3.118.2.3 cmd_tag	204
3.118.2.4 rev	204
3.119ehsm_mbox_cancel_channel_rps Struct Reference	204
3.119.1 Detailed Description	205
3.119.2 Member Data Documentation	205
3.119.2.1 api_type	205
3.119.2.2 cancel_type	205
3.119.2.3 cmd_tag	205
3.119.2.4 ret_code	205
3.119.2.5 rev	205
3.120ehsm_mbox_mgr_channel_req Struct Reference	206
3.120.1 Detailed Description	206
3.120.2 Member Data Documentation	206
3.120.2.1 change_control_field_cmd	206
3.120.2.2 change_lifecycle_cmd	206
3.120.2.3 cmd_id	206
3.120.2.4 ehsm_cmd	207
3.120.2.5 low_power_cmd	207
3.120.2.6 self_test_cmd	207
3.120.2.7 sensor_resp_init_cmd	207
3.120.2.8 set_baudrate_cmd	207
3.121ehsm_module_status_cmd Struct Reference	207
3.121.1 Detailed Description	208
3.121.2 Member Data Documentation	208
3.121.2.1 algo_id	208
3.121.2.2 key_auth_addr	208
3.121.2.3 key_auth_size	208
3.121.2.4 key_handle	208
3.121.2.5 reserved	209
3.121.2.6 signatrue	209

3.121.2.7 signatrue_size	209
3.121.2.8 status_addr	209
3.121.2.9 status_size	209
3.121.2.10type	209
3.122ehsm_module_status_st Struct Reference	210
3.122.1 Detailed Description	210
3.122.2 Member Data Documentation	210
3.122.2.1 algo_id	210
3.122.2.2 key_auth_size	210
3.122.2.3 key_auth_value	211
3.122.2.4 key_handle	211
3.122.2.5 sign	211
3.122.2.6 sign_size	211
3.122.2.7 status	211
3.122.2.8 status_size	211
3.122.2.9 type	212
3.123ehsm_otp_read_cmd Struct Reference	212
3.123.1 Detailed Description	212
3.123.2 Member Data Documentation	212
3.123.2.1 ehsm_src_addr	212
3.123.2.2 host_dst_addr	212
3.123.2.3 size	213
3.124ehsm_otp_read_param_st Struct Reference	213
3.124.1 Detailed Description	213
3.124.2 Member Data Documentation	213
3.124.2.1 flash_read_addr	213
3.124.2.2 otp_data_addr	213
3.124.2.3 read_data_size	214
3.125ehsm_otp_write_cmd Struct Reference	214
3.125.1 Detailed Description	214
3.125.2 Member Data Documentation	214

3.125.2.1 ehsm_dst_addr	214
3.125.2.2 host_src_addr	214
3.125.2.3 size	215
3.126ehsm_otp_write_param_st Struct Reference	215
3.126.1 Detailed Description	215
3.126.2 Member Data Documentation	215
3.126.2.1 flash_write_addr	215
3.126.2.2 otp_data_addr	215
3.126.2.3 write_data_size	216
3.127ehsm_prikey_data_ Union Reference	216
3.127.1 Detailed Description	216
3.127.2 Member Data Documentation	216
3.127.2.1 dh	216
3.127.2.2 dh_k	216
3.127.2.3 ecc_k	217
3.127.2.4 kdf_k	217
3.127.2.5 rsa_crt	217
3.127.2.6 rsa_d	217
3.127.2.7 sym_k	217
3.128ehsm_pub_key_ Struct Reference	217
3.128.1 Detailed Description	218
3.128.2 Member Data Documentation	218
3.128.2.1 algo_id	218
3.128.2.2 key_pub_data	218
3.128.2.3 key_size_info	218
3.129ehsm_pubkey_data_ Union Reference	218
3.129.1 Detailed Description	219
3.129.2 Member Data Documentation	219
3.129.2.1 dh	219
3.129.2.2 ecc	219
3.129.2.3 rsa	219

3.130ehsm_rng_generate_cmd Struct Reference	219
3.130.1 Detailed Description	220
3.130.2 Member Data Documentation	220
3.130.2.1 algorithm	220
3.130.2.2 random_data_addr	220
3.130.2.3 request_size	220
3.130.2.4 rev1	220
3.130.2.5 rev2	221
3.130.2.6 rev3	221
3.130.2.7 rev4	221
3.130.2.8 rev_key_auth_size	221
3.130.2.9 rev_key_auth_value	221
3.130.2.10rev_key_handle	221
3.131ehsm_rsa crt_param_ Struct Reference	222
3.131.1 Detailed Description	222
3.131.2 Member Data Documentation	222
3.131.2.1 dp	222
3.131.2.2 dq	222
3.131.2.3 p	222
3.131.2.4 q	223
3.131.2.5 u	223
3.132ehsm_rsa_dh_key_size_ Struct Reference	223
3.132.1 Detailed Description	223
3.132.2 Member Data Documentation	223
3.132.2.1 reserved	223
3.132.2.2 rsa_dh_g_size	224
3.132.2.3 rsa_dh_p_size	224
3.132.2.4 rsa_dh_q_size	224
3.133ehsm_rsa_key_size_ Struct Reference	224
3.133.1 Detailed Description	224
3.133.2 Member Data Documentation	224

3.133.2.1 reserved	225
3.133.2.2 rsa_d_size	225
3.133.2.3 rsa_e_size	225
3.133.2.4 rsa_n_size	225
3.134ehsm_rsa_pubkey Struct Reference	225
3.134.1 Detailed Description	225
3.134.2 Member Data Documentation	226
3.134.2.1 e	226
3.134.2.2 n	226
3.135ehsm_se_key_ Struct Reference	226
3.135.1 Detailed Description	226
3.135.2 Member Data Documentation	226
3.135.2.1 algo_id	227
3.135.2.2 auth_size	227
3.135.2.3 auth_value	227
3.135.2.4 key_data	227
3.135.2.5 key_handle	227
3.135.2.6 key_size_info	227
3.135.2.7 reserved	228
3.136ehsm_secure_boot_st Struct Reference	228
3.136.1 Detailed Description	228
3.136.2 Member Data Documentation	228
3.136.2.1 encrypt_iv_addr	228
3.136.2.2 encrypt_iv_size	229
3.136.2.3 header_addr	229
3.136.2.4 header_size	229
3.136.2.5 image_addr	229
3.136.2.6 image_size	229
3.136.2.7 need_encryption	229
3.136.2.8 pubkey_addr	230
3.136.2.9 pubkey_size	230

3.136.2.10rev1	230
3.136.2.11sign_addr	230
3.136.2.12sign_size	230
3.136.2.13storage_alg	230
3.136.2.14type	231
3.137ehsm_self_test_cmd Struct Reference	231
3.137.1 Detailed Description	231
3.137.2 Member Data Documentation	231
3.137.2.1 test_type	231
3.138ehsm_sensor_init_param_st Struct Reference	231
3.138.1 Detailed Description	232
3.138.2 Member Data Documentation	232
3.138.2.1 data	232
3.138.2.2 size	232
3.139ehsm_sensor_resp_init_cmd Struct Reference	232
3.139.1 Detailed Description	232
3.139.2 Member Data Documentation	232
3.139.2.1 data_addr	233
3.139.2.2 data_size	233
3.140ehsm_service Struct Reference	233
3.140.1 Detailed Description	233
3.140.2 Member Data Documentation	233
3.140.2.1 reqhdl	233
3.140.2.2 rsphdl	234
3.140.2.3 service_id	234
3.140.2.4 timeout	234
3.141ehsm_service_info Struct Reference	234
3.141.1 Detailed Description	234
3.141.2 Member Data Documentation	234
3.141.2.1 api_type	235
3.141.2.2 cb	235

3.141.2.3 priority	235
3.141.2.4 req_type	235
3.141.2.5 service_ctx	235
3.142ehsm_set_baudrate_cmd Struct Reference	235
3.142.1 Detailed Description	236
3.142.2 Member Data Documentation	236
3.142.2.1 baud_div	236
3.143ehsm_she_get_id_param_st Struct Reference	236
3.143.1 Detailed Description	236
3.143.2 Member Data Documentation	236
3.143.2.1 challenge	236
3.143.2.2 challenge_size	237
3.143.2.3 signatrue	237
3.143.2.4 signatrue_size	237
3.143.2.5 status	237
3.143.2.6 status_size	237
3.144ehsm_she_key_host_param Struct Reference	237
3.144.1 Detailed Description	238
3.144.2 Member Data Documentation	238
3.144.2.1 m1	238
3.144.2.2 m2	238
3.144.2.3 m3	238
3.144.2.4 m4	238
3.144.2.5 m5	239
3.144.2.6 she_ext_flag	239
3.145ehsm_she_key_param Struct Reference	239
3.145.1 Detailed Description	239
3.145.2 Member Data Documentation	239
3.145.2.1 m1	239
3.145.2.2 m2	240
3.145.2.3 m3	240

3.145.2.4 m4	240
3.145.2.5 m5	240
3.145.2.6 she_ext_flag	240
3.146ehsm_she_key_st Struct Reference	240
3.146.1 Detailed Description	241
3.146.2 Member Data Documentation	241
3.146.2.1 counter	241
3.146.2.2 raw_key	241
3.146.2.3 reserved	241
3.146.2.4 secure_flag	241
3.147ehsm_she_load_export_key_cmd Struct Reference	242
3.147.1 Detailed Description	242
3.147.2 Member Data Documentation	242
3.147.2.1 m1	242
3.147.2.2 m2	242
3.147.2.3 m3	242
3.147.2.4 m4	243
3.147.2.5 m5	243
3.147.2.6 she_ext_flag	243
3.148ehsm_she_load_plain_key_cmd Struct Reference	243
3.148.1 Detailed Description	243
3.148.2 Member Data Documentation	243
3.148.2.1 key_data	244
3.149ehsm_she_plain_key_host_param Struct Reference	244
3.149.1 Detailed Description	244
3.149.2 Member Data Documentation	244
3.149.2.1 key_data	244
3.150ehsm_she_plain_key_param Struct Reference	244
3.150.1 Detailed Description	245
3.150.2 Member Data Documentation	245
3.150.2.1 key_data	245

3.151ehsm_sm9_exchg_gen_usertmp_cmd Struct Reference	245
3.151.1 Detailed Description	245
3.151.2 Member Data Documentation	245
3.151.2.1 kgc_public_key	245
3.151.2.2 peer_id	246
3.151.2.3 peer_id_size	246
3.151.2.4 rev1	246
3.151.2.5 rev_key_auth_size	246
3.151.2.6 rev_key_handle	246
3.151.2.7 type	246
3.152ehsm_sm9_exchg_key_cmd Struct Reference	247
3.152.1 Detailed Description	247
3.152.2 Member Data Documentation	247
3.152.2.1 info_stuct	247
3.152.2.2 key_size	247
3.152.2.3 rev1	247
3.152.2.4 rev2	248
3.152.2.5 rev3	248
3.152.2.6 rev4	248
3.152.2.7 role	248
3.152.2.8 type	248
3.152.2.9 user_priv_key_handle	248
3.152.2.10user_tmp_key_handle	249
3.153ehsm_sm9_exchg_key_cmd_child Struct Reference	249
3.153.1 Detailed Description	249
3.153.2 Member Data Documentation	249
3.153.2.1 fp12g	249
3.153.2.2 kgc_pub_key	249
3.153.2.3 peer_id	250
3.153.2.4 peer_id_size	250
3.153.2.5 peer_tmp_pub	250

3.153.2.6 s1_s2	250
3.153.2.7 sa_sb	250
3.153.2.8 self_id	250
3.153.2.9 self_id_size	251
3.154ehsm_sm9_exckey_gen_tmpkey_param Struct Reference	251
3.154.1 Detailed Description	251
3.154.2 Member Data Documentation	251
3.154.2.1 key_handle	251
3.154.2.2 kgc_pub_key	251
3.154.2.3 peer_id	252
3.154.2.4 peer_id_size	252
3.154.2.5 priv_key_type	252
3.154.2.6 type	252
3.155ehsm_sm9_export_key_cmd Struct Reference	252
3.155.1 Detailed Description	253
3.155.2 Member Data Documentation	253
3.155.2.1 authenticated_key	253
3.155.2.2 authenticated_key_size	253
3.155.2.3 encrypted_key	253
3.155.2.4 encrypted_key_size	253
3.155.2.5 key_handle	253
3.155.2.6 rev1	254
3.155.2.7 rev_key_auth	254
3.155.2.8 rev_key_auth_size	254
3.156ehsm_sm9_gen_mast_pubkey Struct Reference	254
3.156.1 Detailed Description	254
3.156.2 Member Data Documentation	254
3.156.2.1 key_type	255
3.156.2.2 pub_key	255
3.157ehsm_sm9_gen_tmp_pubkey_param Struct Reference	255
3.157.1 Detailed Description	255

3.157.2 Member Data Documentation	255
3.157.2.1 id_size	255
3.157.2.2 key_handle	256
3.157.2.3 pub_key	256
3.157.2.4 user_id	256
3.158ehsm_sm9_get_mast_pubkey_cmd Struct Reference	256
3.158.1 Detailed Description	256
3.158.2 Member Data Documentation	256
3.158.2.1 master_key_type	257
3.158.2.2 public_key_addr	257
3.158.2.3 reserved2	257
3.159ehsm_sm9_get_tmp_pubkey_cmd Struct Reference	257
3.159.1 Detailed Description	257
3.159.2 Member Data Documentation	257
3.159.2.1 key_handle	258
3.159.2.2 public_key_addr	258
3.159.2.3 reserved1	258
3.159.2.4 reserved2	258
3.159.2.5 user_id	258
3.159.2.6 user_id_size	258
3.160ehsm_sm9_import_key_cmd Struct Reference	259
3.160.1 Detailed Description	259
3.160.2 Member Data Documentation	259
3.160.2.1 authenticated_key	259
3.160.2.2 authenticated_key_size	259
3.160.2.3 encrypted_key	259
3.160.2.4 encrypted_key_size	260
3.160.2.5 key_is_plain	260
3.160.2.6 rev1	260
3.160.2.7 rev_key_auth	260
3.160.2.8 rev_key_auth_size	260

3.160.2.9 rev_key_handle	260
3.160.2.10type	261
3.161ehsm_sm9_inexport_key_param Struct Reference	261
3.161.1 Detailed Description	261
3.161.2 Member Data Documentation	261
3.161.2.1 key_auth_size	261
3.161.2.2 key_auth_value	261
3.161.2.3 key_blob	262
3.161.2.4 key_blob_size	262
3.161.2.5 key_handle	262
3.161.2.6 key_is_plain	262
3.161.2.7 type	262
3.162ehsm_sm9_remove_key_cmd Struct Reference	262
3.162.1 Detailed Description	263
3.162.2 Member Data Documentation	263
3.162.2.1 key_handle	263
3.162.2.2 reserved1	263
3.163ehsm_sm9_unwrap_key_cmd Struct Reference	263
3.163.1 Detailed Description	264
3.163.2 Member Data Documentation	264
3.163.2.1 cipher_addr	264
3.163.2.2 cipher_size	264
3.163.2.3 id_addr	264
3.163.2.4 id_size	264
3.163.2.5 key_addr	264
3.163.2.6 key_size	265
3.163.2.7 rev1	265
3.163.2.8 rev_key_auth	265
3.163.2.9 rev_key_auth_size	265
3.163.2.10user_priv_key_handle	265
3.164ehsm_sm9_unwrap_key_param Struct Reference	265

3.164.1 Detailed Description	266
3.164.2 Member Data Documentation	266
3.164.2.1 cipher_addr	266
3.164.2.2 cipher_size	266
3.164.2.3 id_addr	266
3.164.2.4 id_size	266
3.164.2.5 key_addr	267
3.164.2.6 key_size	267
3.164.2.7 user_priv_key_handle	267
3.165ehsm_sm9_wrap_key_cmd Struct Reference	267
3.165.1 Detailed Description	267
3.165.2 Member Data Documentation	268
3.165.2.1 fp12g	268
3.165.2.2 hid	268
3.165.2.3 id_addr	268
3.165.2.4 id_size	268
3.165.2.5 key_addr	268
3.165.2.6 key_size	268
3.165.2.7 pub_key	269
3.165.2.8 rev1	269
3.165.2.9 rev2	269
3.165.2.10rev3	269
3.165.2.11rev4	269
3.165.2.12rev_key_auth_size	269
3.165.2.13rev_key_handle	270
3.166ehsm_sm9_wrap_key_param Struct Reference	270
3.166.1 Detailed Description	270
3.166.2 Member Data Documentation	270
3.166.2.1 fp12g	270
3.166.2.2 hid	270
3.166.2.3 id_size	271

3.166.2.4 key_addr	271
3.166.2.5 key_size	271
3.166.2.6 pub_key	271
3.166.2.7 user_id	271
3.167ehsm_soc_image_verify_cmd Struct Reference	271
3.167.1 Detailed Description	272
3.167.2 Member Data Documentation	272
3.167.2.1 header_addr	272
3.167.2.2 header_size	272
3.167.2.3 pubkey_addr	272
3.167.2.4 pubkey_size	273
3.167.2.5 resered1	273
3.167.2.6 resered2	273
3.167.2.7 resered3	273
3.167.2.8 storage_alg	273
3.167.2.9 storage_encryption_flag	273
3.167.2.10storage_image_addr	274
3.167.2.11storage_image_size	274
3.167.2.12storage_iv_addr	274
3.167.2.13storage_iv_size	274
3.167.2.14storage_sign_addr	274
3.167.2.15storage_sign_size	274
3.167.2.16type	275
3.168ehsm_soc_image_verify_st Struct Reference	275
3.168.1 Detailed Description	275
3.168.2 Member Data Documentation	275
3.168.2.1 encrypt_iv_addr	275
3.168.2.2 encrypt_iv_size	276
3.168.2.3 header_addr	276
3.168.2.4 header_size	276
3.168.2.5 image_addr	276

3.168.2.6 image_size	276
3.168.2.7 need_encryption	276
3.168.2.8 pubkey_addr	277
3.168.2.9 pubkey_size	277
3.168.2.10rev1	277
3.168.2.11sign_addr	277
3.168.2.12sign_size	277
3.168.2.13storage_alg	277
3.168.2.14type	278
3.169ehsm_soc_secure_boot_status_st Struct Reference	278
3.169.1 Detailed Description	278
3.169.2 Member Data Documentation	278
3.169.2.1 status	278
3.170ehsm_storage_area_param_st Struct Reference	278
3.170.1 Detailed Description	279
3.170.2 Member Data Documentation	279
3.170.2.1 addr	279
3.170.2.2 size	279
3.171ehsm_sym_key_size_ Struct Reference	279
3.171.1 Detailed Description	279
3.171.2 Member Data Documentation	279
3.171.2.1 key_size	280
3.171.2.2 reserved	280
3.172ehsm_tick_value Struct Reference	280
3.172.1 Detailed Description	280
3.172.2 Member Data Documentation	280
3.172.2.1 current_ticks	280
3.172.2.2 tick_accuracy	281
3.172.2.3 tick_length	281
3.173ehsm_uart_cmd Struct Reference	281
3.173.1 Detailed Description	281

3.173.2 Member Data Documentation	281
3.173.2.1 uart_cmd_buffer	281
3.174hash_hmac_t Struct Reference	282
3.174.1 Detailed Description	282
3.174.2 Member Data Documentation	282
3.174.2.1 hash_hmac	282
3.174.2.2 hash_hmac_size	282
3.174.2.3 utc_time	282
3.175hash_testvec Struct Reference	282
3.175.1 Detailed Description	283
3.175.2 Member Data Documentation	283
3.175.2.1 digest	283
3.175.2.2 digest_error	283
3.175.2.3 iv	283
3.175.2.4 iv_len	284
3.175.2.5 key	284
3.175.2.6 ksize	284
3.175.2.7 np	284
3.175.2.8 plaintext	284
3.175.2.9 psize	284
3.175.2.10setkey_error	285
3.175.2.11tap	285
3.176HSM_AsymCfgType Struct Reference	285
3.176.1 Detailed Description	285
3.176.2 Member Data Documentation	285
3.176.2.1 AsymAlgo	286
3.176.2.2 CipherDir	286
3.176.2.3 HAlgo	286
3.176.2.4 KeyId	286
3.176.2.5 Padding	286
3.176.2.6 SignDir	287

3.176.2.7 Sync	287
3.177HSM_BootCfgType Struct Reference	287
3.177.1 Detailed Description	288
3.177.2 Member Data Documentation	288
3.177.2.1 BootAddr	288
3.177.2.2 HeaderAddr	288
3.177.2.3 HeaderSize	288
3.177.2.4 InitLSram	288
3.177.2.5 InitUSram	288
3.177.2.6 LogEnable	289
3.177.2.7 PllFreq	289
3.177.2.8 PubKeyAddr	289
3.177.2.9 Reserve1	289
3.177.2.10ResetDisable	289
3.177.2.11SignAddr	289
3.177.2.12StbWaitHSM	290
3.177.2.13StbWaitVerify	290
3.177.2.14SwitchClock	290
3.177.2.15VerifySize	290
3.177.2.16VersionAddr	290
3.177.2.17VersionUpdateEn	290
3.177.2.18WdgTimeout	291
3.178HSM_CMacCfgType Struct Reference	291
3.178.1 Detailed Description	291
3.178.2 Member Data Documentation	291
3.178.2.1 KeyId	291
3.178.2.2 MacDir	291
3.178.2.3 SymAlgo	292
3.178.2.4 Sync	292
3.179HSM_DebugAuthConfigType Struct Reference	292
3.179.1 Detailed Description	292

3.179.2 Member Data Documentation	292
3.179.2.1 Alg	293
3.179.2.2 PubKey	293
3.179.2.3 PubKeySize	293
3.179.2.4 Signature	293
3.179.2.5 SignatureSize	293
3.179.2.6 Type	293
3.180HSM_DeriveKeyCfgType Struct Reference	294
3.180.1 Detailed Description	294
3.180.2 Member Data Documentation	294
3.180.2.1 Kdf	294
3.180.2.2 KeyId	294
3.180.2.3 KeySize	295
3.180.2.4 KeyType	295
3.180.2.5 KeyUsage	295
3.180.2.6 KeyUsageSize	295
3.180.2.7 SaltData	295
3.180.2.8 SaltDataSize	295
3.180.2.9 ValidUntil	296
3.181HSM_DhParamType Struct Reference	296
3.181.1 Detailed Description	296
3.181.2 Member Data Documentation	296
3.181.2.1 g	296
3.181.2.2 p	296
3.181.2.3 q	297
3.182HSM_DhPriKeyType Struct Reference	297
3.182.1 Detailed Description	297
3.182.2 Member Data Documentation	297
3.182.2.1 Priv	297
3.183Hsm_DhPubKeyType_ Struct Reference	297
3.183.1 Detailed Description	298

3.183.2 Member Data Documentation	298
3.183.2.1 DhParam	298
3.183.2.2 pub	298
3.184HSM_EccPubKeyType Struct Reference	298
3.184.1 Detailed Description	298
3.184.2 Member Data Documentation	299
3.184.2.1 p	299
3.185HSM_FlashKeyPageType Struct Reference	299
3.185.1 Detailed Description	299
3.185.2 Member Data Documentation	299
3.185.2.1 KeyInfo	299
3.185.2.2 PageReverse	300
3.185.2.3 PageValid	300
3.186HSM_FlashKeyType Struct Reference	300
3.186.1 Detailed Description	300
3.186.2 Member Data Documentation	300
3.186.2.1 HandleInfo	300
3.186.2.2 IndexInfo	301
3.186.2.3 SlotInfo	301
3.187HSM_GenKeyCfgType Struct Reference	301
3.187.1 Detailed Description	301
3.187.2 Member Data Documentation	301
3.187.2.1 KeyAlgo	301
3.187.2.2 KeyId	302
3.187.2.3 KeySize	302
3.187.2.4 KeyType	302
3.187.2.5 KeyUsage	302
3.187.2.6 KeyUsageSize	302
3.187.2.7 ValidUntil	302
3.188HSM_HMacCfgType Struct Reference	303
3.188.1 Detailed Description	303

3.188.2 Member Data Documentation	303
3.188.2.1 HAlgo	303
3.188.2.2 KeyId	303
3.188.2.3 MacDir	303
3.188.2.4 Sync	304
3.189HSM_ImageVerifyType__ Struct Reference	304
3.189.1 Detailed Description	304
3.189.2 Member Data Documentation	304
3.189.2.1 HeaderAddr	305
3.189.2.2 HeaderSize	305
3.189.2.3 PubkeyAddr	305
3.189.2.4 PubkeySize	305
3.189.2.5 StorageAlg	305
3.189.2.6 StorageEncryptionFlag	305
3.189.2.7 StorageImageAddr	306
3.189.2.8 StorageImageSize	306
3.189.2.9 StorageIvAddr	306
3.189.2.10StorageIvSize	306
3.189.2.11StorageSignAddr	306
3.189.2.12StorageSignSize	306
3.189.2.13Type	307
3.189.2.14UpdateVersionFlag	307
3.189.2.15VersionAddr	307
3.189.2.16VersionSize	307
3.190HSM_InOutMacType Struct Reference	307
3.190.1 Detailed Description	308
3.190.2 Member Data Documentation	308
3.190.2.1 BasicInOut	308
3.190.2.2 MacInBuf	308
3.190.2.3 MacInBufLen	308
3.190.2.4 Vry	308

3.191HSM_InOutSignType Struct Reference	309
3.191.1 Detailed Description	309
3.191.2 Member Data Documentation	309
3.191.2.1 BasicInOut	309
3.191.2.2 SignInBuf	309
3.191.2.3 SignInBufLen	309
3.191.2.4 Vry	310
3.192HSM_InOutType Struct Reference	310
3.192.1 Detailed Description	310
3.192.2 Member Data Documentation	310
3.192.2.1 InBuf	310
3.192.2.2 InBufLen	310
3.192.2.3 OutBuf	311
3.192.2.4 OutBufLen	311
3.193HSM_KeyActUseFlagsType Struct Reference	311
3.193.1 Detailed Description	311
3.193.2 Member Data Documentation	312
3.193.2.1 createkey	312
3.193.2.2 decrypt	312
3.193.2.3 encrypt	312
3.193.2.4 remove	312
3.193.2.5 secureboot	312
3.193.2.6 securestorage	312
3.193.2.7 sign	313
3.193.2.8 timestamp	313
3.193.2.9 transport	313
3.193.2.10utcsync	313
3.193.2.11verify	313
3.194HSM_KeyFlagsElementType Struct Reference	313
3.194.1 Detailed Description	314
3.194.2 Member Data Documentation	314

3.194.2.1 auth_flag	314
3.194.2.2 auth_size	314
3.194.2.3 auth_value	314
3.194.2.4 auth_value_exist_flags	314
3.194.2.5 trnsp_flags	315
3.194.2.6 use_flags	315
3.195HSM_KeyHandleInfoType Struct Reference	315
3.195.1 Detailed Description	315
3.195.2 Member Data Documentation	315
3.195.2.1 AuthSize	315
3.195.2.2 AuthValue	316
3.195.2.3 KeyHandle	316
3.196HSM_KeyIndexInfoType Struct Reference	316
3.196.1 Detailed Description	316
3.196.2 Member Data Documentation	316
3.196.2.1 KeyIndex	316
3.196.2.2 KeyValid	317
3.197HSM_KeySlotInfoType Struct Reference	317
3.197.1 Detailed Description	317
3.197.2 Member Data Documentation	317
3.197.2.1 SlotIndex	317
3.197.2.2 SlotValid	317
3.198HSM_KeyStatusType Struct Reference	318
3.198.1 Detailed Description	318
3.198.2 Member Data Documentation	318
3.198.2.1 CertificationAuth	318
3.198.2.2 CertificationAuthSize	318
3.198.2.3 CertificationKeyld	318
3.198.2.4 KeyStatus	319
3.198.2.5 KeyStatusSize	319
3.198.2.6 TargetKeyld	319

3.199HSM_KeyUsagesType Struct Reference	319
3.199.1 Detailed Description	319
3.199.2 Member Data Documentation	320
3.199.2.1 decrypt	320
3.199.2.2 dhkey	320
3.199.2.3 encrypt	320
3.199.2.4 remove	320
3.199.2.5 secureboot	320
3.199.2.6 securestorage	320
3.199.2.7 sign	321
3.199.2.8 timestamp	321
3.199.2.9 transport	321
3.199.2.10utcsync	321
3.199.2.11verify	321
3.200HSM_PlainKeyCfgType Struct Reference	321
3.200.1 Detailed Description	322
3.200.2 Member Data Documentation	322
3.200.2.1 AuthValue	322
3.200.2.2 AuthValueSize	322
3.200.2.3 ExtParam	323
3.200.2.4 KeyAlgo	323
3.200.2.5 KeyId	323
3.200.2.6 KeyType	323
3.200.2.7 KeyUsages	323
3.200.2.8 KeyUsagesCnt	324
3.200.2.9 PrivKey	324
3.200.2.10PrivKeyLen	324
3.200.2.11PubKey	324
3.200.2.12PubKeyLen	324
3.200.2.13RandomKeySize	325
3.200.2.14ValidUtil	325

3.201Hsm_PriKeyDataType_ Union Reference	325
3.201.1 Detailed Description	325
3.201.2 Member Data Documentation	325
3.201.2.1 Dh	326
3.201.2.2 DhKey	326
3.201.2.3 EccKey	326
3.201.2.4 KdfKey	326
3.201.2.5 RsaCtr	326
3.201.2.6 RsaD	326
3.201.2.7 SymKey	327
3.202Hsm_PubKeyDataType_ Union Reference	327
3.202.1 Detailed Description	327
3.202.2 Member Data Documentation	327
3.202.2.1 Dh	327
3.202.2.2 Ecc	327
3.202.2.3 Rsa	328
3.203HSM_RamKeyHeadType Struct Reference	328
3.203.1 Detailed Description	328
3.203.2 Member Data Documentation	328
3.203.2.1 KeyInfo	328
3.203.2.2 KeyMemStart	328
3.203.2.3 KeyMemUsedSize	329
3.203.2.4 KeyNum	329
3.204HSM_RamKeyInfoType Struct Reference	329
3.204.1 Detailed Description	329
3.204.2 Member Data Documentation	329
3.204.2.1 KeyHandleAddr	329
3.204.2.2 KeyIndex	330
3.204.2.3 Used	330
3.205Hsm_RsaCrtType_ Struct Reference	330
3.205.1 Detailed Description	330

3.205.2 Member Data Documentation	330
3.205.2.1 dp	331
3.205.2.2 dq	331
3.205.2.3 p	331
3.205.2.4 q	331
3.205.2.5 u	331
3.206HSM_RsaPubKeyType Struct Reference	331
3.206.1 Detailed Description	332
3.206.2 Member Data Documentation	332
3.206.2.1 e	332
3.206.2.2 n	332
3.207HSM_SecretKeyCfgType Struct Reference	332
3.207.1 Detailed Description	333
3.207.2 Member Data Documentation	333
3.207.2.1 AuthValue	333
3.207.2.2 AuthValueSize	333
3.207.2.3 KeyAlgo	333
3.207.2.4 KeyType	333
3.207.2.5 SecretKeyBlob	333
3.207.2.6 SetKeyBlobSize	334
3.207.2.7 SetKeyUseFlag	334
3.208HSM_SecureUpgradeType__ Struct Reference	334
3.208.1 Detailed Description	335
3.208.2 Member Data Documentation	335
3.208.2.1 CheckVersionFlag	335
3.208.2.2 CtxAddr	335
3.208.2.3 CtxSize	335
3.208.2.4 HeaderAddr	335
3.208.2.5 HeaderSize	335
3.208.2.6 MacSignAddr	336
3.208.2.7 MacSignSize	336

3.208.2.8 ProcessMode	336
3.208.2.9 Rev	336
3.208.2.10StorageAlg	336
3.208.2.11StorageEncryptionFlag	336
3.208.2.12StorageImageAddr	337
3.208.2.13StorageImageSize	337
3.208.2.14StorageIvAddr	337
3.208.2.15StorageIvSize	337
3.208.2.16UpgradeAlg	337
3.208.2.17UpgradeDecryptionFlag	337
3.208.2.18UpgradeImageAddr	338
3.208.2.19UpgradeImageSize	338
3.208.2.20UpgradeIvAddr	338
3.208.2.21UpgradeIvSize	338
3.208.2.22UpgradePubkeyAddr	338
3.208.2.23UpgradePubkeySize	338
3.208.2.24UpgradeSignAddr	339
3.208.2.25UpgradeSignSize	339
3.208.2.26UpgradeVersionAddr	339
3.208.2.27UpgradeVersionSize	339
3.209HSM_SymCfgType Struct Reference	339
3.209.1 Detailed Description	340
3.209.2 Member Data Documentation	340
3.209.2.1 CipherDir	340
3.209.2.2 CMode	340
3.209.2.3 Iv	340
3.209.2.4 IvLen	340
3.209.2.5 KeyId	341
3.209.2.6 Padding	341
3.209.2.7 SymAlgo	341
3.209.2.8 Sync	341

3.210kdf_testvec Struct Reference	341
3.210.1 Detailed Description	342
3.210.2 Member Data Documentation	342
3.210.2.1 key	342
3.210.2.2 ksize	342
3.210.2.3 plaintext	342
3.210.2.4 psize	342
3.210.2.5 salt	343
3.210.2.6 salt_size	343
3.210.2.7 shared_info	343
3.210.2.8 shared_info_size	343
3.211key_act_use_flags_t Struct Reference	343
3.211.1 Detailed Description	344
3.211.2 Member Data Documentation	344
3.211.2.1 createkey	344
3.211.2.2 decrypt	344
3.211.2.3 encrypt	344
3.211.2.4 remove	344
3.211.2.5 secureboot	344
3.211.2.6 securestorage	345
3.211.2.7 sign	345
3.211.2.8 timestamp	345
3.211.2.9 transport	345
3.211.2.10utcsync	345
3.211.2.11verify	345
3.212key_info_st Struct Reference	346
3.212.1 Detailed Description	346
3.212.2 Member Data Documentation	346
3.212.2.1 dh_key_size	346
3.212.2.2 dh_param	346
3.212.2.3 kdf_key_size	346

3.212.2.4 key_handle	347
3.212.2.5 random_key_size	347
3.212.2.6 rsa_e_bytes_size	347
3.212.2.7 storage_info	347
3.212.2.8 storage_key_type	347
3.213kpp_testvec Struct Reference	347
3.213.1 Detailed Description	348
3.213.2 Member Data Documentation	348
3.213.2.1 b_public	348
3.213.2.2 b_public_size	349
3.213.2.3 b_secret	349
3.213.2.4 b_secret_size	349
3.213.2.5 expected_a_public	349
3.213.2.6 expected_a_public_size	349
3.213.2.7 expected_ss	349
3.213.2.8 expected_ss_size	350
3.213.2.9 g	350
3.213.2.10g_size	350
3.213.2.11genkey	350
3.213.2.12p	350
3.213.2.13p_size	350
3.213.2.14q	351
3.213.2.15q_size	351
3.213.2.16secret	351
3.213.2.17secret_size	351
3.214mac_t Struct Reference	351
3.214.1 Detailed Description	351
3.214.2 Member Data Documentation	352
3.214.2.1 mac_size	352
3.214.2.2 mac_value	352
3.214.2.3 utc_time	352

3.215mailbox_channel Struct Reference	352
3.215.1 Detailed Description	352
3.215.2 Member Data Documentation	353
3.215.2.1 cmd_inprogres	353
3.215.2.2 h2s_note_bit	353
3.215.2.3 h2s_start_addr	353
3.215.2.4 h2s_word_size	353
3.215.2.5 s2h_note_bit	353
3.215.2.6 s2h_start_addr	353
3.215.2.7 s2h_word_size	354
3.215.2.8 surpport_asyn	354
3.215.2.9 type	354
3.216ehsm_cmd_cipher_st::osr_cmd_hdr_u Union Reference	354
3.216.1 Detailed Description	354
3.216.2 Member Data Documentation	354
3.216.2.1 hdr_eccp_keygen	355
3.216.2.2 hdr_ecise	355
3.216.2.3 hdr_pke	355
3.216.2.4 hdr_rng	355
3.216.2.5 hdr_rsa_keygen	355
3.216.2.6 hdr_ske	355
3.216.2.7 hdr_sm9	356
3.217rsacipher_testvec Struct Reference	356
3.217.1 Detailed Description	356
3.217.2 Member Data Documentation	356
3.217.2.1 c	357
3.217.2.2 c_size	357
3.217.2.3 d	357
3.217.2.4 d_byte_size	357
3.217.2.5 dp	357
3.217.2.6 dp_byte_size	357

3.217.2.7 dq	358
3.217.2.8 dq_byte_size	358
3.217.2.9 e	358
3.217.2.10e_byte_size	358
3.217.2.11hash_alg	358
3.217.2.12s crt_mode	358
3.217.2.13m	359
3.217.2.14m_size	359
3.217.2.15n	359
3.217.2.16n_byte_size	359
3.217.2.17p	359
3.217.2.18p_byte_size	359
3.217.2.19public_key_vec	360
3.217.2.20q	360
3.217.2.21q_byte_size	360
3.217.2.22salt	360
3.217.2.23salt_byte_size	360
3.217.2.24siggen_sigver_test	360
3.217.2.25u	361
3.217.2.26u_byte_size	361
3.218signature_t Struct Reference	361
3.218.1 Detailed Description	361
3.218.2 Member Data Documentation	361
3.218.2.1 signature	361
3.218.2.2 signature_size	362
3.218.2.3 utc_time	362
3.219sm2_ext_param Struct Reference	362
3.219.1 Detailed Description	362
3.219.2 Member Data Documentation	362
3.219.2.1 local_tmp_key_auth_size	362
3.219.2.2 local_tmp_key_auth_value	363

3.219.2.3 local_tmp_key_handle	363
3.219.2.4 peer_pubkey	363
3.219.2.5 peer_temp_pubkey	363
3.219.2.6 s1_s2_value	363
3.219.2.7 sa_sb_value	363
3.219.2.8 sm2_role	364
3.220sm9cipher_testvec Struct Reference	364
3.220.1 Detailed Description	364
3.220.2 Member Data Documentation	364
3.220.2.1 c	364
3.220.2.2 c_sz	365
3.220.2.3 enc_typde	365
3.220.2.4 h	365
3.220.2.5 hid	365
3.220.2.6 id	365
3.220.2.7 id_sz	365
3.220.2.8 K2len	366
3.220.2.9 m	366
3.220.2.10m_sz	366
3.220.2.11padding	366
3.220.2.12Ppub	366
3.220.2.13priv	366
3.220.2.14r	367
3.220.2.15r_sz	367
3.220.2.16sig	367
3.220.2.17siggen_sigver_test	367
3.221soc_image_upgrade_info Struct Reference	367
3.221.1 Detailed Description	368
3.221.2 Member Data Documentation	368
3.221.2.1 check_version_flag	368
3.221.2.2 cmd_input	368

3.221.2.3 header	369
3.221.2.4 header_size	369
3.221.2.5 mac_sign	369
3.221.2.6 mac_sign_size	369
3.221.2.7 process_mode	369
3.221.2.8 storage_alg	369
3.221.2.9 storage_encryption_flag	370
3.221.2.10 storage_image	370
3.221.2.11 storage_image_size	370
3.221.2.12 storage_iv	370
3.221.2.13 storage_iv_size	370
3.221.2.14 upgrade_alg	370
3.221.2.15 upgrade_decryption_flag	371
3.221.2.16 upgrade_image	371
3.221.2.17 upgrade_image_size	371
3.221.2.18 upgrade_iv	371
3.221.2.19 upgrade_iv_size	371
3.221.2.20 upgrade_pubkey	371
3.221.2.21 upgrade_pubkey_size	372
3.221.2.22 upgrade_sign	372
3.221.2.23 upgrade_sign_size	372
3.221.2.24 upgrade_version	372
3.221.2.25 upgrade_version_size	372
3.222 soc_image_upgrade_input Struct Reference	372
3.222.1 Detailed Description	373
3.222.2 Member Data Documentation	373
3.222.2.1 check_version_flag	373
3.222.2.2 ctx_addr	373
3.222.2.3 ctx_size	374
3.222.2.4 header_addr	374
3.222.2.5 header_size	374

3.222.2.6 mac_sign_addr	374
3.222.2.7 mac_sign_size	374
3.222.2.8 process_mode	374
3.222.2.9 rev	375
3.222.2.10storage_alg	375
3.222.2.11storage_encryption_flag	375
3.222.2.12storage_image_addr	375
3.222.2.13storage_image_size	375
3.222.2.14storage_iv_addr	375
3.222.2.15storage_iv_size	376
3.222.2.16upgrade_alg	376
3.222.2.17upgrade_decryption_flag	376
3.222.2.18upgrade_image_addr	376
3.222.2.19upgrade_image_size	376
3.222.2.20upgrade_iv_addr	376
3.222.2.21upgrade_iv_size	377
3.222.2.22upgrade_pubkey_addr	377
3.222.2.23upgrade_pubkey_size	377
3.222.2.24upgrade_sign_addr	377
3.222.2.25upgrade_sign_size	377
3.222.2.26upgrade_version_addr	377
3.222.2.27upgrade_version_size	378
3.223soc_image_verify_info Struct Reference	378
3.223.1 Detailed Description	378
3.223.2 Member Data Documentation	378
3.223.2.1 cmd_input	378
3.223.2.2 header	379
3.223.2.3 header_size	379
3.223.2.4 pubkey	379
3.223.2.5 pubkey_size	379
3.223.2.6 storage_alg	379

3.223.2.7 storage_encryption_flag	379
3.223.2.8 storage_image	380
3.223.2.9 storage_image_size	380
3.223.2.10 storage_iv	380
3.223.2.11 storage_iv_size	380
3.223.2.12 storage_sign	380
3.223.2.13 storage_sign_size	380
3.223.2.14 type	381
3.223.2.15 update_version_flag	381
3.223.2.16 version	381
3.223.2.17 version_size	381
3.224 soc_image_verify_input Struct Reference	381
3.224.1 Detailed Description	382
3.224.2 Member Data Documentation	382
3.224.2.1 header_addr	382
3.224.2.2 header_size	382
3.224.2.3 pubkey_addr	382
3.224.2.4 pubkey_size	382
3.224.2.5 storage_alg	382
3.224.2.6 storage_encryption_flag	383
3.224.2.7 storage_image_addr	383
3.224.2.8 storage_image_size	383
3.224.2.9 storage_iv_addr	383
3.224.2.10 storage_iv_size	383
3.224.2.11 storage_sign_addr	383
3.224.2.12 storage_sign_size	384
3.224.2.13 type	384
3.224.2.14 update_version_flag	384
3.224.2.15 version_addr	384
3.224.2.16 version_size	384

4 File Documentation	385
4.1 AC784xx_API_Reference_Manual_HSM.pdf File Reference	385
4.2 AC784xx_Hsm_Reg.h File Reference	385
4.2.1 Detailed Description	386
4.2.2 Macro Definition Documentation	386
4.2.2.1 HSM_CMD_BOOTROM_SET_BAUDRATE	386
4.2.2.2 HSM_CMD_ENCRYPT_KEY	386
4.2.2.3 HSM_CMD_GET_RANDOM_KEY	386
4.2.2.4 OTP_BASE_ADDR	386
4.2.2.5 OTP_ERR_RSP_CTRL_ADDR	387
4.2.2.6 OTP_FW_CTRL_FIELD_ADDR_H	387
4.2.2.7 OTP_FW_CTRL_FIELD_ADDR_L	387
4.2.2.8 OTP_HOST_CTRL_FIELD_ADDR_H	387
4.2.2.9 OTP_HOST_CTRL_FIELD_ADDR_L	387
4.2.2.10 OTP_HSM_ENABLE_ADDR	387
4.2.2.11 OTP_HSM_VERSION_ADDR	388
4.2.2.12 OTP_HW_CTRL_FIELD_ADDR	388
4.2.2.13 OTP_KEY_ADDR	388
4.2.2.14 OTP_KEY_ATTR_BYTE_LENGTH	388
4.2.2.15 OTP_KEY_ATTR_ENCODE_EACH_LENGTH	388
4.2.2.16 OTP_KEY_ATTR_LC	388
4.2.2.17 OTP_KEY_CRC	389
4.2.2.18 OTP_KEY_CRC_SIZE	389
4.2.2.19 OTP_KEY_SIZE	389
4.2.2.20 OTP_LIFE_CYCLE_ADDR	389
4.2.2.21 OTP_SECURE_BOOT_ADDR	389
4.2.2.22 OTP_SIZE	389
4.2.2.23 OTP_SOC_VERSION_ADDR	390
4.2.2.24 OTP_UID_ADDR	390
4.2.2.25 OTP_VERSION_ENCODE_LENGTH	390
4.2.2.26 OTP_VERSION_LENGTH	390

4.2.2.27	SOC_CMD_GET_HSM_FW_VERSION	390
4.2.2.28	SOC_CMD_IMAGE_UPGRADE_UPGRADE	390
4.2.2.29	SOC_CMD_IMAGE_VERIFY	391
4.3	Asr_Standard_Types.h File Reference	391
4.3.1	Macro Definition Documentation	394
4.3.1.1	CRYPTO_E_BUSY	394
4.3.1.2	CRYPTO_E_ENTROPY_EXHAUSTION	394
4.3.1.3	CRYPTO_E_JOB_CANCELED	394
4.3.1.4	CRYPTO_E_KEY_EMPTY	395
4.3.1.5	CRYPTO_E_KEY_NOT_AVAILABLE	395
4.3.1.6	CRYPTO_E_KEY_NOT_VALID	395
4.3.1.7	CRYPTO_E_KEY_READ_FAIL	395
4.3.1.8	CRYPTO_E_KEY_SIZE_MISMATCH	395
4.3.1.9	CRYPTO_E_KEY_WRITE_FAIL	396
4.3.1.10	CRYPTO_E_QUEUE_FULL	396
4.3.1.11	CRYPTO_E_SMALL_BUFFER	396
4.3.1.12	CRYPTO_KE_CERTIFICATE_CURRENT_TIME	396
4.3.1.13	CRYPTO_KE_CERTIFICATE_DATA	396
4.3.1.14	CRYPTO_KE_CERTIFICATE_EXTENSIONS	396
4.3.1.15	CRYPTO_KE_CERTIFICATE_ISSUER	397
4.3.1.16	CRYPTO_KE_CERTIFICATE_PARSING_FORMAT	397
4.3.1.17	CRYPTO_KE_CERTIFICATE_SERIALNUMBER	397
4.3.1.18	CRYPTO_KE_CERTIFICATE_SIGNATURE	397
4.3.1.19	CRYPTO_KE_CERTIFICATE_SIGNATURE_ALGORITHM	397
4.3.1.20	CRYPTO_KE_CERTIFICATE_SUBJECT	397
4.3.1.21	CRYPTO_KE_CERTIFICATE_SUBJECT_PUBLIC_KEY	398
4.3.1.22	CRYPTO_KE_CERTIFICATE_VALIDITY_NOT_AFTER	398
4.3.1.23	CRYPTO_KE_CERTIFICATE_VALIDITY_NOT_BEFORE	398
4.3.1.24	CRYPTO_KE_CERTIFICATE_VERSION	398
4.3.1.25	CRYPTO_KE_CIPHER_2NDKEY	398
4.3.1.26	CRYPTO_KE_CIPHER_IV	398

4.3.1.27	CRYPTO_KE_CIPHER_KEY	399
4.3.1.28	CRYPTO_KE_CIPHER_PROOF	399
4.3.1.29	CRYPTO_KE_KEYDERIVATION_ALGORITHM	399
4.3.1.30	CRYPTO_KE_KEYDERIVATION_ITERATIONS	399
4.3.1.31	CRYPTO_KE_KEYDERIVATION_PASSWD	399
4.3.1.32	CRYPTO_KE_KEYDERIVATION_SALT	399
4.3.1.33	CRYPTO_KE_KEYEXCHANGE_ALGORITHM	400
4.3.1.34	CRYPTO_KE_KEYEXCHANGE_BASE	400
4.3.1.35	CRYPTO_KE_KEYEXCHANGE_OWNPUKEY	400
4.3.1.36	CRYPTO_KE_KEYEXCHANGE_PRIVKEY	400
4.3.1.37	CRYPTO_KE_KEYGENERATE_ALGORITHM	400
4.3.1.38	CRYPTO_KE_KEYGENERATE_KEY	400
4.3.1.39	CRYPTO_KE_KEYGENERATE_SEED	401
4.3.1.40	CRYPTO_KE_MAC_KEY	401
4.3.1.41	CRYPTO_KE_MAC_PROOF	401
4.3.1.42	CRYPTO_KE_RANDOM_ALGORITHM	401
4.3.1.43	CRYPTO_KE_RANDOM_SEED_STATE	401
4.3.1.44	CRYPTO_KE_SIGNATURE_KEY	401
4.3.1.45	CYRPTO_KE_KEYEXCHANGE_SHAREDVALUE	402
4.3.1.46	E_NOT_OK	402
4.3.1.47	E_OK	402
4.3.2	Typedef Documentation	402
4.3.2.1	Crypto_ConfigType	402
4.3.2.2	Std_HsmReturnType	402
4.3.3	Enumeration Type Documentation	402
4.3.3.1	Crypto_AlgorithmFamilyType	402
4.3.3.2	Crypto_AlgorithmModeType	403
4.3.3.3	Crypto_InputOutputRedirectionConfigType	404
4.3.3.4	Crypto_JobStateType	404
4.3.3.5	Crypto_KeyElementReadAccessType	405
4.3.3.6	Crypto_KeyElementWriteAccessType	405

4.3.3.7	Crypto_OperationModeType	405
4.3.3.8	Crypto_ProcessingType	406
4.3.3.9	Crypto_ServiceInfoType	406
4.3.3.10	Crypto_VerifyResultType	407
4.3.3.11	CryptoDriverStateType	408
4.3.3.12	CryptoKeyFormatType	408
4.4	eHSM_Com_Struct_Ip.h File Reference	408
4.4.1	Macro Definition Documentation	412
4.4.1.1	CODE_VALID_FLAG	413
4.4.1.2	DEFAULT_RSAKEY_E_SIZE	413
4.4.1.3	EHSM_CODE_VERIFY_FALG	413
4.4.1.4	EHSM_EVITA_AUTH_VALUE_MAX_SIZE	413
4.4.1.5	EHSM_FAST_CMAC_AES128	413
4.4.1.6	EHSM_FAST_CMAC_EVITA_KEY	413
4.4.1.7	EHSM_FAST_CMAC_GEN	414
4.4.1.8	EHSM_FAST_CMAC_SHE_KEY	414
4.4.1.9	EHSM_FAST_CMAC_SM4	414
4.4.1.10	EHSM_FAST_CMAC_VERIFY	414
4.4.1.11	EHSM_GET_STATUS_ERRORS	414
4.4.1.12	EHSM_GET_STATUS_MEM	414
4.4.1.13	EHSM_GET_STATUS_SBB	415
4.4.1.14	EHSM_GET_STATUS_SHE	415
4.4.1.15	EHSM_SELF_TEST_ALL	415
4.4.1.16	EHSM_SELF_TEST_HASH	415
4.4.1.17	EHSM_SELF_TEST_HASH_MD5	415
4.4.1.18	EHSM_SELF_TEST_HASH_SHA1	416
4.4.1.19	EHSM_SELF_TEST_HASH_SHA2	416
4.4.1.20	EHSM_SELF_TEST_HASH_SHA256	416
4.4.1.21	EHSM_SELF_TEST_HASH_SHA3	416
4.4.1.22	EHSM_SELF_TEST_HASH_SM3	416
4.4.1.23	EHSM_SELF_TEST_PKE	416

4.4.1.24	EHSM_SELF_TEST_PKE_ECC	417
4.4.1.25	EHSM_SELF_TEST_PKE_RSA	417
4.4.1.26	EHSM_SELF_TEST_PKE_SM2	417
4.4.1.27	EHSM_SELF_TEST_PKE_SM9	417
4.4.1.28	EHSM_SELF_TEST_SKE	417
4.4.1.29	EHSM_SELF_TEST_SKE_AES	417
4.4.1.30	EHSM_SELF_TEST_SKE_DES	418
4.4.1.31	EHSM_SELF_TEST_SKE_SM4	418
4.4.1.32	EHSM_SELF_TEST_SKE_TDES	418
4.4.1.33	EHSM_SELF_TEST_TRNG	418
4.4.1.34	EHSM_SHE_M1_MAX_SIZE	418
4.4.1.35	EHSM_SHE_M2_MAX_SIZE	418
4.4.1.36	EHSM_SHE_M3_MAX_SIZE	419
4.4.1.37	EHSM_SHE_M4_MAX_SIZE	419
4.4.1.38	EHSM_SHE_M5_MAX_SIZE	419
4.4.1.39	IMAGE_ANALYSIS_CODE	419
4.4.1.40	IMAGE_DECRYPT_CODE	419
4.4.1.41	IMAGE_ENCRYPT_CODE	419
4.4.1.42	IMAGE_PUBLIC_KEY_MAX_LENGTH	420
4.4.1.43	IMAGE_SIGNATURE_MAX_LENGTH	420
4.4.1.44	OTP_CONTROL_FILED_BYTE_SIZE	420
4.4.1.45	SECURE_BOOT_TYPE_IMAGE_VERIFY	420
4.4.1.46	SECURE_BOOT_TYPE_SECURE_BOOT	420
4.4.1.47	SOC_BOOT_TYPE_PARALLEL	420
4.4.1.48	SOC_BOOT_TYPE_SEQUENTIAL	421
4.4.1.49	SOC_CODE_VERIFY_FALG	421
4.4.1.50	UPGRADE_VALID_FLAG	421
4.4.2	Typedef Documentation	421
4.4.2.1	ehsm_create_dh_key_param_st	421
4.4.2.2	ehsm_create_evita_key_param_st	421
4.4.2.3	ehsm_create_random_key_param_st	421

4.4.2.4	ehsm_crypto_randomgenerate_param_st	421
4.4.2.5	ehsm_emu_status_st	422
4.4.2.6	ehsm_evita_key_export_st	422
4.4.2.7	ehsm_evita_memory_info_st	422
4.4.2.8	ehsm_exchange_sm9_key_param_st	422
4.4.2.9	ehsm_fw_encrypt_key_st	422
4.4.2.10	ehsm_fw_random_key_st	422
4.4.2.11	ehsm_gen_sm9_key_param_st	422
4.4.2.12	ehsm_gen_sm9_master_key_param_st	422
4.4.2.13	ehsm_gen_sm9_userpriv_key_param_st	423
4.4.2.14	ehsm_get_pub_from_priv_param_st	423
4.4.2.15	ehsm_image_upgrade_st	423
4.4.2.16	ehsm_key_copy_param_st	423
4.4.2.17	ehsm_key_derived_param_st	423
4.4.2.18	ehsm_key_remove_param_st	423
4.4.2.19	ehsm_key_status_param_st	423
4.4.2.20	ehsm_keyexchange_key_info_st	423
4.4.2.21	ehsm_she_key_host_param_st	424
4.4.2.22	ehsm_she_key_param_st	424
4.4.2.23	ehsm_she_plain_key_host_param_st	424
4.4.2.24	ehsm_she_plain_key_param_st	424
4.4.2.25	ehsm_sm9_exckey_gen_tmpkey_st	424
4.4.2.26	ehsm_sm9_gen_mast_pubkey_st	424
4.4.2.27	ehsm_sm9_gen_tmp_pubkey_st	424
4.4.2.28	ehsm_sm9_inexport_key_param_st	424
4.4.2.29	ehsm_sm9_unwrap_key_param_st	425
4.4.2.30	ehsm_sm9_wrap_key_param_st	425
4.4.2.31	ehsm_soc_image_upgrade_info_st	425
4.4.2.32	ehsm_soc_image_upgrade_input_st	425
4.4.2.33	ehsm_soc_image_verify_info_st	425
4.4.2.34	ehsm_soc_image_verify_input_st	425

4.4.2.35	sm2_ext_param_st	425
4.4.3	Enumeration Type Documentation	425
4.4.3.1	crypto_key_derive_type_e	425
4.4.3.2	ehsm_api_type_e	426
4.4.3.3	ehsm_challenge_type_e	426
4.4.3.4	ehsm_code_upgrade_alg_e	426
4.4.3.5	ehsm_code_verify_alg_e	427
4.4.3.6	ehsm_control_field_type_e	427
4.4.3.7	ehsm_debug_auth_alg_e	427
4.4.3.8	ehsm_dh_mode_e	428
4.4.3.9	ehsm_fw_encrypt_key_slot_e	428
4.4.3.10	ehsm_fw_encrypt_key_type_e	428
4.4.3.11	ehsm_fw_random_key_slot_e	429
4.4.3.12	ehsm_fw_random_key_type_e	429
4.4.3.13	ehsm_gen_sm9_key_type_e	429
4.4.3.14	ehsm_image_process_mode_e	430
4.4.3.15	ehsm_key_mem_type_e	430
4.4.3.16	ehsm_lifecycle_e	430
4.4.3.17	ehsm_rsa_key_type_e	430
4.4.3.18	ehsm_SM9_exchg_key_role_e	431
4.4.3.19	ehsm_sm9_master_key_type_e	431
4.4.3.20	ehsm_sm9_user_privkey_type_e	431
4.4.3.21	ehsm_uart_baudrate_e	432
4.4.3.22	sm2_key_exchange_role_e	432
4.5	eHSM_Compt_Bitmap.c File Reference	432
4.5.1	Function Documentation	433
4.5.1.1	ehsm_bitmap_clr()	433
4.5.1.2	ehsm_bitmap_count()	433
4.5.1.3	ehsm_bitmap_first()	433
4.5.1.4	ehsm_bitmap_init()	434
4.5.1.5	ehsm_bitmap_last()	434

4.5.1.6	ehsm_bitmap_reset()	434
4.5.1.7	ehsm_bitmap_set()	434
4.6	eHSM_Compt_Bitmap.h File Reference	435
4.6.1	Typedef Documentation	435
4.6.1.1	bitmap_st	435
4.6.2	Function Documentation	435
4.6.2.1	ehsm_bitmap_clr()	436
4.6.2.2	ehsm_bitmap_count()	436
4.6.2.3	ehsm_bitmap_create()	436
4.6.2.4	ehsm_bitmap_destroy()	436
4.6.2.5	ehsm_bitmap_first()	437
4.6.2.6	ehsm_bitmap_init()	437
4.6.2.7	ehsm_bitmap_last()	437
4.6.2.8	ehsm_bitmap_reset()	437
4.6.2.9	ehsm_bitmap_set()	438
4.7	eHSM_Compt_List.h File Reference	438
4.7.1	Macro Definition Documentation	438
4.7.1.1	container_of	438
4.7.1.2	dlist_entry	439
4.7.1.3	dlist_for_each	439
4.7.1.4	dlist_for_each_safe	439
4.7.1.5	DLIST_HEAD	439
4.7.1.6	DLIST_HEAD_INIT	439
4.7.1.7	DLIST_POISON1	440
4.7.1.8	DLIST_POISON2	440
4.7.1.9	OFFSET	440
4.8	eHSM_Config_Ip.h File Reference	440
4.8.1	Macro Definition Documentation	446
4.8.1.1	CONFIG_EHSM_ARCH_HOST_MAILBOX_POLLING	446
4.8.1.2	CONFIG_EHSM_ARCH_V_CMD_QUEUE_SIZE	447
4.8.1.3	CONFIG_EHSM_ARCH_V_CRYPTOBJ_HASH_QUEUE_SIZE	447

4.8.1.4	CONFIG_EHSM_ARCH_V_CRYPTOBJ_K_QUEUE_SIZE	447
4.8.1.5	CONFIG_EHSM_ARCH_V_CRYPTOBJ_PKE_QUEUE_SIZE	447
4.8.1.6	CONFIG_EHSM_ARCH_V_CRYPTOBJ_SKE_QUEUE_SIZE	447
4.8.1.7	CONFIG_EHSM_ARCH_V_CRYPTOBJ_SYSMGR_QUEUE_SIZE	447
4.8.1.8	CONFIG_EHSM_ARCH_V_CRYPTOBJ_TRNG_QUEUE_SIZE	448
4.8.1.9	CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT	448
4.8.1.10	CONFIG_EHSM_ARCH_V_JTAG_TIMEOUT	448
4.8.1.11	CONFIG_EHSM_ARCH_V_MAILBOX_TIMEOUT	448
4.8.1.12	CONFIG_EHSM_ARCH_V_RSA_K_CMD_TIMEOUT	448
4.8.1.13	CONFIG_EHSM_AUTOSAR	449
4.8.1.14	CONFIG_EHSM_EVITA	449
4.8.1.15	CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_PARTIAL_ACCESS	449
4.8.1.16	CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_PERSIST	449
4.8.1.17	CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_READ_ACCESS	449
4.8.1.18	CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_WRITE_ACCESS	449
4.8.1.19	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_PARTIAL_ACCESS	450
4.8.1.20	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_PERSIST	450
4.8.1.21	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_READ_ACCESS	450
4.8.1.22	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_WRITE_ACCESS	450
4.8.1.23	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_PARTIAL_ACCESS	450
4.8.1.24	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_PERSIST	450
4.8.1.25	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_READ_ACCESS	451
4.8.1.26	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_WRITE_ACCESS	451
4.8.1.27	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_PARTIAL_ACCESS	451
4.8.1.28	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_PERSIST	451
4.8.1.29	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_READ_ACCESS	451
4.8.1.30	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_WRITE_ACCESS	451
4.8.1.31	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_PARTIAL_ACCESS	452
4.8.1.32	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_PERSIST	452
4.8.1.33	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_READ_ACCESS	452

4.8.1.34	CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_WRITE_ACCESS	452
4.8.1.35	CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_PARTIAL_ACCESS	452
4.8.1.36	CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_PERSIST	452
4.8.1.37	CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_READ_ACCESS	453
4.8.1.38	CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_WRITE_ACCESS	453
4.8.1.39	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_PARTIAL_ACCESS	453
4.8.1.40	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_PERSIST	453
4.8.1.41	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_READ_ACCESS	453
4.8.1.42	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_WRITE_ACCESS	453
4.8.1.43	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_PARTIAL_ACCESS	454
4.8.1.44	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_PERSIST	454
4.8.1.45	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_READ_ACCESS	454
4.8.1.46	CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_WRITE_ACCESS	454
4.8.1.47	CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PARTIAL_ACCESS	454
4.8.1.48	CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PERSIST	454
4.8.1.49	CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_READ_ACCESS	455
4.8.1.50	CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_WRITE_ACCESS	455
4.8.1.51	CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_PARTIAL_ACCESS	455
4.8.1.52	CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_PERSIST	455
4.8.1.53	CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_READ_ACCESS	455
4.8.1.54	CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_WRITE_ACCESS	455
4.8.1.55	CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_PARTIAL_ACCESS	456
4.8.1.56	CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_PERSIST	456
4.8.1.57	CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_READ_ACCESS	456
4.8.1.58	CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_WRITE_ACCESS	456
4.8.1.59	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_PARTIAL_ACCESS	456
4.8.1.60	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_PERSIST	456
4.8.1.61	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_READ_ACCESS	457
4.8.1.62	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_WRITE_ACCESS	457
4.8.1.63	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_PARTIAL_ACCESS	457
4.8.1.64	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_PERSIST	457

4.8.1.65	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_READ_ACCESS	457
4.8.1.66	CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_WRITE_ACCESS	457
4.8.1.67	CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_PARTIAL_ACCESS	458
4.8.1.68	CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_PERSIST	458
4.8.1.69	CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_READ_ACCESS	458
4.8.1.70	CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_WRITE_ACCESS	458
4.8.1.71	CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_PARTIAL_ACCESS	458
4.8.1.72	CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_PERSIST	458
4.8.1.73	CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_READ_ACCESS	459
4.8.1.74	CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_WRITE_ACCESS	459
4.8.1.75	CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_PARTIAL_ACCESS	459
4.8.1.76	CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_PERSIST	459
4.8.1.77	CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_READ_ACCESS	459
4.8.1.78	CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_WRITE_ACCESS	459
4.8.1.79	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_PARTIAL_ACCESS	460
4.8.1.80	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_PERSIST	460
4.8.1.81	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_READ_ACCESS	460
4.8.1.82	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_WRITE_ACCESS	460
4.8.1.83	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_PARTIAL_ACCESS	460
4.8.1.84	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_PERSIST	460
4.8.1.85	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_READ_ACCESS	461
4.8.1.86	CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_WRITE_ACCESS	461
4.8.1.87	CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_PARTIAL_ACCESS	461
4.8.1.88	CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_PERSIST	461
4.8.1.89	CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_READ_ACCESS	461
4.8.1.90	CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_WRITE_ACCESS	461
4.8.1.91	CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_PARTIAL_ACCESS	462
4.8.1.92	CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_PERSIST	462
4.8.1.93	CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_READ_ACCESS	462
4.8.1.94	CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_WRITE_ACCESS	462
4.8.1.95	CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_PARTIAL_ACCESS	462

4.8.1.96	CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_PERSIST	462
4.8.1.97	CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_READ_ACCESS	463
4.8.1.98	CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_WRITE_ACCESS	463
4.8.1.99	CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PARTIAL_ACCESS	463
4.8.1.100	CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PERSIST	463
4.8.1.101	CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_READ_ACCESS	463
4.8.1.102	CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_WRITE_ACCESS	463
4.8.1.103	CONFIG_EHSM_KMGR_V_ASR_K_STATUS_PARTIAL_ACCESS	464
4.8.1.104	CONFIG_EHSM_KMGR_V_ASR_K_STATUS_PERSIST	464
4.8.1.105	CONFIG_EHSM_KMGR_V_ASR_K_STATUS_READ_ACCESS	464
4.8.1.106	CONFIG_EHSM_KMGR_V_ASR_K_STATUS_WRITE_ACCESS	464
4.8.1.107	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_PARTIAL_ACCESS	464
4.8.1.108	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_PERSIST	464
4.8.1.109	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_READ_ACCESS	465
4.8.1.110	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_WRITE_ACCESS	465
4.8.1.111	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_PARTIAL_ACCESS	465
4.8.1.112	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_PERSIST	465
4.8.1.113	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_READ_ACCESS	465
4.8.1.114	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_WRITE_ACCESS	465
4.8.1.115	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_PARTIAL_ACCESS	466
4.8.1.116	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_PERSIST	466
4.8.1.117	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_READ_ACCESS	466
4.8.1.118	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_WRITE_ACCESS	466
4.8.1.119	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_PARTIAL_ACCESS	466
4.8.1.120	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_PERSIST	466
4.8.1.121	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_READ_ACCESS	467
4.8.1.122	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_WRITE_ACCESS	467
4.8.1.123	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_PARTIAL_ACCESS	467
4.8.1.124	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_PERSIST	467
4.8.1.125	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_READ_ACCESS	467
4.8.1.126	CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_WRITE_ACCESS	467

4.8.1.127 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_PARTIAL_ACCESS	468
4.8.1.128 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_PERSIST	468
4.8.1.129 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_READ_ACCESS	468
4.8.1.130 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_WRITE_ACCESS	468
4.8.1.131 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_PARTIAL_ACCESS	468
4.8.1.132 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_PERSIST	468
4.8.1.133 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_READ_ACCESS	469
4.8.1.134 CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_WRITE_ACCESS	469
4.8.1.135 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_PARTIAL_ACCESS	469
4.8.1.136 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_PERSIST	469
4.8.1.137 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_READ_ACCESS	469
4.8.1.138 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_WRITE_ACCESS	469
4.8.1.139 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_PARTIAL_ACCESS	470
4.8.1.140 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_PERSIST	470
4.8.1.141 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_READ_ACCESS	470
4.8.1.142 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_WRITE_ACCESS	470
4.8.1.143 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_PARTIAL_ACCESS	470
4.8.1.144 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_PERSIST	470
4.8.1.145 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_READ_ACCESS	471
4.8.1.146 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_WRITE_ACCESS	471
4.8.1.147 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_PARTIAL_ACCESS	471
4.8.1.148 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_PERSIST	471
4.8.1.149 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_READ_ACCESS	471
4.8.1.150 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_WRITE_ACCESS	471
4.8.1.151 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_PARTIAL_ACCESS	472
4.8.1.152 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_PERSIST	472
4.8.1.153 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_READ_ACCESS	472
4.8.1.154 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_WRITE_ACCESS	472
4.8.1.155 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_PARTIAL_ACCESS	472
4.8.1.156 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_PERSIST	472
4.8.1.157 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_READ_ACCESS	473

4.8.1.158 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_WRITE_ACCESS	473
4.8.1.159 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_PARTIAL_ACCESS . . .	473
4.8.1.160 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_PERSIST	473
4.8.1.161 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_READ_ACCESS	473
4.8.1.162 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_WRITE_ACCESS	473
4.8.1.163 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_PARTIAL_ACCESS	474
4.8.1.164 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_PERSIST	474
4.8.1.165 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_READ_ACCESS . .	474
4.8.1.166 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_WRITE_ACCESS .	474
4.8.1.167 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_PARTIAL_ACCESS	474
4.8.1.168 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_PERSIST .	474
4.8.1.169 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_READ_ACCESS	475
4.8.1.170 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_WRITE_ACCESS	475
4.8.1.171 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_PARTIAL_ACCESS	475
4.8.1.172 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_PERSIST . .	475
4.8.1.173 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_READ_ACCESS	475
4.8.1.174 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_WRITE_ACCESS	475
4.8.1.175 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_PARTIAL_ACCESS . . .	476
4.8.1.176 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_PERSIST	476
4.8.1.177 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_READ_ACCESS	476
4.8.1.178 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_WRITE_ACCESS	476
4.8.1.179 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_PARTIAL_ACCESS	476
4.8.1.180 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_PERSIST . . .	476
4.8.1.181 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_READ_ACCESS	477
4.8.1.182 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_WRITE_ACCESS	477
4.8.1.183 CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_PARTIAL_ACCESS	477

4.8.1.184	CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_PERSIST . . .	477
4.8.1.185	CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_READ_ACCESS ESS	477
4.8.1.186	CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_WRITE_ACCESS CESS	477
4.8.1.187	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_PARTIAL_ACCESS . .	478
4.8.1.188	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_PERSIST	478
4.8.1.189	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_READ_ACCESS	478
4.8.1.190	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_WRITE_ACCESS	478
4.8.1.191	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_PARTIAL_ACCESS . .	478
4.8.1.192	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_PERSIST	478
4.8.1.193	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_READ_ACCESS	479
4.8.1.194	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_WRITE_ACCESS	479
4.8.1.195	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_PARTIAL_ACCESS	479
4.8.1.196	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_PERSIST	479
4.8.1.197	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_READ_ACCESS	479
4.8.1.198	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_WRITE_ACCESS	479
4.8.1.199	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_PARTIAL_ACCESS	480
4.8.1.200	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_PERSIST	480
4.8.1.201	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_READ_ACCESS	480
4.8.1.202	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_WRITE_ACCESS	480
4.8.1.203	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_PARTIAL_ACCESS	480
4.8.1.204	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_PERSIST	480
4.8.1.205	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_READ_ACCESS	481
4.8.1.206	CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_WRITE_ACCESS	481
4.8.1.207	CONFIG_EHSM_KMGR_V_ASR_MAC_K_PARTIAL_ACCESS	481
4.8.1.208	CONFIG_EHSM_KMGR_V_ASR_MAC_K_PERSIST	481
4.8.1.209	CONFIG_EHSM_KMGR_V_ASR_MAC_K_READ_ACCESS	481
4.8.1.210	CONFIG_EHSM_KMGR_V_ASR_MAC_K_WRITE_ACCESS	481
4.8.1.211	CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_PARTIAL_ACCESS	482
4.8.1.212	CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_PERSIST	482
4.8.1.213	CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_READ_ACCESS	482

4.8.1.214	CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_WRITE_ACCESS	482
4.8.1.215	CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_PARTIAL_ACCESS	482
4.8.1.216	CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_PERSIST	482
4.8.1.217	CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_READ_ACCESS	483
4.8.1.218	CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_WRITE_ACCESS	483
4.8.1.219	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PARTIAL_ACCESS	483
4.8.1.220	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PERSIST	483
4.8.1.221	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_READ_ACCESS	483
4.8.1.222	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_WRITE_ACCESS	483
4.8.1.223	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_PARTIAL_ACCESS ESS	484
4.8.1.224	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_PERSIST	484
4.8.1.225	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_READ_ACCESS	484
4.8.1.226	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_WRITE_ACCESS	484
4.8.1.227	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PARTIAL_ACCESS	484
4.8.1.228	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PERSIST	484
4.8.1.229	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_READ_ACCESS	485
4.8.1.230	CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_WRITE_ACCESS	485
4.8.1.231	CONFIG_EHSM_SHE	485
4.9	eHSM_Debug_Ip.h File Reference	485
4.9.1	Macro Definition Documentation	486
4.9.1.1	AUTOSAR_LOG_DEBUG	486
4.9.1.2	AUTOSAR_LOG_ERROR	487
4.9.1.3	AUTOSAR_LOG_INFO	487
4.9.1.4	AUTOSAR_LOG_WARN	487
4.9.1.5	AUTOSAR_PURE_LOG_DEBUG	487
4.9.1.6	COMMON_LOG_DEBUG	487
4.9.1.7	COMMON_LOG_ERROR	488
4.9.1.8	COMMON_LOG_INFO	488
4.9.1.9	COMMON_LOG_WARN	488
4.9.1.10	COMMON_PURE_LOG_DEBUG	488
4.9.1.11	CONFIG_HOST_AUTOSAR_DEBUG_ENABLE	488

4.9.1.12	CONFIG_HOST_COMMON_DEBUG_ENABLE	488
4.9.1.13	CONFIG_HOST_CUSTOM_DEBUG_ENABLE	489
4.9.1.14	CONFIG_HOST_EVITA_DEBUG_ENABLE	489
4.9.1.15	CONFIG_HOST_PERFORMANCE_DEBUG_ENABLE	489
4.9.1.16	CONFIG_HOST_SHE_DEBUG_ENABLE	489
4.9.1.17	CONFIG_HOST_V_LOG_DEBUG	489
4.9.1.18	CONFIG_HOST_V_LOG_ERR	489
4.9.1.19	CONFIG_HOST_V_LOG_INFO	490
4.9.1.20	CONFIG_HOST_V_LOG_LEVEL	490
4.9.1.21	CONFIG_HOST_V_LOG_WARN	490
4.9.1.22	CUSTOM_LOG_DEBUG	490
4.9.1.23	CUSTOM_LOG_ERROR	490
4.9.1.24	CUSTOM_LOG_INFO	490
4.9.1.25	CUSTOM_LOG_WARN	491
4.9.1.26	CUSTOM_PURE_LOG_DEBUG	491
4.9.1.27	EVITA_LOG_DEBUG	491
4.9.1.28	EVITA_LOG_ERROR	491
4.9.1.29	EVITA_LOG_INFO	491
4.9.1.30	EVITA_LOG_WARN	492
4.9.1.31	EVITA_PURE_LOG_DEBUG	492
4.9.1.32	filename	492
4.9.1.33	HOST_LOG	492
4.9.1.34	HOST_LOG_DEBUG	492
4.9.1.35	HOST_LOG_ERROR	493
4.9.1.36	HOST_LOG_INFO	493
4.9.1.37	HOST_LOG_WARN	493
4.9.1.38	HOST_PURE_LOG	493
4.9.1.39	PERFORMANCE_LOG_DEBUG	493
4.9.1.40	PERFORMANCE_LOG_ERROR	493
4.9.1.41	PERFORMANCE_LOG_INFO	494
4.9.1.42	PERFORMANCE_LOG_WARN	494

4.9.1.43	PERFORMANCE_PURE_LOG_DEBUG	494
4.9.1.44	PURE_LOG	494
4.9.1.45	SHE_LOG_DEBUG	494
4.9.1.46	SHE_LOG_ERROR	495
4.9.1.47	SHE_LOG_INFO	495
4.9.1.48	SHE_LOG_WARN	495
4.9.1.49	SHE_PURE_LOG_DEBUG	495
4.9.2	Function Documentation	495
4.9.2.1	Debug_Printf()	495
4.10	eHSM_Dspt_CryObj_lp.c File Reference	496
4.10.1	Function Documentation	496
4.10.1.1	ehsm_add_cmd_to_priority_queue()	496
4.10.1.2	ehsm_add_cmd_to_sent_queue()	496
4.10.1.3	ehsm_crypto_object_get_cmd_done()	497
4.10.1.4	ehsm_crypto_object_get_job()	497
4.10.1.5	ehsm_crypto_object_init()	497
4.10.1.6	ehsm_crypto_object_is_free()	497
4.10.1.7	ehsm_del_cmd_from_priority_queue()	498
4.10.1.8	ehsm_del_cmd_from_sent_queue()	498
4.10.1.9	ehsm_fetch_cmd_from_crypto_object()	498
4.10.1.10	hw_interrupt_disable()	498
4.10.1.11	hw_interrupt_enable()	498
4.11	eHSM_Dspt_CryObj_lp.h File Reference	498
4.11.1	Typedef Documentation	499
4.11.1.1	crypto_object_st	499
4.11.2	Enumeration Type Documentation	499
4.11.2.1	crypto_object_state_e	499
4.11.3	Function Documentation	500
4.11.3.1	ehsm_add_cmd_to_priority_queue()	500
4.11.3.2	ehsm_add_cmd_to_sent_queue()	500
4.11.3.3	ehsm_crypto_object_get_cmd_done()	500

4.11.3.4	ehsm_crypto_object_get_job()	500
4.11.3.5	ehsm_crypto_object_init()	501
4.11.3.6	ehsm_crypto_object_is_free()	501
4.11.3.7	ehsm_crypto_object_submit_cmd()	501
4.11.3.8	ehsm_del_cmd_from_priority_queue()	501
4.11.3.9	ehsm_del_cmd_from_sent_queue()	502
4.11.3.10	ehsm_fetch_cmd_from_crypto_object()	502
4.12	eHSM_Dspt_Ip.c File Reference	502
4.12.1	Function Documentation	503
4.12.1.1	Crypto_MainFunction()	503
4.12.1.2	ehsm_dispatcher_init()	503
4.12.1.3	ehsm_mbox_polling()	503
4.12.1.4	ehsm_remove_cmd_from_queue()	503
4.12.1.5	ehsm_submit_cmd_req()	504
4.12.1.6	hw_interrupt_disable()	504
4.12.1.7	hw_interrupt_enable()	504
4.12.1.8	ptest_time_counting_end()	504
4.12.1.9	ptest_time_counting_get_state()	504
4.13	eHSM_Dspt_Ip.h File Reference	504
4.13.1	Function Documentation	505
4.13.1.1	Crypto_MainFunction()	505
4.13.1.2	ehsm_dispatcher_init()	505
4.13.1.3	ehsm_remove_cmd_from_queue()	505
4.13.1.4	ehsm_submit_cmd_req()	505
4.14	eHSM_Err_Code_Ip.h File Reference	506
4.14.1	Macro Definition Documentation	511
4.14.1.1	EHSM_ERR_ALGORITHM_ERROR	511
4.14.1.2	EHSM_ERR_ALL_COUNTER_BUSY	511
4.14.1.3	EHSM_ERR_ALL_KEY_SPACE_OCCUPIED	511
4.14.1.4	EHSM_ERR_AUTH_FAILED	512
4.14.1.5	EHSM_ERR_CHALLENGE_EXPIRED	512

4.14.1.6	EHSM_ERR_CHALLENGE_FAILED	512
4.14.1.7	EHSM_ERR_CHECK_TIME_FAILED	512
4.14.1.8	EHSM_ERR_CHECK_TIME_STAMP_ERROR	512
4.14.1.9	EHSM_ERR_CMD_CANCELED	513
4.14.1.10	EHSM_ERR_CODE_MOVE_ERROR	513
4.14.1.11	EHSM_ERR_COUNTER_AUTH_FAILED	513
4.14.1.12	EHSM_ERR_COUNTER_NOT_INIT	513
4.14.1.13	EHSM_ERR_COUNTER_WRONG_ID	513
4.14.1.14	EHSM_ERR_CRYPTOCERT_PARSE_FAILED	514
4.14.1.15	EHSM_ERR_CRYPTOCERT_VERIFY_FAILED	514
4.14.1.16	EHSM_ERR_CRYPTOCERT_WRONG_ID_SIZE	514
4.14.1.17	EHSM_ERR_CRYPTOCERT_WRONG_K2_SIZE	514
4.14.1.18	EHSM_ERR_DATA_CHECK_ERROR	514
4.14.1.19	EHSM_ERR_DATA_EMPTY	515
4.14.1.20	EHSM_ERR_DEBUG_AUTH_FAILED	515
4.14.1.21	EHSM_ERR_DRBG_BUFFER_NULL	515
4.14.1.22	EHSM_ERR_DRBG_DF_OVERFLOW	515
4.14.1.23	EHSM_ERR_DRBG_LENGTH_INVALID	515
4.14.1.24	EHSM_ERR_DRBG_LENGTH_NOT_MUL_8	516
4.14.1.25	EHSM_ERR_DRBG_RESEED_FAILED	516
4.14.1.26	EHSM_ERR_EHSM_BUSY	516
4.14.1.27	EHSM_ERR_EHSM_LIFECYCLE_LIMIT	516
4.14.1.28	EHSM_ERR_EQUAL_VERSION_COUNTER	516
4.14.1.29	EHSM_ERR_ERC_NO_SECURE_BOOT	517
4.14.1.30	EHSM_ERR_GENERAL_ERROR	517
4.14.1.31	EHSM_ERR_HASH_BUFFER_NULL	517
4.14.1.32	EHSM_ERR_HASH_CONFIG_INVALID	517
4.14.1.33	EHSM_ERR_HASH_INPUT_INVALID	517
4.14.1.34	EHSM_ERR_HASH_LEN_OVERFLOW	518
4.14.1.35	EHSM_ERR_HASH_OUTPUT_ZERO_ALL	518
4.14.1.36	EHSM_ERR_HASH_WORK_ERROR	518

4.14.1.37 EHSM_ERR_HMAC_VERIFY_FAILED	518
4.14.1.38 EHSM_ERR_IMAGE_VERIFY_FAILED	518
4.14.1.39 EHSM_ERR_INVALID_CODE_FLAG	519
4.14.1.40 EHSM_ERR_INVALID_COUNTER_INCREMENTATION	519
4.14.1.41 EHSM_ERR_INVALID_KEY_FLAG	519
4.14.1.42 EHSM_ERR_IPCORE_DH_INTEGER_TOO_BIG	519
4.14.1.43 EHSM_ERR_IPCORE_DH_INVALID_INPUT	519
4.14.1.44 EHSM_ERR_IPCORE_DH_POINTER_NULL	519
4.14.1.45 EHSM_ERR_IPCORE_DH_VALUE_ONE	520
4.14.1.46 EHSM_ERR_IPCORE_DH_ZERO_ALL	520
4.14.1.47 EHSM_ERR_IPCORE_ECDH_INTEGER_TOO_BIG	520
4.14.1.48 EHSM_ERR_IPCORE_ECDH_INVALID_INPUT	520
4.14.1.49 EHSM_ERR_IPCORE_ECDH_POINTOR_NULL	520
4.14.1.50 EHSM_ERR_IPCORE_ECDH_ZERO_ALL	520
4.14.1.51 EHSM_ERR_IPCORE_ECDSA_INTEGER_TOO_BIG	521
4.14.1.52 EHSM_ERR_IPCORE_ECDSA_INVALID_INPUT	521
4.14.1.53 EHSM_ERR_IPCORE_ECDSA_POINTOR_NULL	521
4.14.1.54 EHSM_ERR_IPCORE_ECDSA_VERIFY_FAILED	521
4.14.1.55 EHSM_ERR_IPCORE_ECDSA_ZERO_ALL	521
4.14.1.56 EHSM_ERR_IPCORE_ECIES_ERROR	521
4.14.1.57 EHSM_ERR_IPCORE_ECIES_INTEGER_TOO_BIG	522
4.14.1.58 EHSM_ERR_IPCORE_ECIES_INVALID_INPUT	522
4.14.1.59 EHSM_ERR_IPCORE_ECIES_POINTOR_NULL	522
4.14.1.60 EHSM_ERR_IPCORE_ECIES_ZERO_ALL	522
4.14.1.61 EHSM_ERR_IPCORE_HASH_BUFFER_NULL	522
4.14.1.62 EHSM_ERR_IPCORE_HASH_CONFIG_INVALID	522
4.14.1.63 EHSM_ERR_IPCORE_HASH_ERROR	523
4.14.1.64 EHSM_ERR_IPCORE_HASH_INPUT_INVALID	523
4.14.1.65 EHSM_ERR_IPCORE_HASH_LEN_OVERFLOW	523
4.14.1.66 EHSM_ERR_IPCORE_HASH_OUTPUT_ZERO_ALL	523
4.14.1.67 EHSM_ERR_IPCORE_PKE_ERROR	523

4.14.1.68 EHSM_ERR_IPCORE_PKE_FINISHED	523
4.14.1.69 EHSM_ERR_IPCORE_PKE_INFINITY_POINT	524
4.14.1.70 EHSM_ERR_IPCORE_PKE_INTEGER_TOO_BIG	524
4.14.1.71 EHSM_ERR_IPCORE_PKE_INVALID_INPUT	524
4.14.1.72 EHSM_ERR_IPCORE_PKE_NO_MODINV	524
4.14.1.73 EHSM_ERR_IPCORE_PKE_NOT_ON_CURVE	524
4.14.1.74 EHSM_ERR_IPCORE_PKE_ZERO_ALL	524
4.14.1.75 EHSM_ERR_IPCORE_SKE_ATTACK_ALARM	525
4.14.1.76 EHSM_ERR_IPCORE_SKE_BUFFER_NULL	525
4.14.1.77 EHSM_ERR_IPCORE_SKE_CONFIG_INVALID	525
4.14.1.78 EHSM_ERR_IPCORE_SKE_ERROR	525
4.14.1.79 EHSM_ERR_IPCORE_SKE_INPUT_INVALID	525
4.14.1.80 EHSM_ERR_IPCORE_SKE_PADDING_ERROR	525
4.14.1.81 EHSM_ERR_IPCORE_SM2_BUFFER_NULL	526
4.14.1.82 EHSM_ERR_IPCORE_SM2_DECRYPT_VERIFY_FAILED	526
4.14.1.83 EHSM_ERR_IPCORE_SM2_EXCHANGE_ROLE_INVALID	526
4.14.1.84 EHSM_ERR_IPCORE_SM2_INPUT_INVALID	526
4.14.1.85 EHSM_ERR_IPCORE_SM2_INTEGER_TOO_BIG	526
4.14.1.86 EHSM_ERR_IPCORE_SM2_NOT_ON_CURVE	526
4.14.1.87 EHSM_ERR_IPCORE_SM2_VERIFY_FAILED	527
4.14.1.88 EHSM_ERR_IPCORE_SM2_ZERO_ALL	527
4.14.1.89 EHSM_ERR_IPCORE_TRNG_BUFFER_NULL	527
4.14.1.90 EHSM_ERR_IPCORE_TRNG_ERROR	527
4.14.1.91 EHSM_ERR_IPCORE_TRNG_HT_ERROR	527
4.14.1.92 EHSM_ERR_IPCORE_TRNG_INVALID_CONFIG	527
4.14.1.93 EHSM_ERR_IPCORE_TRNG_INVALID_INPUT	528
4.14.1.94 EHSM_ERR_IPCORE_TRNG_TIMEOUT_ERROR	528
4.14.1.95 EHSM_ERR_JOB_CANCELED	528
4.14.1.96 EHSM_ERR_KEY_BUFF_SMALLER	528
4.14.1.97 EHSM_ERR_KEY_EMPTY_ERROR	528
4.14.1.98 EHSM_ERR_KEY_INVALID_ERROR	528

4.14.1.99	EHSM_ERR_KEY_NOT_AVAILABLE_ERROR	529
4.14.1.100	EHSM_ERR_KEY_STORE_FULL	529
4.14.1.101	EHSM_ERR_KEY_UPDATE_ERROR	529
4.14.1.102	EHSM_ERR_KMGR_READ_ERROR	529
4.14.1.103	EHSM_ERR_MAILBOX_SUCCESS	529
4.14.1.104	EHSM_ERR_MEMORY_FAILURE	530
4.14.1.105	EHSM_ERR_MIDDLE_SW	530
4.14.1.106	EHSM_ERR_NO_CHALLENGE_AVAILABLE	530
4.14.1.107	EHSM_ERR_NOT_NIT	530
4.14.1.108	EHSM_ERR_NOT_SUPPORT	530
4.14.1.109	EHSM_ERR_OUT_OF_MEM	531
4.14.1.110	EHSM_ERR_PARAM_ERROR	531
4.14.1.111	EHSM_ERR_PKE_ED25519_MSG_FLOW	531
4.14.1.112	EHSM_ERR_PKE_SIGN_FAILED	531
4.14.1.113	EHSM_ERR_PKE_VERIFY_FAILED	531
4.14.1.114	EHSM_ERR_PKE_WORK_ERROR	532
4.14.1.115	EHSM_ERR_PKE_WRONG_CURVE_ID	532
4.14.1.116	EHSM_ERR_PKE_WRONG_E_SIZE	532
4.14.1.117	EHSM_ERR_PKE_WRONG_KDF_ALG	532
4.14.1.118	EHSM_ERR_PKE_WRONG_N_SIZE	532
4.14.1.119	EHSM_ERR_PKE_WRONG_RSA_CRT_MODE	533
4.14.1.120	EHSM_ERR_QUEUE_FULL	533
4.14.1.121	EHSM_ERR_REG_TWICE_NOT_MATCH	533
4.14.1.122	EHSM_ERR_REMOVE_IMPOSSIBLE	533
4.14.1.123	EHSM_ERR_SKE_IV_SHOULD_NOT_NULL	533
4.14.1.124	EHSM_ERR_SKE_MAC_VRY_FAILED	534
4.14.1.125	EHSM_ERR_SKE_WORK_ERROR	534
4.14.1.126	EHSM_ERR_SKE_WRONG_AAD_SIZE	534
4.14.1.127	EHSM_ERR_SKE_WRONG_ALG	534
4.14.1.128	EHSM_ERR_SKE_WRONG_IV_SIZE	534
4.14.1.129	EHSM_ERR_SKE_WRONG_K_SIZE	535

4.14.1.130	EHSM_ERR_SKE_WRONG_L_SIZE	535
4.14.1.131	EHSM_ERR_SKE_WRONG_MODE	535
4.14.1.132	EHSM_ERR_SKE_WRONG_PADIDNG	535
4.14.1.133	EHSM_ERR_SKE_WRONG_TAG_SIZE	535
4.14.1.134	EHSM_ERR_SMALL_BUFFER	536
4.14.1.135	EHSM_ERR_SW_SUCCESS	536
4.14.1.136	EHSM_ERR_TIME_CHALLENGE_EXPIRED	536
4.14.1.137	EHSM_ERR_TIME_STAMP_EXPIRED	536
4.14.1.138	EHSM_ERR_TIME_STAMP_VERIFY_FAILED	536
4.14.1.139	EHSM_ERR_TRANSPORT_IMPOSSIBLE	537
4.14.1.140	EHSM_ERR_TRNG_BUFFER_NULL	537
4.14.1.141	EHSM_ERR_TRNG_HT_ERROR	537
4.14.1.142	EHSM_ERR_TRNG_INVALID_CONFIG	537
4.14.1.143	EHSM_ERR_TRNG_INVALID_INPUT	537
4.14.1.144	EHSM_ERR_TRNG_TIMEOUT_ERROR	538
4.14.1.145	EHSM_ERR_TRNG_WORK_ERROR	538
4.14.1.146	EHSM_ERR_UTC_SYNCHRONIZATION_FAILED	538
4.14.1.147	EHSM_ERR_UTC_TIMER_INVALID_INDEX	538
4.14.1.148	EHSM_ERR_UTC_TIMER_NOT_SYNC	538
4.14.1.149	EHSM_ERR_WRITE_PROTECTED	539
4.14.1.150	EHSM_ERR_WRONG_ALGORITHM	539
4.14.1.151	EHSM_ERR_WRONG_AUTHORIZATION	539
4.14.1.152	EHSM_ERR_WRONG_CERT_KEY_HANDLE	539
4.14.1.153	EHSM_ERR_WRONG_CHALLENGE_TYPE	539
4.14.1.154	EHSM_ERR_WRONG_CONTEXT	539
4.14.1.155	EHSM_ERR_WRONG_DATA_LENGTH	540
4.14.1.156	EHSM_ERR_WRONG_DIRECT	540
4.14.1.157	EHSM_ERR_WRONG_EHSM_ADDR	540
4.14.1.158	EHSM_ERR_WRONG_JOB_ID	540
4.14.1.159	EHSM_ERR_WRONG_KEY_COMBINATION	540
4.14.1.160	EHSM_ERR_WRONG_KEY_DERIVE_FUNC	541

4.14.1.161EHSM_ERR_WRONG_KEY_HANDLE	541
4.14.1.162EHSM_ERR_WRONG_KEY_LEVEL	541
4.14.1.163EHSM_ERR_WRONG_KEY_LIFE_LIMIT	541
4.14.1.164EHSM_ERR_WRONG_KEY_SIZE	541
4.14.1.165EHSM_ERR_WRONG_KEY_TYPE	541
4.14.1.166EHSM_ERR_WRONG_KEY_USAGE	542
4.14.1.167EHSM_ERR_WRONG_MODULE_TYPE	542
4.14.1.168EHSM_ERR_WRONG_PADDING_TYPE	542
4.14.1.169EHSM_ERR_WRONG_PROC_MODE	542
4.14.1.170EHSM_ERR_WRONG_PUB_KEY	542
4.14.1.171EHSM_ERR_WRONG_REMOTE_KEY_HANDLE	543
4.14.1.172EHSM_ERR_WRONG_SALT_SIZE	543
4.14.1.173EHSM_ERR_WRONG_UTC_TIME	543
4.14.1.174EHSM_ERR_WRONG_VERSION_COUNTER	543
4.15 eHSM_Exclusive_Area.h File Reference	543
4.15.1 Function Documentation	543
4.15.1.1 Exclusive_area_enter()	544
4.15.1.2 Exclusive_area_exit()	544
4.16 eHSM_If_Asr_Cipher_Ip.c File Reference	544
4.16.1 Macro Definition Documentation	545
4.16.1.1 IV_BUFF_SIZE	545
4.16.2 Function Documentation	545
4.16.2.1 ehsm_aead_request()	545
4.16.2.2 ehsm_encrypt_request()	545
4.16.2.3 ehsm_hash_hmac()	546
4.16.2.4 ehsm_mac_request()	546
4.16.2.5 ehsm_random_generate()	547
4.16.2.6 ehsm_signature_gen_vry()	547
4.17 eHSM_If_Asr_Cipher_Ip.h File Reference	548
4.17.1 Function Documentation	548
4.17.1.1 ehsm_aead_request()	548

4.17.1.2	ehsm_encrypt_request()	549
4.17.1.3	ehsm_hash_hmac()	549
4.17.1.4	ehsm_mac_request()	549
4.17.1.5	ehsm_random_generate()	550
4.17.1.6	ehsm_signature_gen_vry()	550
4.18	eHSM_If_Asr_ErrCode_Ip.c File Reference	551
4.18.1	Function Documentation	551
4.18.1.1	ehsm_autosar_convert_ret_code()	551
4.19	eHSM_If_Asr_ErrCode_Ip.h File Reference	552
4.19.1	Function Documentation	552
4.19.1.1	ehsm_autosar_convert_ret_code()	552
4.20	eHSM_If_Asr_Ip.h File Reference	552
4.20.1	Function Documentation	553
4.20.1.1	ehsm_certificate_parse()	554
4.20.1.2	ehsm_certificate_verify()	554
4.20.1.3	ehsm_get_version()	555
4.20.1.4	ehsm_init()	555
4.20.1.5	ehsm_job_cancel()	556
4.20.1.6	ehsm_job_submit()	556
4.20.1.7	ehsm_key_copy()	557
4.20.1.8	ehsm_key_derive()	558
4.20.1.9	ehsm_key_element_copy()	559
4.20.1.10	ehsm_key_element_copy_partial()	559
4.20.1.11	ehsm_key_element_get()	560
4.20.1.12	ehsm_key_element_ids_get()	561
4.20.1.13	ehsm_key_element_set()	562
4.20.1.14	ehsm_key_exchange_calcpubval()	563
4.20.1.15	ehsm_key_exchange_calcsecret()	563
4.20.1.16	ehsm_key_generate()	564
4.20.1.17	ehsm_key_set_valid()	564
4.20.1.18	ehsm_remove_key_extend()	565

4.21 eHSM_If_Asr_Job_Ip.c File Reference	566
4.21.1 Function Documentation	566
4.21.1.1 ehsm_init()	566
4.21.1.2 ehsm_job_cancel()	567
4.21.1.3 ehsm_job_submit()	567
4.22 eHSM_If_Asr_Job_Ip.h File Reference	568
4.22.1 Macro Definition Documentation	568
4.22.1.1 Crypto_GetJobKey	568
4.22.1.2 Crypto_GetKeyIdx	569
4.23 eHSM_If_Asr_Key_Ip.c File Reference	569
4.24 eHSM_If_Asr_Key_Ip.h File Reference	569
4.24.1 Function Documentation	569
4.24.1.1 ehsm_calcpubval_with_job()	569
4.24.1.2 ehsm_calcsecret_with_job()	570
4.24.1.3 ehsm_get_ehsm_key_type()	570
4.24.1.4 ehsm_key_element_type_get_ex()	570
4.24.1.5 ehsm_key_is_valid()	570
4.24.1.6 ehsm_key_mgr_init()	570
4.24.1.7 ehsm_keyderi_with_job()	570
4.24.1.8 ehsm_keyexport_with_job()	571
4.24.1.9 ehsm_keygen_with_job()	571
4.24.1.10 ehsm_keyimport_with_job()	571
4.24.1.11 ehsm_keyremove_with_job()	571
4.24.1.12 ehsm_keysetvalid_with_job()	571
4.25 eHSM_If_Asr_KeyCfg_Ip.c File Reference	571
4.25.1 Function Documentation	572
4.25.1.1 ehsm_get_crypto_driver_object()	572
4.25.1.2 ehsm_get_crypto_ke_info()	572
4.25.1.3 ehsm_printf_ke_size()	572
4.25.2 Variable Documentation	573
4.25.2.1 certificate_key_elements	573

4.25.2.2	cipher_key_elements	573
4.25.2.3	copy_key_elements	573
4.25.2.4	crypto_keys	574
4.25.2.5	CryptoDriverObjects	574
4.25.2.6	derive_key_elements	574
4.25.2.7	exchange_key_elements	574
4.25.2.8	export_key_elements	574
4.25.2.9	generate_key_elements	575
4.25.2.10	Hash_CryptoPrimitives	575
4.25.2.11	import_key_elements	575
4.25.2.12	Key_CryptoPrimitives	575
4.25.2.13	key_remove_elements	576
4.25.2.14	key_status_elements	576
4.25.2.15	mac_key_elements	577
4.25.2.16	Pke_CryptoPrimitives	577
4.25.2.17	she_key_elements	577
4.25.2.18	she_plain_key_elements	578
4.25.2.19	signature_key_elements	578
4.25.2.20	Ske_CryptoPrimitives	579
4.25.2.21	Trng_CryptoPrimitives	579
4.26	eHSM_If_Asr_KeyCfg_Ip.h File Reference	580
4.26.1	Macro Definition Documentation	582
4.26.1.1	CRYPTO_CERTIFICATE_KEY_NUM	582
4.26.1.2	CRYPTO_EVITA_CERTIFICATE_KEY_ELEMENT_SIZE	582
4.26.1.3	CRYPTO_EVITA_CIPHER_KEY_ELEMENT_SIZE	583
4.26.1.4	CRYPTO_EVITA_CIPHER_KEY_NUM	583
4.26.1.5	CRYPTO_EVITA_COPY_KEY_ELEMENT_SIZE	583
4.26.1.6	CRYPTO_EVITA_DERIVE_KEY_ELEMENT_SIZE	583
4.26.1.7	CRYPTO_EVITA_DERIVE_KEY_NUM	584
4.26.1.8	CRYPTO_EVITA_EXCHANGE_KEY_ELEMENT_SIZE	584
4.26.1.9	CRYPTO_EVITA_EXCHANGE_KEY_NUM	584

4.26.1.10 CRYPTO_EVITA_EXPORT_KEY_ELEMENT_SIZE	584
4.26.1.11 CRYPTO_EVITA_EXPORT_KEY_NUM	584
4.26.1.12 CRYPTO_EVITA_GENERATE_KEY_ELEMENT_SIZE	585
4.26.1.13 CRYPTO_EVITA_GENERATE_KEY_NUM	585
4.26.1.14 CRYPTO_EVITA_IMPORT_KEY_ELEMENT_SIZE	585
4.26.1.15 CRYPTO_EVITA_IMPORT_KEY_NUM	585
4.26.1.16 CRYPTO_EVITA_KEY_COPY_KEY_NUM	585
4.26.1.17 CRYPTO_EVITA_KEY_STATUS_ELEMENT_SIZE	586
4.26.1.18 CRYPTO_EVITA_KEY_STATUS_NUM	586
4.26.1.19 CRYPTO_EVITA_MAC_KEY_ELEMENT_SIZE	586
4.26.1.20 CRYPTO_EVITA_REMOVE_KEY_ELEMENT_SIZE	586
4.26.1.21 CRYPTO_EVITA_REMOVE_KEY_NUM	586
4.26.1.22 CRYPTO_EVITA_SIGNATURE_KEY_ELEMENT_SIZE	586
4.26.1.23 CRYPTO_EVITA_SIGNATURE_KEY_NUM	587
4.26.1.24 CRYPTO_INVALID_KEY_ID	587
4.26.1.25 CRYPTO_KE_AEAD_TAG_SIZE	587
4.26.1.26 CRYPTO_KE_CIPHER_2NDKEY_SIZE_SIZE	587
4.26.1.27 CRYPTO_KE_CIPHER_CIPHER_ALG_SIZE	587
4.26.1.28 CRYPTO_KE_CIPHER_CURVE_ID_SIZE	587
4.26.1.29 CRYPTO_KE_CIPHER_IV_SIZE	588
4.26.1.30 CRYPTO_KE_CIPHER_KDF_ALG_SIZE	588
4.26.1.31 CRYPTO_KE_CIPHER_KEY_SIZE	588
4.26.1.32 CRYPTO_KE_CIPHER_MAC_ALG_SIZE	588
4.26.1.33 CRYPTO_KE_CIPHER_MAC_SIZE_SIZE	588
4.26.1.34 CRYPTO_KE_CIPHER_PROOF_SIZE	588
4.26.1.35 CRYPTO_KE_COPY_KEY_PARENT_KEY_SIZE	589
4.26.1.36 CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE_SIZE	589
4.26.1.37 CRYPTO_KE_EXPORT_KEY_BLOB_SIZE	589
4.26.1.38 CRYPTO_KE_EXPORT_KEY_SIZE	589
4.26.1.39 CRYPTO_KE_EXT_SHE_KEY_SIZE	589
4.26.1.40 CRYPTO_KE_IMPORT_KEY_SIZE	589

4.26.1.41 CRYPTO_KE_IMPORTED_KEY_KEYHANDLE_SIZE	590
4.26.1.42 CRYPTO_KE_KEY_MATERIAL_SIZE	590
4.26.1.43 CRYPTO_KE_KEY_STATUS_BLOB_SIZE	590
4.26.1.44 CRYPTO_KE_KEY_STATUS_SIZE	590
4.26.1.45 CRYPTO_KE_KEYDERIVATION_ALGORITHM_SIZE	590
4.26.1.46 CRYPTO_KE_KEYDERIVATION_ITERATIONS_SIZE	590
4.26.1.47 CRYPTO_KE_KEYDERIVATION_KEY_SIZE	591
4.26.1.48 CRYPTO_KE_KEYDERIVATION_KEYHANDLE_SIZE	591
4.26.1.49 CRYPTO_KE_KEYDERIVATION_PASSWD_SIZE	591
4.26.1.50 CRYPTO_KE_KEYDERIVATION_SALT_SIZE	591
4.26.1.51 CRYPTO_KE_KEYDERIVATION_TYPE_SIZE	591
4.26.1.52 CRYPTO_KE_KEYEXCHANGE_ALGORITHM_SIZE	591
4.26.1.53 CRYPTO_KE_KEYEXCHANGE_BASE_SIZE	592
4.26.1.54 CRYPTO_KE_KEYEXCHANGE_KEYINFO_SIZE	592
4.26.1.55 CRYPTO_KE_KEYEXCHANGE_OWNPUKEY_SIZE	592
4.26.1.56 CRYPTO_KE_KEYEXCHANGE_PEERPUKEY_SIZE	592
4.26.1.57 CRYPTO_KE_KEYEXCHANGE_PRIVKEY_SIZE	592
4.26.1.58 CRYPTO_KE_KEYEXCHANGE_PUBTYPE_SIZE	592
4.26.1.59 CRYPTO_KE_KEYEXCHANGE_SM2_LOCALTMPKINFO_SIZE	593
4.26.1.60 CRYPTO_KE_KEYEXCHANGE_SM2_PEERTMPPUBK_SIZE	593
4.26.1.61 CRYPTO_KE_KEYEXCHANGE_SM2_ROLE_SIZE	593
4.26.1.62 CRYPTO_KE_KEYEXCHANGE_SM2_S1_S2_VALUE_SIZE	593
4.26.1.63 CRYPTO_KE_KEYEXCHANGE_SM2_SA_SB_VALUE_SIZE	593
4.26.1.64 CRYPTO_KE_KEYGENERATE_ALGORITHM_SIZE	593
4.26.1.65 CRYPTO_KE_KEYGENERATE_DH_KEY_INFO_SIZE	594
4.26.1.66 CRYPTO_KE_KEYGENERATE_KEY_SIZE	594
4.26.1.67 CRYPTO_KE_KEYGENERATE_KEYINFO_SIZE	594
4.26.1.68 CRYPTO_KE_KEYGENERATE_SEED_SIZE	594
4.26.1.69 CRYPTO_KE_MAC_KEY_SIZE	594
4.26.1.70 CRYPTO_KE_MAC_PROOF_SIZE	594
4.26.1.71 CRYPTO_KE_RANDOM_ALGORITHM_SIZE	595

4.26.1.72 CRYPTO_KE_RANDOM_SEED_STATE_SIZE	595
4.26.1.73 CRYPTO_KE_REMOVE_KEY_SIZE	595
4.26.1.74 CRYPTO_KE_SHE_PLAIN_KEY_SIZE	595
4.26.1.75 CRYPTO_KE_SIGNATURE_KEY_SIZE	595
4.26.1.76 CRYPTO_KE_SIGNATURE_RSA_CRT_MODE_SIZE	595
4.26.1.77 CRYPTO_KE_SIGNATURE_TIMESTAMPED_SIZE	596
4.26.1.78 CRYPTO_KEY10_EVITA_IMPORT_KEY	596
4.26.1.79 CRYPTO_KEY11_EVITA_EXPORT_KEY	596
4.26.1.80 CRYPTO_KEY12_EVITA_REMOVE_KEY	596
4.26.1.81 CRYPTO_KEY13_EVITA_KEY_STATUS	596
4.26.1.82 CRYPTO_KEY14_EVITA_KEY_COPY_KEY	596
4.26.1.83 CRYPTO_KEY15_EVITA_KEY_COPY_KEY	597
4.26.1.84 CRYPTO_KEY16_CERTIFICATE_KEY	597
4.26.1.85 CRYPTO_KEY17_CERTIFICATE_KEY	597
4.26.1.86 CRYPTO_KEY18_MAC_KEY	597
4.26.1.87 CRYPTO_KEY1_SHE_KEY	597
4.26.1.88 CRYPTO_KEY2_SHE_PLAIN_KEY	597
4.26.1.89 CRYPTO_KEY3_EVITA_SIGNATURE_KEY	598
4.26.1.90 CRYPTO_KEY4_EVITA_CIPHER_KEY	598
4.26.1.91 CRYPTO_KEY5_EVITA_EXCHANGE_KEY	598
4.26.1.92 CRYPTO_KEY6_EVITA_EXCHANGE_KEY	598
4.26.1.93 CRYPTO_KEY7_EVITA_DERIVE_KEY	598
4.26.1.94 CRYPTO_KEY8_EVITA_DERIVE_KEY	598
4.26.1.95 CRYPTO_KEY9_EVITA_GENERATE_KEY	599
4.26.1.96 CRYPTO_MAC_KEY_NUM	599
4.26.1.97 CRYPTO_MAX_KEY_NUM	599
4.26.1.98 CRYPTO_SHE_KEY_ELEMENT_SIZE	599
4.26.1.99 CRYPTO_SHE_KEY_NUM	599
4.26.1.100CRYPTO_SHE_PLAIN_KEY_ELEMENT_SIZE	600
4.26.1.101CRYPTO_SHE_PLAIN_KEY_NUM	600
4.26.1.102CRYPTO_KE_KEYEXCHANGE_SHAREDVALUE_SIZE	600

4.26.1.103	KEY_ELEMENT_NUM	600
4.26.1.104	KEY_ELEMENT_RESERVER	601
4.26.1.105	KEY_ELEMENT_VALUE_BUFFER_RESERVER	601
4.26.1.106	KEY_ELEMENT_VALUE_BUFFER_SIZE	601
4.26.2	Enumeration Type Documentation	601
4.26.2.1	crypto_object_type_e	601
4.26.3	Function Documentation	602
4.26.3.1	ehsm_get_crypto_driver_object()	602
4.26.3.2	ehsm_get_crypto_ke_info()	602
4.26.3.3	ehsm_printf_ke_size()	602
4.27	eHSM_If_Asr_Types_lp.h File Reference	602
4.27.1	Macro Definition Documentation	605
4.27.1.1	CRYPTO_INDEX_M1_U32	605
4.27.1.2	CRYPTO_INDEX_M2_U32	605
4.27.1.3	CRYPTO_INDEX_M3_U32	605
4.27.1.4	CRYPTO_INDEX_M4_U32	606
4.27.1.5	CRYPTO_INDEX_M5_U32	606
4.27.1.6	CRYPTO_KE_CERTIFICATE_SIGNEDDATA	606
4.27.1.7	CRYPTO_KE_CIPHER_CIPHER_ALG	606
4.27.1.8	CRYPTO_KE_CIPHER_CURVE_ID	606
4.27.1.9	CRYPTO_KE_CIPHER_KDF_ALG	606
4.27.1.10	CRYPTO_KE_CIPHER_MAC_ALG	607
4.27.1.11	CRYPTO_KE_CIPHER_MAC_SIZE	607
4.27.1.12	CRYPTO_KE_CIPHER_TAG_SIZE	607
4.27.1.13	CRYPTO_KE_COPY_KEY_PARENT_KEY	607
4.27.1.14	CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE	607
4.27.1.15	CRYPTO_KE_EXPORT_KEY	607
4.27.1.16	CRYPTO_KE_EXPORT_KEY_BLOB	608
4.27.1.17	CRYPTO_KE_EXT_SHE_KEY	608
4.27.1.18	CRYPTO_KE_IMPORT_KEY	608
4.27.1.19	CRYPTO_KE_IMPORTED_KEY_KEYHANDLE	608

4.27.1.20 CRYPTO_KE_KEY_HANDLE	608
4.27.1.21 CRYPTO_KE_KEY_MATERIAL	608
4.27.1.22 CRYPTO_KE_KEY_STATUS	609
4.27.1.23 CRYPTO_KE_KEY_STATUS_BLOB	609
4.27.1.24 CRYPTO_KE_KEYDERIVATION_KEY	609
4.27.1.25 CRYPTO_KE_KEYDERIVATION_KEYHANDLE [1/2]	609
4.27.1.26 CRYPTO_KE_KEYDERIVATION_KEYHANDLE [2/2]	609
4.27.1.27 CRYPTO_KE_KEYDERIVATION_TYPE	609
4.27.1.28 CRYPTO_KE_KEYEXCHANGE_KEYINFO	610
4.27.1.29 CRYPTO_KE_KEYEXCHANGE_PEERPUBKEY	610
4.27.1.30 CRYPTO_KE_KEYEXCHANGE_PUBTYPE	610
4.27.1.31 CRYPTO_KE_KEYEXCHANGE_SM2_LOCALTMPKINFO	610
4.27.1.32 CRYPTO_KE_KEYEXCHANGE_SM2_PEERTMPPUBK	610
4.27.1.33 CRYPTO_KE_KEYEXCHANGE_SM2_ROLE	610
4.27.1.34 CRYPTO_KE_KEYEXCHANGE_SM2_S1_S2_VALUE	611
4.27.1.35 CRYPTO_KE_KEYEXCHANGE_SM2_SA_SB_VALUE	611
4.27.1.36 CRYPTO_KE_KEYGENERATE_DH_KEY_INFO	611
4.27.1.37 CRYPTO_KE_KEYGENERATE_KEYINFO	611
4.27.1.38 CRYPTO_KE_REMOVE_KEY	611
4.27.1.39 CRYPTO_KE_SIGNATURE_RSA_CRT_MODE	611
4.27.1.40 CRYPTO_KE_SIGNATURE_TIMESTAMPED	612
4.27.1.41 CRYPTO_KEY_ATTR_ALLOW_PARTIAL_ACCESS	612
4.27.1.42 CRYPTO_KEY_ATTR_READ_ACCESS	612
4.27.1.43 CRYPTO_KEY_ATTR_WRITE_ACCESS	612
4.27.1.44 CRYPTO_M1_SIZE_U32	612
4.27.1.45 CRYPTO_M2_SIZE_U32	612
4.27.1.46 CRYPTO_M3_SIZE_U32	613
4.27.1.47 CRYPTO_M4_SIZE_U32	613
4.27.1.48 CRYPTO_M5_SIZE_U32	613
4.27.1.49 CRYPTO_MAX_AUTH_VALUE_SIZE	613
4.27.1.50 CRYPTO_MAX_KEY_ELEMENTS_OF_KEY_TYPE	613

4.27.1.51 CRYPTO_MAX_KEY_FLAG_ELEMENT	613
4.27.1.52 CRYPTO_SHE_MAC_KEY_ID [1/2]	614
4.27.1.53 CRYPTO_SHE_MAC_KEY_ID [2/2]	614
4.27.1.54 CRYPTO_SHE_MAC_KEY_NUM	614
4.27.1.55 CRYPTO_SHE_SIZE_IN_U32	614
4.27.1.56 CRYPTO_SHE_SIZE_OUT_U32	614
4.27.1.57 EVITA_KEY_USE_FLAG_DECRYPT	614
4.27.1.58 EVITA_KEY_USE_FLAG_DHKE	615
4.27.1.59 EVITA_KEY_USE_FLAG_ENCRYPT	615
4.27.1.60 EVITA_KEY_USE_FLAG_REMOVE	615
4.27.1.61 EVITA_KEY_USE_FLAG_SECUREBOOT	615
4.27.1.62 EVITA_KEY_USE_FLAG_SECURESTORAGE	615
4.27.1.63 EVITA_KEY_USE_FLAG_SIGN	615
4.27.1.64 EVITA_KEY_USE_FLAG_TIMESTAMP	616
4.27.1.65 EVITA_KEY_USE_FLAG_TRANSPORT	616
4.27.1.66 EVITA_KEY_USE_FLAG_UTCSYNC	616
4.27.1.67 EVITA_KEY_USE_FLAG_VERIFY	616
4.27.2 Typedef Documentation	616
4.27.2.1 crypto_copy_key_dh_key_info_st	616
4.27.2.2 crypto_copy_key_info_st	616
4.27.2.3 crypto_create_evita_key_info_st	617
4.27.2.4 crypto_evita_key_info_st	617
4.27.2.5 crypto_exported_key_st	617
4.27.2.6 crypto_import_evita_key_info_st	617
4.27.2.7 crypto_key_derive_info_st	617
4.27.2.8 crypto_key_export_info_st	617
4.27.2.9 crypto_key_status_info_st	617
4.27.2.10 crypto_she_key_st	617
4.27.2.11 ehsm_crypto_key_st	617
4.27.3 Enumeration Type Documentation	617
4.27.3.1 crypto_key_type_e	617

4.27.3.2	ehsm_key_type_e	618
4.27.3.3	ehsm_key_use_state_e	618
4.27.3.4	key_storage_type_e	619
4.27.4	Function Documentation	619
4.27.4.1	CryIf_CallbackNotification()	619
4.28	eHSM_If_Evita_AsymCper_Ip.c File Reference	619
4.28.1	Function Documentation	620
4.28.1.1	Sign_Finish()	620
4.28.1.2	Sign_Init()	620
4.28.1.3	Sign_Update()	621
4.28.1.4	Verify_Finish()	622
4.28.1.5	Verify_Init()	622
4.28.1.6	Verify_Update()	623
4.29	eHSM_If_Evita_Counter_Ip.c File Reference	624
4.30	eHSM_If_Evita_ErrCode_Ip.c File Reference	624
4.30.1	Function Documentation	624
4.30.1.1	ehsm_evita_convert_ret_code()	624
4.30.1.2	Evita_Check_Authorization_Code()	625
4.30.1.3	Evita_Check_Key_Handle()	625
4.31	eHSM_If_Evita_ErrCode_Ip.h File Reference	626
4.31.1	Function Documentation	626
4.31.1.1	ehsm_evita_convert_ret_code()	626
4.31.1.2	Evita_Check_Authorization_Code()	627
4.31.1.3	Evita_Check_Key_Handle()	627
4.32	eHSM_If_Evita_Hash_Ip.c File Reference	628
4.32.1	Macro Definition Documentation	629
4.32.1.1	HASH_SIZE	629
4.32.1.2	X [1/2]	629
4.32.1.3	X [2/2]	629
4.32.2	Function Documentation	630
4.32.2.1	Hash_Finish()	630

4.32.2.2 Hash_Init()	630
4.32.2.3 Hash_Update()	631
4.33 eHSM_If_Evita_Ip.h File Reference	632
4.33.1 Function Documentation	635
4.33.1.1 Aead_Finish()	635
4.33.1.2 Aead_Init()	636
4.33.1.3 Aead_Process()	637
4.33.1.4 Check_Time_Stamp()	637
4.33.1.5 Cipher_Finish()	638
4.33.1.6 Cipher_Init()	639
4.33.1.7 Cipher_Process()	640
4.33.1.8 Create_Counter()	640
4.33.1.9 Create_Derived_Key()	641
4.33.1.10 Create_Dh_Key()	642
4.33.1.11 Create_Random_Dh_Key_Pair()	643
4.33.1.12 Create_Random_Key()	644
4.33.1.13 Create_Time_Stamp()	645
4.33.1.14 Delete_Counter()	646
4.33.1.15 ehsm_get_pub_from_priv()	646
4.33.1.16 ehsm_self_test()	647
4.33.1.17 Get_Tick_Count()	648
4.33.1.18 Get_Time_Sync_Challenge()	648
4.33.1.19 Get_UTC_Time()	648
4.33.1.20 Hash_Finish()	649
4.33.1.21 Hash_Init()	649
4.33.1.22 Hash_Update()	650
4.33.1.23 Increment_Counter()	651
4.33.1.24 Key_Export()	651
4.33.1.25 Key_Import()	653
4.33.1.26 Key_Remove()	654
4.33.1.27 Key_Status()	654

4.33.1.28 MAC_Finish()	655
4.33.1.29 MAC_Init()	656
4.33.1.30 MAC_Update()	657
4.33.1.31 Module_Status()	657
4.33.1.32 Read_Counter()	658
4.33.1.33 RNG_Get_Random()	658
4.33.1.34 Set_UTC_Time()	659
4.33.1.35 Sign_Finish()	660
4.33.1.36 Sign_Init()	660
4.33.1.37 Sign_Update()	661
4.33.1.38 Verify_Finish()	662
4.33.1.39 Verify_Init()	662
4.33.1.40 Verify_Update()	663
4.34 eHSM_If_Evita_Key_Ip.c File Reference	664
4.34.1 Function Documentation	665
4.34.1.1 _translate_bool_array_to_use_flags()	665
4.34.1.2 Create_Derived_Key()	665
4.34.1.3 Create_Dh_Key()	666
4.34.1.4 Create_Random_Dh_Key_Pair()	667
4.34.1.5 Create_Random_Key()	668
4.34.1.6 ehsm_get_pub_from_priv()	669
4.34.1.7 Key_Export()	670
4.34.1.8 Key_Import()	671
4.34.1.9 Key_Remove()	672
4.34.1.10 Key_Status()	673
4.34.1.11 Module_Status()	674
4.35 eHSM_If_Evita_Rng_Ip.c File Reference	674
4.35.1 Function Documentation	675
4.35.1.1 RNG_Get_Random()	675
4.36 eHSM_If_Evita_SymCper_Ip.c File Reference	676
4.36.1 Function Documentation	677

4.36.1.1	Aead_Finish()	677
4.36.1.2	Aead_Init()	677
4.36.1.3	Aead_Process()	678
4.36.1.4	Cipher_Finish()	679
4.36.1.5	Cipher_Init()	680
4.36.1.6	Cipher_Process()	681
4.36.1.7	MAC_Finish()	681
4.36.1.8	MAC_Init()	683
4.36.1.9	MAC_Update()	684
4.37	eHSM_If_Evita_Timer_Ip.c File Reference	684
4.38	eHSM_If_Evita_Types_Ip.h File Reference	685
4.38.1	Macro Definition Documentation	688
4.38.1.1	ALIGN_BYTE	689
4.38.1.2	EHSM_CONTEXT_SIZE	689
4.38.1.3	EHSM_ECC_KEY_PAIR_MAX_SIZE	689
4.38.1.4	EHSM_EVITA_KEY_NUM	689
4.38.1.5	EHSM_KEY_AUTH_VALUE_MAX_SIZE	689
4.38.1.6	EHSM_KEY_DATA_MAX_SIZE [1/3]	689
4.38.1.7	EHSM_KEY_DATA_MAX_SIZE [2/3]	690
4.38.1.8	EHSM_KEY_DATA_MAX_SIZE [3/3]	690
4.38.1.9	EHSM_KEY_HEAD_SIZE	690
4.38.1.10	EHSM_KEY_SIZE_INFO_MAX_LEN	690
4.38.1.11	EHSM_RSA_DH_KEY_PAIR_MAX_SIZE	690
4.38.1.12	EHSM_RSA_KEY_PAIR_MAX_SIZE	690
4.38.1.13	EHSM_SM2_SM2_KEY_MAX_SIZE	691
4.38.1.14	EHSM_SYM_KEY_PAIR_MAX_SIZE	691
4.38.1.15	EVITA_ALGORITHM_ERROR	691
4.38.1.16	EVITA_ALL_COUNTERS_OCCUPIED	691
4.38.1.17	EVITA_ALL_KEY_SPACE_OCCUPIED	691
4.38.1.18	EVITA_ALL_SESSIONS_OCCUPIED	691
4.38.1.19	EVITA_AUTH_TYPE_NONE	692

4.38.1.20 EVITA_AUTH_TYPE_PASSWD	692
4.38.1.21 EVITA_AUTHORIZATION_FAILED	692
4.38.1.22 EVITA_CLOCK_NOT_SYNCHRONIZED	692
4.38.1.23 EVITA_GENERAL_ERROR	692
4.38.1.24 EVITA_HASH_BUF_SIZE	692
4.38.1.25 EVITA_INVALID_COUNTER_INCREMENTATION	693
4.38.1.26 EVITA_INVALID_KEY_FLAG	693
4.38.1.27 EVITA_INVALID_KEY_SIZE	693
4.38.1.28 EVITA_INVALID_MSG_SIZE	693
4.38.1.29 EVITA_INVALID_TIME_STAMP	693
4.38.1.30 EVITA_INVALID_UTC_TIME	693
4.38.1.31 EVITA_KEY_DERIVE_KDFX963	694
4.38.1.32 EVITA_KEY_DERIVE_PBKDF2	694
4.38.1.33 EVITA_KEY_MAX_SIZE	694
4.38.1.34 EVITA_KEY_SIGNATRUE_SIZE	694
4.38.1.35 EVITA_KEY_TRNSP_EXT	694
4.38.1.36 EVITA_KEY_TRNSP_INI	694
4.38.1.37 EVITA_KEY_TRNSP_MIG	695
4.38.1.38 EVITA_KEY_TRNSP_OEM	695
4.38.1.39 EVITA_KEY_USE_FLAG_DECRYPT	695
4.38.1.40 EVITA_KEY_USE_FLAG_ENCRYPT	695
4.38.1.41 EVITA_KEY_USE_FLAG_KEYCREATION	695
4.38.1.42 EVITA_KEY_USE_FLAG_REMOVE	695
4.38.1.43 EVITA_KEY_USE_FLAG_SECUREBOOT	696
4.38.1.44 EVITA_KEY_USE_FLAG_SECURESTORAGE	696
4.38.1.45 EVITA_KEY_USE_FLAG_SIGN	696
4.38.1.46 EVITA_KEY_USE_FLAG_TIMESTAMP	696
4.38.1.47 EVITA_KEY_USE_FLAG_TRANSPORT	696
4.38.1.48 EVITA_KEY_USE_FLAG_UTCSYNC	696
4.38.1.49 EVITA_KEY_USE_FLAG_VERIFY	697
4.38.1.50 EVITA_MAC_BUF_SIZE	697

4.38.1.51 EVITA_MAC_LENGTH_OVERSIZE	697
4.38.1.52 EVITA_MAX_CHUNK_SIZE	697
4.38.1.53 EVITA_MAX_OTP_KEY_SIZE	697
4.38.1.54 EVITA_MAX_RANDOM_KEY_SIZE	697
4.38.1.55 EVITA_OK	698
4.38.1.56 EVITA_OTP_ASYM_KEY_SIZE	698
4.38.1.57 EVITA_OTP_PRIVKEY_SIZE	698
4.38.1.58 EVITA_OTP_PUBKEY_SIZE	698
4.38.1.59 EVITA_OTP_SYM_KEY_SIZE	698
4.38.1.60 EVITA_PRNG_REQUEST_OVERSIZE	698
4.38.1.61 EVITA_REMOVE_IMPOSSIBLE	699
4.38.1.62 EVITA_SALT_VALUE_MAX_SIZE	699
4.38.1.63 EVITA_SIGNATURE_BUF_SIZE	699
4.38.1.64 EVITA_STATUS_TYPE_NOT_AVAILABLE	699
4.38.1.65 EVITA_TEST_CASE_FAILED	699
4.38.1.66 EVITA_TEST_CASE_NOT_AVAILABLE	699
4.38.1.67 EVITA_TRANSPORT_IMPOSSIBLE	700
4.38.1.68 EVITA_TRNG_SEED_FAILURE	700
4.38.1.69 EVITA_UNKNOWN_COUNTER_ID	700
4.38.1.70 EVITA_UTC_CHALLENGE_EXPIRED	700
4.38.1.71 EVITA_UTC_SYNCHRONIZATION_FAILED	700
4.38.1.72 EVITA_WRONG_AUTHORIZATION	700
4.38.1.73 EVITA_WRONG_CERT_KEY_HANDLE	701
4.38.1.74 EVITA_WRONG_CHUNK_SIZE	701
4.38.1.75 EVITA_WRONG_ECR_INDEX	701
4.38.1.76 EVITA_WRONG_IV	701
4.38.1.77 EVITA_WRONG_KEY_COMBINATION	701
4.38.1.78 EVITA_WRONG_KEY_HANDLE	701
4.38.1.79 EVITA_WRONG_REMOTE_KEY_HANDLE	702
4.38.1.80 EVITA_WRONG_SESSION_HANDLE	702
4.38.1.81 RNG_REQUEST_MAX	702

4.38.1.82 SKE_CTX_BUF_SIZE	702
4.38.2 Typedef Documentation	702
4.38.2.1 ehsm_counter_value_st	702
4.38.2.2 ehsm_dh_param_size_info_st	702
4.38.2.3 ehsm_dh_param_st	703
4.38.2.4 ehsm_dh_prikey_st	703
4.38.2.5 ehsm_dh_pubkey_st	703
4.38.2.6 ehsm_ecc_key_size_st	703
4.38.2.7 ehsm_ecc_pubkey_st	703
4.38.2.8 ehsm_key_attr_data_st	703
4.38.2.9 ehsm_key_signatrue_st	703
4.38.2.10 ehsm_prikey_data_st	703
4.38.2.11 ehsm_pubkey_data_st	704
4.38.2.12 ehsm_rsa_ctr_st	704
4.38.2.13 ehsm_rsa_dh_key_size_st	704
4.38.2.14 ehsm_rsa_key_size_st	704
4.38.2.15 ehsm_rsa_pubkey_st	704
4.38.2.16 ehsm_sym_key_size_st	704
4.38.2.17 ehsm_tick_value_st	704
4.38.2.18 ehsm_utc_time_t	704
4.38.2.19 hash_hmac_st	705
4.38.2.20 mac_st	705
4.38.2.21 signature_st	705
4.38.2.22 time_stamp_st	705
4.38.3 Enumeration Type Documentation	705
4.38.3.1 cipher_mode_e	705
4.38.3.2 ehsm_evita_key_handle_e	705
4.38.3.3 hash_mode_e	706
4.38.3.4 mac_mode_e	706
4.38.3.5 operation_mode_e	706
4.38.3.6 padding_scheme_e	707

4.38.3.7	session_status_e	707
4.38.3.8	storage_key_type_e	707
4.38.4	Function Documentation	708
4.38.4.1	ALIGN_BYTE()	708
4.38.5	Variable Documentation	708
4.38.5.1	activeUseFlag	708
4.38.5.2	algo_id	708
4.38.5.3	attr	708
4.38.5.4	auth_sign_data	709
4.38.5.5	auth_size	709
4.38.5.6	auth_value	709
4.38.5.7	cert_data	709
4.38.5.8	cert_size	709
4.38.5.9	dh_pubkey_bytes_size	709
4.38.5.10	evita_internal_key	710
4.38.5.11	key	710
4.38.5.12	key_data	710
4.38.5.13	key_handle	710
4.38.5.14	key_pub_data	710
4.38.5.15	key_sign_data	710
4.38.5.16	key_signatrue	711
4.38.5.17	key_size_info	711
4.38.5.18	key_usage	711
4.38.5.19	keyId	711
4.38.5.20	keyIdSize	711
4.38.5.21	mem_location	711
4.38.5.22	prikey	712
4.38.5.23	prikey_enc_size	712
4.38.5.24	pubkey	712
4.38.5.25	reserved	712
4.38.5.26	rsa_e_bytes_size	712

4.38.5.27 size_info	712
4.38.5.28 valid_util	712
4.39 eHSM_If_Ext_Ip.h File Reference	713
4.39.1 Enumeration Type Documentation	715
4.39.1.1 ehsm_power_mode_e	715
4.39.2 Function Documentation	715
4.39.2.1 ehsm_change_controlfield()	715
4.39.2.2 ehsm_change_lifecycle()	716
4.39.2.3 ehsm_close_debug()	716
4.39.2.4 ehsm_debug_auth()	717
4.39.2.5 ehsm_encrypt_key()	718
4.39.2.6 ehsm_get_challenge()	718
4.39.2.7 ehsm_get_emu_status()	719
4.39.2.8 ehsm_get_random_key()	719
4.39.2.9 ehsm_hsm_fw_upgrade_finish()	720
4.39.2.10 ehsm_hsm_fw_upgrade_init()	721
4.39.2.11 ehsm_hsm_fw_upgrade_update()	721
4.39.2.12 ehsm_hsm_fw_upgrade_verify_finish()	722
4.39.2.13 ehsm_hsm_fw_upgrade_verify_init()	723
4.39.2.14 ehsm_hsm_fw_upgrade_verify_update()	723
4.39.2.15 ehsm_low_power()	724
4.39.2.16 ehsm_read_otp_data()	725
4.39.2.17 ehsm_secure_boot()	725
4.39.2.18 ehsm_set_uart_baudrate()	726
4.39.2.19 ehsm_sm9_cipher()	726
4.39.2.20 ehsm_sm9_exchg_key()	727
4.39.2.21 ehsm_sm9_exckey_gen_tmpkey()	728
4.39.2.22 ehsm_sm9_export_key()	729
4.39.2.23 ehsm_sm9_gen_mastpubkey_from_mastprivkey()	730
4.39.2.24 ehsm_sm9_gen_tmppubkey_from_tmpprivkey()	730
4.39.2.25 ehsm_sm9_generate_master_key()	732

4.39.2.26 ehsm_sm9_generate_priv_key()	732
4.39.2.27 ehsm_sm9_import_key()	733
4.39.2.28 ehsm_sm9_remove_key()	734
4.39.2.29 ehsm_sm9_sign()	734
4.39.2.30 ehsm_sm9_unwrap_key()	735
4.39.2.31 ehsm_sm9_wrap_key()	736
4.39.2.32 ehsm_write_otp_data()	737
4.40 eHSM_If_Ext_Sm9_Ip.c File Reference	737
4.40.1 Macro Definition Documentation	738
4.40.1.1 EHSM_CRYPT0_V_SM9_MAX_ID_SIZE	738
4.41 eHSM_If_Ext_SysMgr_Ip.c File Reference	738
4.41.1 Macro Definition Documentation	740
4.41.1.1 EHSM_IMAGE_VERIFY_TYPE_FW_UPGRADE	740
4.41.1.2 EHSM_IMAGE_VERIFY_TYPE_SOC_BOOT	740
4.41.1.3 EHSM_IMAGE_VERIFY_TYPE_SOC_UPGRADE	740
4.41.2 Function Documentation	740
4.41.2.1 ehsm_change_controlfield()	740
4.41.2.2 ehsm_change_lifecycle()	741
4.41.2.3 ehsm_close_debug()	741
4.41.2.4 ehsm_debug_auth()	742
4.41.2.5 ehsm_encrypt_key()	743
4.41.2.6 ehsm_get_challenge()	743
4.41.2.7 ehsm_get_emu_status()	744
4.41.2.8 ehsm_get_random_key()	745
4.41.2.9 ehsm_hsm_fw_upgrade_finish()	745
4.41.2.10 ehsm_hsm_fw_upgrade_init()	746
4.41.2.11 ehsm_hsm_fw_upgrade_update()	747
4.41.2.12 ehsm_hsm_fw_upgrade_verify_finish()	747
4.41.2.13 ehsm_hsm_fw_upgrade_verify_init()	748
4.41.2.14 ehsm_hsm_fw_upgrade_verify_update()	749
4.41.2.15 ehsm_low_power()	749

4.41.2.16 ehsm_read_otp_data()	750
4.41.2.17 ehsm_secure_boot()	750
4.41.2.18 ehsm_self_test()	751
4.41.2.19 ehsm_set_uart_baudrate()	751
4.41.2.20 ehsm_write_otp_data()	752
4.41.3 Variable Documentation	753
4.41.3.1 g_ctx	753
4.42 eHSM_If_Ext_Types_lp.h File Reference	753
4.42.1 Macro Definition Documentation	753
4.42.1.1 CONFIG_SM9_ENC_DEFAULT_HID_VALUE	753
4.42.1.2 CONFIG_SM9_EXCHG_DEFAULT_HID_VALUE	754
4.42.1.3 CONFIG_SM9_SIGN_DEFAULT_HID_VALUE	754
4.42.1.4 SM2_PUBLIC_KEY_SIZE	754
4.42.1.5 SM2_S1_S2_SIZE	754
4.42.2 Enumeration Type Documentation	754
4.42.2.1 ehsm_gen_key_alg_e	754
4.43 eHSM_If_She_ErrCode_lp.c File Reference	755
4.43.1 Function Documentation	755
4.43.1.1 ehsm_she_convert_ret_code()	755
4.44 eHSM_If_She_ErrCode_lp.h File Reference	756
4.44.1 Function Documentation	756
4.44.1.1 ehsm_she_convert_ret_code()	756
4.45 eHSM_If_She_lp.c File Reference	756
4.45.1 Macro Definition Documentation	758
4.45.1.1 CONFIG_EHSM_KMGR_V_SHE_BOOT_MAC_K	758
4.45.1.2 CONFIG_EHSM_KMGR_V_SHE_K_MIN	758
4.45.1.3 EHSM_SOC_BOOT_STATUS_FAIL	758
4.45.1.4 EHSM_SOC_BOOT_STATUS_OK	758
4.45.2 Function Documentation	758
4.45.2.1 ehsm_she_cancel()	758
4.45.2.2 she_boot_failure()	759

4.45.2.3	she_boot_ok()	759
4.45.2.4	she_crypto_cbc()	760
4.45.2.5	she_crypto_ecb()	760
4.45.2.6	she_crypto_ecb_extend()	761
4.45.2.7	she_debug()	761
4.45.2.8	she_export_ram_key()	762
4.45.2.9	she_extend_seed()	763
4.45.2.10	she_generate_mac()	763
4.45.2.11	she_get_id()	765
4.45.2.12	she_get_status()	766
4.45.2.13	she_init_rng()	766
4.45.2.14	she_load_key()	767
4.45.2.15	she_load_key_extend()	768
4.45.2.16	she_load_plain_key()	769
4.45.2.17	she_rnd()	769
4.45.2.18	she_secure_boot()	770
4.45.2.19	she_verify_mac()	770
4.46	eHSM_If_She_Ip.h File Reference	771
4.46.1	Macro Definition Documentation	772
4.46.1.1	AES_CTRDRBG	772
4.46.1.2	SM4_CTRDRBG	772
4.46.2	Function Documentation	772
4.46.2.1	ehsm_she_cancel()	773
4.46.2.2	she_boot_failure()	773
4.46.2.3	she_boot_ok()	774
4.46.2.4	she_crypto_cbc()	774
4.46.2.5	she_crypto_ecb()	775
4.46.2.6	she_crypto_ecb_extend()	775
4.46.2.7	she_debug()	776
4.46.2.8	she_export_ram_key()	776
4.46.2.9	she_extend_seed()	777

4.46.2.10 she_generate_mac()	778
4.46.2.11 she_get_id()	778
4.46.2.12 she_get_status()	779
4.46.2.13 she_init_rng()	779
4.46.2.14 she_load_key()	780
4.46.2.15 she_load_key_extend()	781
4.46.2.16 she_load_plain_key()	782
4.46.2.17 she_rnd()	782
4.46.2.18 she_secure_boot()	783
4.46.2.19 she_verify_mac()	784
4.47 eHSM_If_She_Types_Ip.h File Reference	784
4.47.1 Macro Definition Documentation	786
4.47.1.1 EHSM_SHE_KEY_MAC_VERIFY_ONLY	786
4.47.1.2 EHSM_SHE_KEY_MAX_SIZE	786
4.47.1.3 EHSM_SHE_KEY_PROP_BOOT_PRT	786
4.47.1.4 EHSM_SHE_KEY_PROP_DEBUG_PRT	786
4.47.1.5 EHSM_SHE_KEY_PROP_KEY_USAGE	786
4.47.1.6 EHSM_SHE_KEY_PROP_WILDCARD	786
4.47.1.7 EHSM_SHE_KEY_PROP_WR_PRT	787
4.47.1.8 EHSM_SHE_KEY_SIZE	787
4.47.1.9 EHSM_SHE_M1_STD_SIZE	787
4.47.1.10 EHSM_SHE_M4_STD_SIZE	787
4.47.1.11 EHSM_SHE_NVM_KEY_NUM	787
4.47.1.12 EHSM_SHE_OTP_KEY_ATTR_SIZE	787
4.47.1.13 EHSM_SHE_UID_MAX_SIZE	788
4.47.1.14 ERC_BUSY	788
4.47.1.15 ERC_GENERAL_ERROR	788
4.47.1.16 ERC_KEY_EMPTY	788
4.47.1.17 ERC_KEY_INVALID	788
4.47.1.18 ERC_KEY_NOT_AVAILABLE	789
4.47.1.19 ERC_KEY_UPDATE_ERROR	789

4.47.1.20	ERC_KEY_WRITE_PROTECTED	789
4.47.1.21	ERC_MEMORY_FAILURE	789
4.47.1.22	ERC_NO_DEBUGGING	789
4.47.1.23	ERC_NO_ERROR	790
4.47.1.24	ERC_NO_SECURE_BOOT	790
4.47.1.25	ERC_RNG_SEED	790
4.47.1.26	ERC_SEQUENCE_ERROR	790
4.47.2	Typedef Documentation	790
4.47.2.1	ehsm_she_status_type_e	790
4.47.3	Enumeration Type Documentation	790
4.47.3.1	ehsm_she_key_handle_e	790
4.47.3.2	ehsm_she_status_type_	791
4.48	eHSM_IntCfg_Ip.h File Reference	791
4.48.1	Macro Definition Documentation	795
4.48.1.1	CONFIG_EHSM_ARCH_MAIN_HOOK	795
4.48.1.2	CONFIG_EHSM_ARCH_MULTI_CHANNEL	796
4.48.1.3	CONFIG_EHSM_ARCH_OS_NONE	796
4.48.1.4	CONFIG_EHSM_ARCH_SHARE_MEM	796
4.48.1.5	CONFIG_EHSM_ARCH_V_REQ_HASH_MAX_SIZE	796
4.48.1.6	CONFIG_EHSM_ARCH_V_REQ_SKE_MAX_SIZE	796
4.48.1.7	CONFIG_EHSM_COUNTER_AUTO_INCREASE	797
4.48.1.8	CONFIG_EHSM_CRYPT0_AEAD	797
4.48.1.9	CONFIG_EHSM_CRYPT0_ALGOFAM_AES	797
4.48.1.10	CONFIG_EHSM_CRYPT0_ALGOFAM_CTRDRBG	797
4.48.1.11	CONFIG_EHSM_CRYPT0_ALGOFAM_DH	797
4.48.1.12	CONFIG_EHSM_CRYPT0_ALGOFAM_ECC	797
4.48.1.13	CONFIG_EHSM_CRYPT0_ALGOFAM_ED25519	798
4.48.1.14	CONFIG_EHSM_CRYPT0_ALGOFAM_MD5	798
4.48.1.15	CONFIG_EHSM_CRYPT0_ALGOFAM_PBKDF2	798
4.48.1.16	CONFIG_EHSM_CRYPT0_ALGOFAM_RSA	798
4.48.1.17	CONFIG_EHSM_CRYPT0_ALGOFAM_RSA_1024	798

4.48.1.18 CONFIG_EHSM_CRYPTO_ALGOFAM_RSA_1024_CRT	798
4.48.1.19 CONFIG_EHSM_CRYPTO_ALGOFAM_RSA_2048	799
4.48.1.20 CONFIG_EHSM_CRYPTO_ALGOFAM_RSA_2048_CRT	799
4.48.1.21 CONFIG_EHSM_CRYPTO_ALGOFAM_SECP256R1	799
4.48.1.22 CONFIG_EHSM_CRYPTO_ALGOFAM_SECP384R1	799
4.48.1.23 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA1	799
4.48.1.24 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA2	799
4.48.1.25 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA224	800
4.48.1.26 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA256	800
4.48.1.27 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA384	800
4.48.1.28 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA512	800
4.48.1.29 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA512_224	800
4.48.1.30 CONFIG_EHSM_CRYPTO_ALGOFAM_SHA512_256	800
4.48.1.31 CONFIG_EHSM_CRYPTO_ALGOFAM_SM2	801
4.48.1.32 CONFIG_EHSM_CRYPTO_ALGOFAM_SM3	801
4.48.1.33 CONFIG_EHSM_CRYPTO_ALGOFAM_SM4	801
4.48.1.34 CONFIG_EHSM_CRYPTO_ALGOFAM_X963	801
4.48.1.35 CONFIG_EHSM_CRYPTO_ALGOMODE_CBC	801
4.48.1.36 CONFIG_EHSM_CRYPTO_ALGOMODE_CBC_MAC	801
4.48.1.37 CONFIG_EHSM_CRYPTO_ALGOMODE_CFB	802
4.48.1.38 CONFIG_EHSM_CRYPTO_ALGOMODE_CMAC	802
4.48.1.39 CONFIG_EHSM_CRYPTO_ALGOMODE_CTR	802
4.48.1.40 CONFIG_EHSM_CRYPTO_ALGOMODE_ECB	802
4.48.1.41 CONFIG_EHSM_CRYPTO_ALGOMODE_GMAC	802
4.48.1.42 CONFIG_EHSM_CRYPTO_ALGOMODE_OFB	802
4.48.1.43 CONFIG_EHSM_CRYPTO_ALGOMODE_RSASSA_PSS	803
4.48.1.44 CONFIG_EHSM_CRYPTO_RSA_CRT_MODE	803
4.48.1.45 CONFIG_EHSM_CRYPTO_V_CERT_MAX_PUB_K_SIZE	803
4.48.1.46 CONFIG_EHSM_CRYPTO_V_CERT_MAX_SIZE	803
4.48.1.47 CONFIG_EHSM_CRYPTO_V_GCM_MAX_AAD_SIZE	803
4.48.1.48 CONFIG_EHSM_CRYPTO_V_HMAC_MAX_KSIZE	804

4.48.1.49 CONFIG_EHSM_FIRMWARE_UPGRADE	804
4.48.1.50 CONFIG_EHSM_HASH_CORE_NUM	804
4.48.1.51 CONFIG_EHSM_HW_AHB_BYTE	804
4.48.1.52 CONFIG_EHSM_HW_BRANCH_2_0_0	804
4.48.1.53 CONFIG_EHSM_HW_COUNTER	805
4.48.1.54 CONFIG_EHSM_HW_FLASH	805
4.48.1.55 CONFIG_EHSM_HW_FLASH_ECC	805
4.48.1.56 CONFIG_EHSM_HW_GUOMI_LEVEL1	805
4.48.1.57 CONFIG_EHSM_HW_HASH	805
4.48.1.58 CONFIG_EHSM_HW_HASH_DMA	806
4.48.1.59 CONFIG_EHSM_HW_HASH_LP	806
4.48.1.60 CONFIG_EHSM_HW_INSTALL_K_KEK	806
4.48.1.61 CONFIG_EHSM_HW_LIFE_CYCLE_DEBUG_MODE	806
4.48.1.62 CONFIG_EHSM_HW_LIFE_CYCLE_DESTROY_MODE	806
4.48.1.63 CONFIG_EHSM_HW_LIFE_CYCLE_DEVELOP_MODE	807
4.48.1.64 CONFIG_EHSM_HW_LIFE_CYCLE_MANUFACTURE_MODE	807
4.48.1.65 CONFIG_EHSM_HW_LIFE_CYCLE_TEST_MODE	807
4.48.1.66 CONFIG_EHSM_HW_LIFE_CYCLE_USER_MODE	807
4.48.1.67 CONFIG_EHSM_HW_LOW_POWER	807
4.48.1.68 CONFIG_EHSM_HW_OTP	808
4.48.1.69 CONFIG_EHSM_HW_OTP_MAP	808
4.48.1.70 CONFIG_EHSM_HW_PKE	808
4.48.1.71 CONFIG_EHSM_HW_PKE_LP	808
4.48.1.72 CONFIG_EHSM_HW_SKE_DMA	808
4.48.1.73 CONFIG_EHSM_HW_SKE_LP	809
4.48.1.74 CONFIG_EHSM_HW_SKE_SECURE_PORT	809
4.48.1.75 CONFIG_EHSM_HW_TRNG	809
4.48.1.76 CONFIG_EHSM_HW_UTC_TIME	809
4.48.1.77 CONFIG_EHSM_HW_V_CODE_MAX_SIZE	809
4.48.1.78 CONFIG_EHSM_HW_V_CUSTOMER_OTP	810
4.48.1.79 CONFIG_EHSM_HW_V_FLASH_BASE_ADDR	810

4.48.1.80	CONFIG_EHSM_HW_V_FLASH_CUSTOMER	810
4.48.1.81	CONFIG_EHSM_HW_V_FLASH_DATA_ADDR	810
4.48.1.82	CONFIG_EHSM_HW_V_FLASH_DATA_SIZE	810
4.48.1.83	CONFIG_EHSM_HW_V_FLASH_ERASE_CELL_VALUE	810
4.48.1.84	CONFIG_EHSM_HW_V_FLASH_FREE_ECC_IDX1	811
4.48.1.85	CONFIG_EHSM_HW_V_FLASH_FREE_ECC_IDX2	811
4.48.1.86	CONFIG_EHSM_HW_V_FLASH_KEY_ADDR	811
4.48.1.87	CONFIG_EHSM_HW_V_FLASH_KEY_SIZE	811
4.48.1.88	CONFIG_EHSM_HW_V_FLASH_LOG_ADDR	811
4.48.1.89	CONFIG_EHSM_HW_V_FLASH_LOG_SIZE	811
4.48.1.90	CONFIG_EHSM_HW_V_FLASH_NONE	812
4.48.1.91	CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE	812
4.48.1.92	CONFIG_EHSM_HW_V_FLASH_SIMULATE	812
4.48.1.93	CONFIG_EHSM_HW_V_FLASH_SIZE	812
4.48.1.94	CONFIG_EHSM_HW_V_FLASH_SYS_ADDR	812
4.48.1.95	CONFIG_EHSM_HW_V_FLASH_SYS_SIZE	812
4.48.1.96	CONFIG_EHSM_HW_V_FLASH_TYPE	813
4.48.1.97	CONFIG_EHSM_HW_V_FLASH_WRITE_MIN_BYTES	813
4.48.1.98	CONFIG_EHSM_HW_V_NONE_OTP	813
4.48.1.99	CONFIG_EHSM_HW_V_OTP_BASE_ADDR	813
4.48.1.100	CONFIG_EHSM_HW_V_OTP_K_ATTR_LENGTH	813
4.48.1.101	CONFIG_EHSM_HW_V_OTP_KEY_NUM	813
4.48.1.102	CONFIG_EHSM_HW_V_OTP_PAGE_SIZE	814
4.48.1.103	CONFIG_EHSM_HW_V_OTP_SIZE	814
4.48.1.104	CONFIG_EHSM_HW_V_OTP_TYPE	814
4.48.1.105	CONFIG_EHSM_HW_V_OTP_VERSION_LENGTH	814
4.48.1.106	CONFIG_EHSM_HW_V_OTP_WRITE_MIN_BYTES	814
4.48.1.107	CONFIG_EHSM_HW_V_SIMULATE_OTP	814
4.48.1.108	CONFIG_EHSM_HW_V_WORK_FREQ	815
4.48.1.109	CONFIG_EHSM_JTAG_DEBUG_AUTH	815
4.48.1.110	CONFIG_EHSM_KMGR_CHECK_OTP_K_ATTR	815

4.48.1.111	CONFIG_EHSM_KMGR_PLAIN_K_IMPORT	815
4.48.1.112	CONFIG_EHSM_KMGR_V_ECC_K_AREA_SIZE	815
4.48.1.113	CONFIG_EHSM_KMGR_V_ECC_K_END_ADDR	816
4.48.1.114	CONFIG_EHSM_KMGR_V_ECC_K_NUM	816
4.48.1.115	CONFIG_EHSM_KMGR_V_ECC_K_SLOT_SIZE	816
4.48.1.116	CONFIG_EHSM_KMGR_V_ECC_K_START_ADDR	816
4.48.1.117	CONFIG_EHSM_KMGR_V_ECC_RAM_K_NUM	816
4.48.1.118	CONFIG_EHSM_KMGR_V_FLASH_K_START_OFFSET	816
4.48.1.119	CONFIG_EHSM_KMGR_V_FLASH_OTP_K_START_OFFSET	817
4.48.1.120	CONFIG_EHSM_KMGR_V_MAX_AUTH_CODE_SIZE	817
4.48.1.121	CONFIG_EHSM_KMGR_V_OTP_EXT_K_OFF	817
4.48.1.122	CONFIG_EHSM_KMGR_V_OTP_EXT_K_SIZE	817
4.48.1.123	CONFIG_EHSM_KMGR_V_RAM_K_MEM_SIZE	817
4.48.1.124	CONFIG_EHSM_KMGR_V_RSA_K_AREA_SIZE	817
4.48.1.125	CONFIG_EHSM_KMGR_V_RSA_K_END_ADDR	818
4.48.1.126	CONFIG_EHSM_KMGR_V_RSA_K_NUM	818
4.48.1.127	CONFIG_EHSM_KMGR_V_RSA_K_SLOT_SIZE	818
4.48.1.128	CONFIG_EHSM_KMGR_V_RSA_K_START_ADDR	818
4.48.1.129	CONFIG_EHSM_KMGR_V_RSA_RAM_K_NUM	818
4.48.1.130	CONFIG_EHSM_KMGR_V_SHE_K_AREA_SIZE	818
4.48.1.131	CONFIG_EHSM_KMGR_V_SHE_K_END_ADDR	819
4.48.1.132	CONFIG_EHSM_KMGR_V_SHE_K_NUM	819
4.48.1.133	CONFIG_EHSM_KMGR_V_SHE_K_SLOT_SIZE	819
4.48.1.134	CONFIG_EHSM_KMGR_V_SHE_K_START_ADDR	819
4.48.1.135	CONFIG_EHSM_KMGR_V_SM9_K_AREA_SIZE	819
4.48.1.136	CONFIG_EHSM_KMGR_V_SM9_K_END_ADDR	820
4.48.1.137	CONFIG_EHSM_KMGR_V_SM9_K_NUM	820
4.48.1.138	CONFIG_EHSM_KMGR_V_SM9_K_SLOT_SIZE	820
4.48.1.139	CONFIG_EHSM_KMGR_V_SM9_K_START_ADDR	820
4.48.1.140	CONFIG_EHSM_KMGR_V_SYM_K_AREA_SIZE	820
4.48.1.141	CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR	820

4.48.1.142	CONFIG_EHSM_KMGR_V_SYM_K_NUM	821
4.48.1.143	CONFIG_EHSM_KMGR_V_SYM_K_SLOT_SIZE	821
4.48.1.144	CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR	821
4.48.1.145	CONFIG_EHSM_KMGR_V_SYM_RAM_K_NUM	821
4.48.1.146	CONFIG_EHSM_LOG	821
4.48.1.147	CONFIG_EHSM_PKE_CORE_NUM	821
4.48.1.148	CONFIG_EHSM_SHE_SOC_BOOT	822
4.48.1.149	CONFIG_EHSM_SKE_CORE_NUM	822
4.48.1.150	CONFIG_EHSM_SOC_UPGRADE_AND_VERIFY	822
4.48.1.151	CONFIG_EHSM_TRNG_CORE_NUM	822
4.48.1.152	CONFIG_EHSM_UNIT_TEST	822
4.48.1.153	CONFIG_EHSM_USER_AUTH_KYE_IN_KMU	823
4.48.1.154	CONFIG_EHSM_V_LOG_DEBUG	823
4.48.1.155	CONFIG_EHSM_V_LOG_ERR	823
4.48.1.156	CONFIG_EHSM_V_LOG_INFO	823
4.48.1.157	CONFIG_EHSM_V_LOG_LEVEL	823
4.48.1.158	CONFIG_EHSM_V_LOG_WARN	824
4.49	eHSM_Mailbox_CmdId_Ip.h File Reference	824
4.49.1	Macro Definition Documentation	826
4.49.1.1	EHSM_AES_128	826
4.49.1.2	EHSM_AES_192	826
4.49.1.3	EHSM_AES_256	826
4.49.1.4	EHSM_AES_CTRDRBG	827
4.49.1.5	EHSM_CBC_MAC_MODE	827
4.49.1.6	EHSM_CBC_MODE	827
4.49.1.7	EHSM_CFB_MODE	827
4.49.1.8	EHSM_CMAC_MODE	827
4.49.1.9	EHSM_CMD_AEAD_CCM	827
4.49.1.10	EHSM_CMD_AEAD_GCM	828
4.49.1.11	EHSM_CMD_CHANGE_CONTROL_FIELD	828
4.49.1.12	EHSM_CMD_CHANGE_LIFECYCLE	828

4.49.1.13 EHSM_CMD_CLOSE_DEBUG	828
4.49.1.14 EHSM_CMD_COPY_EVITA_KEY	828
4.49.1.15 EHSM_CMD_CREATE_COUNTER	828
4.49.1.16 EHSM_CMD_CREATE_DH_KEY	829
4.49.1.17 EHSM_CMD_DEBUG_AUTHENCATION	829
4.49.1.18 EHSM_CMD_DELETE_COUNTER	829
4.49.1.19 EHSM_CMD_DERIVE_KEY	829
4.49.1.20 EHSM_CMD_ECCP_GEN_KEY	829
4.49.1.21 EHSM_CMD_ECDSA	829
4.49.1.22 EHSM_CMD_ECIES	830
4.49.1.23 EHSM_CMD_EXPORT_KEY	830
4.49.1.24 EHSM_CMD_FW_ENCRYPT_KEY	830
4.49.1.25 EHSM_CMD_FW_GET_RANDOM_KEY	830
4.49.1.26 EHSM_CMD_GEN_DH_KEY_PAIR	830
4.49.1.27 EHSM_CMD_GET_CHALLENGE	830
4.49.1.28 EHSM_CMD_GET_PUB_FROM_PRIV	831
4.49.1.29 EHSM_CMD_GET_SHE_ID	831
4.49.1.30 EHSM_CMD_GET_SHE_STATUS	831
4.49.1.31 EHSM_CMD_HASH	831
4.49.1.32 EHSM_CMD_IMAGE_UPGRADE	831
4.49.1.33 EHSM_CMD_IMAGE_VERIFY	831
4.49.1.34 EHSM_CMD_IMPORT_KEY	832
4.49.1.35 EHSM_CMD_INCREASE_COUNTER	832
4.49.1.36 EHSM_CMD_KEY_REMOVE	832
4.49.1.37 EHSM_CMD_KEY_STATUS	832
4.49.1.38 EHSM_CMD_LOW_POWER	832
4.49.1.39 EHSM_CMD_MAC	832
4.49.1.40 EHSM_CMD_MODULE_STATUS	833
4.49.1.41 EHSM_CMD_READ_COUNTER	833
4.49.1.42 EHSM_CMD_READ_OTP_DATA	833
4.49.1.43 EHSM_CMD_RESET_FIRMWARE	833

4.49.1.44 EHSM_CMD_RNG_GENERATE	833
4.49.1.45 EHSM_CMD_RSA_CIPHER	833
4.49.1.46 EHSM_CMD_RSA_GEN_KEY	834
4.49.1.47 EHSM_CMD_RSA_SIGN	834
4.49.1.48 EHSM_CMD_SELF_TEST	834
4.49.1.49 EHSM_CMD_SENSOR_RESP_INIT	834
4.49.1.50 EHSM_CMD_SET_BAUDRATE	834
4.49.1.51 EHSM_CMD_SHE_LOAD_KEY	834
4.49.1.52 EHSM_CMD_SHE_LOAD_PLAIN_KEY	835
4.49.1.53 EHSM_CMD_SHE_RAM_KEY_EXPORT	835
4.49.1.54 EHSM_CMD_SM2_CIPHER	835
4.49.1.55 EHSM_CMD_SM2_GEN_KEY	835
4.49.1.56 EHSM_CMD_SM2_SIGN	835
4.49.1.57 EHSM_CMD_SOC_BOOT_STATUS	835
4.49.1.58 EHSM_CMD_SOC_IMAGE_VERIFY	836
4.49.1.59 EHSM_CMD_SYM_CIPHER	836
4.49.1.60 EHSM_CMD_SYM_GEN_KEY	836
4.49.1.61 EHSM_CMD_UART_COMMAND [1/2]	836
4.49.1.62 EHSM_CMD_UART_COMMAND [2/2]	836
4.49.1.63 EHSM_CMD_WRITE_OTP_DATA	836
4.49.1.64 EHSM_CRT_MODE	837
4.49.1.65 EHSM_CTR_MODE	837
4.49.1.66 EHSM_DECRYPTION	837
4.49.1.67 EHSM_DES	837
4.49.1.68 EHSM_ECB_MODE	837
4.49.1.69 EHSM_ENCRYPTION	837
4.49.1.70 EHSM_FINISH	838
4.49.1.71 EHSM_GMAC_MODE	838
4.49.1.72 EHSM_INVALID_ALG	838
4.49.1.73 EHSM_INVALID_DIR	838
4.49.1.74 EHSM_MAC_GENERATION	838

4.49.1.75 EHSM_MAC_VERIFICATION	838
4.49.1.76 EHSM_MD5	839
4.49.1.77 EHSM_NO_TIME_STAMP	839
4.49.1.78 EHSM_NONE_CRT	839
4.49.1.79 EHSM_NOPADDING	839
4.49.1.80 EHSM_OFB_MODE	839
4.49.1.81 EHSM_ONEPASS	839
4.49.1.82 EHSM_ONEWITHZEROS	840
4.49.1.83 EHSM_PKCS7	840
4.49.1.84 EHSM_RSASSA_PPS	840
4.49.1.85 EHSM_SHA1	840
4.49.1.86 EHSM_SHA224	840
4.49.1.87 EHSM_SHA256	840
4.49.1.88 EHSM_SHA384	841
4.49.1.89 EHSM_SHA3_224	841
4.49.1.90 EHSM_SHA3_256	841
4.49.1.91 EHSM_SHA3_384	841
4.49.1.92 EHSM_SHA3_512	841
4.49.1.93 EHSM_SHA512	841
4.49.1.94 EHSM_SHA512_224	842
4.49.1.95 EHSM_SHA512_256	842
4.49.1.96 EHSM_SIGN_GENERATION	842
4.49.1.97 EHSM_SIGN_VERIFICATION	842
4.49.1.98 EHSM_SM3	842
4.49.1.99 EHSM_SM4	842
4.49.1.100EHSM_SM4_CTRDRBG	843
4.49.1.101EHSM_START	843
4.49.1.102EHSM_STREAMSTART	843
4.49.1.103EHSM_TDES_128	843
4.49.1.104EHSM_TDES_192	843
4.49.1.105EHSM_UPDATE	843

4.49.1.106 EHSM_USE_TIME_STAMP	844
4.49.1.107 EHSM_XOR	844
4.49.1.108 EHSM_XTS_MODE	844
4.49.2 Enumeration Type Documentation	844
4.49.2.1 cmd_type_e	844
4.50 eHSM_Mailbox_Ip.c File Reference	844
4.50.1 Macro Definition Documentation	845
4.50.1.1 HSMMBX_IRQ_PRIO	845
4.50.2 Function Documentation	845
4.50.2.1 __attribute__()	846
4.50.2.2 ehsm_is_cmd_addr_null()	846
4.50.2.3 ehsm_mbox_init()	846
4.50.2.4 ehsm_mbox_polling()	846
4.50.2.5 ehsm_mbox_send_cmd()	846
4.50.2.6 MAILBOX_Handler()	846
4.50.3 Variable Documentation	847
4.50.3.1 mbox_channel	847
4.51 eHSM_Mailbox_Ip.h File Reference	847
4.51.1 Typedef Documentation	847
4.51.1.1 mailbox_callback	847
4.51.1.2 mailbox_channel_st	848
4.51.2 Function Documentation	848
4.51.2.1 ehsm_get_tick()	848
4.51.2.2 ehsm_is_cmd_addr_null()	848
4.51.2.3 ehsm_mbox_init()	848
4.51.2.4 ehsm_mbox_send_cmd()	848
4.51.2.5 ehsm_tick_form_ms()	848
4.52 eHSM_Mailbox_Prtcl_Ip.h File Reference	849
4.52.1 Macro Definition Documentation	852
4.52.1.1 CMD_TAG_BYTE_SIZE	852
4.52.1.2 CMD_TAG_WORD_INDEX	852

4.52.1.3	CMD_TAG_WORD_SIZE	852
4.52.1.4	EHSM_CANCEL_CERT_TYPE_CMD	853
4.52.1.5	EHSM_CANCEL_SINGLE_CMD	853
4.52.1.6	EHSM_CMD_CIPHER_KEY_TYPE_EVITA	853
4.52.1.7	EHSM_CMD_CIPHER_KEY_TYPE_SHE	853
4.52.1.8	EHSM_LOW_POWER_MODE	853
4.52.1.9	EHSM_NORMAL_MODE	853
4.52.1.10	H2S_SRV_CMD_CANCLE_NOTE_BIT	854
4.52.1.11	H2S_SRV_CMD_CANCLE_WORD_INDEX	854
4.52.1.12	H2S_SRV_CMD_CANCLE_WORD_SIZE	854
4.52.1.13	H2S_SRV_CMD_JTAG_END_BIT	854
4.52.1.14	H2S_SRV_CMD_JTAG_NOTE_BIT	854
4.52.1.15	H2S_SRV_GENERAL_NOTE_BIT	854
4.52.1.16	H2S_SRV_GENERAL_WORD_INDEX	855
4.52.1.17	H2S_SRV_GENERAL_WORD_SIZE	855
4.52.1.18	H2S_SRV_JTAG_WORD_INDEX	855
4.52.1.19	H2S_SRV_JTAG_WORD_SIZE	855
4.52.1.20	H2S_SRV_MGR_NOTE_BIT	855
4.52.1.21	H2S_SRV_MGR_WORD_INDEX	855
4.52.1.22	H2S_SRV_MGR_WORD_SIZE	856
4.52.1.23	HOST_ADDRESS_SIZE	856
4.52.1.24	MAILBOX_CMD_MAX_SIZE	856
4.52.1.25	MAX_IMPORT_KEY_SIZE	856
4.52.1.26	MAX_KEY_AUTH_VALUE_SIZE	856
4.52.1.27	MAX_KEY_DERIVED_PWD_SIZE	856
4.52.1.28	MAX_SM9_CIPHER_KEY_SIZE	857
4.52.1.29	MAX_SM9_USER_ID_SIZE	857
4.52.1.30	MAX_SM9_WARP_KEY_SIZE	857
4.52.1.31	RESPONSE_TAG_INDEX	857
4.52.1.32	S2H_SRV_CMD_CANCLE_NOTE_BIT	857
4.52.1.33	S2H_SRV_CMD_CANCLE_WORD_INDEX	857

4.52.1.34 S2H_SRV_CMD_CANCEL_WORD_SIZE	858
4.52.1.35 S2H_SRV_CMD_JTAG_END_BIT	858
4.52.1.36 S2H_SRV_CMD_JTAG_NOTE_BIT	858
4.52.1.37 S2H_SRV_CMD_JTAG_WORD_INDEX	858
4.52.1.38 S2H_SRV_GENERAL_NOTE_BIT	858
4.52.1.39 S2H_SRV_GENERAL_WORD_INDEX	858
4.52.1.40 S2H_SRV_GENERAL_WORD_SIZE	859
4.52.1.41 S2H_SRV_JTAG_WORD_SIZE	859
4.52.1.42 S2H_SRV_MGR_NOTE_BIT	859
4.52.1.43 S2H_SRV_MGR_WORD_INDEX	859
4.52.1.44 S2H_SRV_MGR_WORD_SIZE	859
4.52.2 Typedef Documentation	859
4.52.2.1 ehsm_change_control_field_cmd_st	859
4.52.2.2 ehsm_change_lifecycle_cmd_st	860
4.52.2.3 ehsm_close_debug_cmd_st	860
4.52.2.4 ehsm_copy_key_cmd_st	860
4.52.2.5 ehsm_create_dh_key_cmd_st	860
4.52.2.6 ehsm_create_dh_sm2_ext_param_st	860
4.52.2.7 ehsm_debug_authentication_cmd_st	860
4.52.2.8 ehsm_derive_key_cmd_st	860
4.52.2.9 ehsm_export_key_cmd_st	860
4.52.2.10 ehsm_fw_encrypt_key_cmd_st	861
4.52.2.11 ehsm_fw_get_random_key_cmd_st	861
4.52.2.12 ehsm_gen_key_cmd_st	861
4.52.2.13 ehsm_gen_sm9_userpriv_key_cmd_st	861
4.52.2.14 ehsm_get_challenge_cmd_st	861
4.52.2.15 ehsm_get_emu_cmd_st	861
4.52.2.16 ehsm_get_pub_from_priv_cmd_st	861
4.52.2.17 ehsm_get_she_id_cmd_st	861
4.52.2.18 ehsm_image_upgrade_cmd_st	862
4.52.2.19 ehsm_image_verify_cmd_st	862

4.52.2.20 ehsm_import_key_cmd_st	862
4.52.2.21 ehsm_key_remove_cmd_st	862
4.52.2.22 ehsm_key_status_cmd_st	862
4.52.2.23 ehsm_low_power_cmd_st	862
4.52.2.24 ehsm_mailbox_req_st	862
4.52.2.25 ehsm_mbox_cancel_channel_req_st	862
4.52.2.26 ehsm_mbox_cancel_channel_rps_st	863
4.52.2.27 ehsm_mbox_mgr_channel_req_st	863
4.52.2.28 ehsm_module_status_cmd_st	863
4.52.2.29 ehsm_otp_read_cmd_st	863
4.52.2.30 ehsm_otp_write_cmd_st	863
4.52.2.31 ehsm_rng_generate_cmd_st	863
4.52.2.32 ehsm_self_test_cmd_st	863
4.52.2.33 ehsm_sensor_resp_init_cmd_st	863
4.52.2.34 ehsm_set_baudrate_cmd_st	864
4.52.2.35 ehsm_she_load_export_key_cmd_st	864
4.52.2.36 ehsm_she_load_plain_key_cmd_st	864
4.52.2.37 ehsm_sm9_exchg_gen_usertmp_cmd_st	864
4.52.2.38 ehsm_sm9_exchg_key_cmd_child_st	864
4.52.2.39 ehsm_sm9_exchg_key_cmd_st	864
4.52.2.40 ehsm_sm9_export_key_cmd_st	864
4.52.2.41 ehsm_sm9_get_mast_pubkey_cmd_st	864
4.52.2.42 ehsm_sm9_get_tmp_pubkey_cmd_st	865
4.52.2.43 ehsm_sm9_import_key_cmd_st	865
4.52.2.44 ehsm_sm9_remove_key_cmd_st	865
4.52.2.45 ehsm_sm9_unwrap_key_cmd_st	865
4.52.2.46 ehsm_sm9_wrap_key_cmd_st	865
4.52.2.47 ehsm_soc_image_verify_cmd_st	865
4.52.2.48 ehsm_uart_cmd_st	865
4.52.3 Enumeration Type Documentation	865
4.52.3.1 mailbox_channel_e	865

4.53 eHSM_Mailbox_Reg_Ip.h File Reference	866
4.53.1 Macro Definition Documentation	867
4.53.1.1 HOST2HSM_ACCESS_PATH	867
4.53.1.2 HSM_STATUS_IN	867
4.53.1.3 HSM_STATUS_IN1	867
4.53.1.4 KBUF_BASE	868
4.53.1.5 KMU_BASE	868
4.53.1.6 MB_H2S_NOTE	868
4.53.1.7 MB_H2S_SOC_INT	868
4.53.1.8 MB_H2S_SOC_INT_EN	868
4.53.1.9 MB_HSM_STATUS0	868
4.53.1.10 MB_HSM_STATUS1	869
4.53.1.11 MB_S2H_NOTE	869
4.53.1.12 MB_S2H_SOC_INT	869
4.53.1.13 MB_S2H_SOC_INT_EN	869
4.53.1.14 MBOX_HOST2HSM_HOST_INT	869
4.53.1.15 MBOX_HOST2HSM_HOST_INT_EN	869
4.53.1.16 MBOX_HOST2HSM_HSM_INT	870
4.53.1.17 MBOX_HOST2HSM_HSM_INT_EN	870
4.53.1.18 MBOX_HSM2HOST_HOST_INT	870
4.53.1.19 MBOX_HSM2HOST_HOST_INT_EN	870
4.53.1.20 MBOX_HSMHOST_HSM_INT	870
4.53.1.21 MBOX_HSMHOST_HSM_INT_EN	870
4.53.1.22 MBOX_SOCBASE	871
4.53.1.23 rATTR_D0	871
4.53.1.24 rATTR_D1	871
4.53.1.25 rATTR_D2	871
4.53.1.26 rERR_ST	871
4.53.1.27 rKBUF	871
4.53.1.28 rKMU_CTRL	872
4.53.1.29 rKMU_INT_EN	872

4.53.1.30 rKMU_STA	872
4.53.1.31 rSN_D0	872
4.53.1.32 rSN_D1	872
4.53.1.33 rSOCMBOX_CMD_D0	872
4.53.1.34 rSOCMBOX_CMD_D1	873
4.53.1.35 rSOCMBOX_CMD_D10	873
4.53.1.36 rSOCMBOX_CMD_D11	873
4.53.1.37 rSOCMBOX_CMD_D12	873
4.53.1.38 rSOCMBOX_CMD_D13	873
4.53.1.39 rSOCMBOX_CMD_D14	873
4.53.1.40 rSOCMBOX_CMD_D15	874
4.53.1.41 rSOCMBOX_CMD_D2	874
4.53.1.42 rSOCMBOX_CMD_D3	874
4.53.1.43 rSOCMBOX_CMD_D4	874
4.53.1.44 rSOCMBOX_CMD_D5	874
4.53.1.45 rSOCMBOX_CMD_D6	874
4.53.1.46 rSOCMBOX_CMD_D7	875
4.53.1.47 rSOCMBOX_CMD_D8	875
4.53.1.48 rSOCMBOX_CMD_D9	875
4.53.1.49 rSOCMBOX_RSP_D0	875
4.53.1.50 rSOCMBOX_RSP_D01	875
4.53.1.51 rVER_D0	875
4.53.1.52 rVER_D1	876
4.53.1.53 rVER_D2	876
4.53.1.54 rVER_D3	876
4.53.1.55 STATUS_BASE	876
4.54 eHSM_Mgr_Ctx_lp.c File Reference	876
4.54.1 Function Documentation	877
4.54.1.1 ehsm_cipher_ctx_data_check()	877
4.54.1.2 ehsm_disjunction_finish()	877
4.54.1.3 ehsm_disjunction_updata()	877

4.54.1.4	ehsm_init_block_mgr()	878
4.54.1.5	eHSM_Mgr_Cipher_Ctx_Find()	878
4.54.1.6	eHSM_Mgr_Cipher_Ctx_Free()	878
4.54.1.7	eHSM_Mgr_Cipher_Ctx_Get_Free()	878
4.54.1.8	ehsm_mgr_ctx_find()	878
4.54.1.9	ehsm_mgr_ctx_free()	879
4.54.1.10	ehsm_mgr_ctx_get_free()	879
4.55	eHSM_Mgr_Ctx_Ip.h File Reference	879
4.55.1	Macro Definition Documentation	880
4.55.1.1	EHSM_ERR_CTX_MGR_BUFFER_DATA_VALID	880
4.55.1.2	EHSM_ERR_CTX_MGR_DATA_NOT_READY	880
4.55.2	Typedef Documentation	880
4.55.2.1	ehsm_aead_data_ptr_st	880
4.55.2.2	ehsm_ctx_block_mgr_st	880
4.55.3	Enumeration Type Documentation	880
4.55.3.1	ehsm_asym_alg_e	880
4.55.4	Function Documentation	881
4.55.4.1	ehsm_cipher_ctx_data_check()	881
4.55.4.2	ehsm_disjunction_finish()	881
4.55.4.3	ehsm_disjunction_updata()	881
4.55.4.4	ehsm_init_block_mgr()	881
4.55.4.5	eHSM_Mgr_Cipher_Ctx_Find()	882
4.55.4.6	eHSM_Mgr_Cipher_Ctx_Free()	882
4.55.4.7	eHSM_Mgr_Cipher_Ctx_Get_Free()	882
4.55.4.8	ehsm_mgr_ctx_find()	882
4.55.4.9	ehsm_mgr_ctx_free()	882
4.55.4.10	ehsm_mgr_ctx_get_free()	882
4.56	ehsm_register_addr.h File Reference	883
4.57	eHSM_Srv_AsymCper_Ip.c File Reference	883
4.57.1	Variable Documentation	883
4.57.1.1	srv_crypto_pke	883

4.58 eHSM_Srv_Ciper_Ip.c File Reference	883
4.58.1 Function Documentation	884
4.58.1.1 _srv_crypto_cipher_reqhdl()	884
4.58.1.2 _srv_crypto_cipher_rsphdl()	884
4.58.2 Variable Documentation	884
4.58.2.1 srv_crypto_ske	884
4.59 eHSM_Srv_Cipher_Ip.h File Reference	885
4.59.1 Typedef Documentation	885
4.59.1.1 ehsm_cmd_cipher_with_rps_st	885
4.59.2 Function Documentation	885
4.59.2.1 _srv_crypto_cipher_reqhdl()	885
4.59.2.2 _srv_crypto_cipher_rsphdl()	885
4.60 eHSM_Srv_CmdReq_Ip.h File Reference	886
4.60.1 Macro Definition Documentation	886
4.60.1.1 EHSM_CMD_PRIORITY_DEFAULT	886
4.60.1.2 MAX_RESPONSE_DATA_SIZE	886
4.60.2 Typedef Documentation	887
4.60.2.1 cmd_release_cb	887
4.60.2.2 cmd_req_cb	887
4.60.2.3 ehsm_cmd_req_st	887
4.60.3 Enumeration Type Documentation	887
4.60.3.1 ehsm_cmd_req_state_e	887
4.60.3.2 ehsm_cmd_req_type_e	888
4.61 eHSM_Srv_Counter_Ip.c File Reference	889
4.62 eHSM_Srv_Ext_Ip.c File Reference	889
4.62.1 Variable Documentation	890
4.62.1.1 g_srv_soc_image_verify	890
4.62.1.2 srv_asr_cancel_cmd	890
4.62.1.3 srv_bootloader_cmd	891
4.62.1.4 srv_change_control_field	891
4.62.1.5 srv_change_lifecycle	891

4.62.1.6	srv_close_debug	892
4.62.1.7	srv_debug_auth	892
4.62.1.8	srv_fw_encrypt_key	892
4.62.1.9	srv_fw_get_random_key	893
4.62.1.10	srv_get_challenge	893
4.62.1.11	srv_get_module_status	893
4.62.1.12	srv_get_she_id	894
4.62.1.13	srv_get_she_status	894
4.62.1.14	srv_image_upgrade	894
4.62.1.15	srv_image_verify	895
4.62.1.16	srv_low_power	895
4.62.1.17	srv_read_otp_data	895
4.62.1.18	srv_self_test	896
4.62.1.19	srv_set_baudrate	896
4.62.1.20	srv_she_cancel_cmd	896
4.62.1.21	srv_soc_boot_status	897
4.62.1.22	srv_write_otp_data	897
4.63	eHSM_Srv_Hash_Ip.c File Reference	897
4.63.1	Variable Documentation	897
4.63.1.1	srv_crypto_hash	898
4.64	eHSM_Srv_Key_Ip.c File Reference	898
4.64.1	Macro Definition Documentation	899
4.64.1.1	DEFAULT_RSA_E_SIZE	899
4.64.2	Variable Documentation	899
4.64.2.1	srv_certificate_parse	899
4.64.2.2	srv_certificate_verify	899
4.64.2.3	srv_create_dh_key	900
4.64.2.4	srv_create_random_key	900
4.64.2.5	srv_derive_key	900
4.64.2.6	srv_export_key	901
4.64.2.7	srv_get_pub_from_priv	901

4.64.2.8	srv_import_key	901
4.64.2.9	srv_key_copy	902
4.64.2.10	srv_key_remove	902
4.64.2.11	srv_key_status	902
4.64.2.12	srv_she_load_key	903
4.64.2.13	srv_she_load_plain_key	903
4.64.2.14	srv_she_ram_key_export	903
4.65	eHSM_Srv_Mgr_Ip.c File Reference	903
4.65.1	Macro Definition Documentation	905
4.65.1.1	COMMAND_REQ_QUANTITY	905
4.65.2	Typedef Documentation	905
4.65.2.1	ehsm_cmd_req_buffer_st	905
4.65.3	Function Documentation	905
4.65.3.1	ehsm_cancel_single_service()	906
4.65.3.2	ehsm_process_asr_service()	906
4.65.3.3	ehsm_process_sync_service()	906
4.65.3.4	ehsm_register_service()	906
4.65.3.5	ehsm_service_init()	906
4.65.3.6	ehsm_set_address_pointer()	907
4.65.3.7	hw_interrupt_disable()	907
4.65.3.8	hw_interrupt_enable()	907
4.65.4	Variable Documentation	907
4.65.4.1	g_cmd_req_buffer	907
4.65.4.2	g_srv_soc_image_verify	907
4.65.4.3	service_table	907
4.65.4.4	srv_asr_candle_cmd	908
4.65.4.5	srv_bootloader_cmd	908
4.65.4.6	srv_certificate_parse	908
4.65.4.7	srv_certificate_verify	908
4.65.4.8	srv_change_control_field	908
4.65.4.9	srv_change_lifecycle	908

4.65.4.10	srv_close_debug	909
4.65.4.11	srv_create_dh_key	909
4.65.4.12	srv_create_random_key	909
4.65.4.13	srv_crypto_hash	909
4.65.4.14	srv_crypto_pke	909
4.65.4.15	srv_crypto_randomgenerate	909
4.65.4.16	srv_crypto_ske	910
4.65.4.17	srv_debug_auth	910
4.65.4.18	srv_derive_key	910
4.65.4.19	srv_export_key	910
4.65.4.20	srv_fw_encrypt_key	910
4.65.4.21	srv_fw_get_random_key	910
4.65.4.22	srv_get_challenge	911
4.65.4.23	srv_get_module_status	911
4.65.4.24	srv_get_pub_from_priv	911
4.65.4.25	srv_get_she_id	911
4.65.4.26	srv_get_she_status	911
4.65.4.27	srv_image_upgrade	911
4.65.4.28	srv_image_verify	912
4.65.4.29	srv_import_key	912
4.65.4.30	srv_key_copy	912
4.65.4.31	srv_key_remove	912
4.65.4.32	srv_key_status	912
4.65.4.33	srv_low_power	912
4.65.4.34	srv_read_otp_data	913
4.65.4.35	srv_set_baudrate	913
4.65.4.36	srv_she_cancle_cmd	913
4.65.4.37	srv_she_load_key	913
4.65.4.38	srv_she_load_plain_key	913
4.65.4.39	srv_she_ram_key_export	913
4.65.4.40	srv_soc_boot_status	914

4.65.4.41	srv_write_otp_data	914
4.66	eHSM_Srv_Mgr_Ip.h File Reference	914
4.66.1	Typedef Documentation	915
4.66.1.1	ehsm_service_info_st	915
4.66.1.2	ehsm_service_st	915
4.66.1.3	service_reqhdl	915
4.66.1.4	service_rsphdl	916
4.66.2	Enumeration Type Documentation	916
4.66.2.1	ehsm_cmd_ext_type_e	916
4.66.3	Function Documentation	917
4.66.3.1	ehsm_cancel_single_service()	917
4.66.3.2	ehsm_process_asr_service()	918
4.66.3.3	ehsm_process_sync_service()	918
4.66.3.4	ehsm_register_service()	918
4.66.3.5	ehsm_service_init()	918
4.66.3.6	ehsm_set_address_pointer()	918
4.67	eHSM_Srv_Rng_Ip.c File Reference	919
4.67.1	Variable Documentation	919
4.67.1.1	srv_crypto_randomgenerate	919
4.67.1.2	srv_crypto_randomseed	919
4.67.1.3	srv_rng_extend_seed	920
4.67.1.4	srv_rng_init	920
4.68	eHSM_Srv_Timer_Ip.c File Reference	920
4.69	eHSM_Types_Ip.h File Reference	920
4.69.1	Macro Definition Documentation	921
4.69.1.1	EHSM_ARRAY_SIZE	921
4.69.1.2	false	921
4.69.1.3	NULL	921
4.69.1.4	true	921
4.69.2	Typedef Documentation	922
4.69.2.1	ehsm_addr_t	922

4.69.2.2	ehsm_bool_t	922
4.69.2.3	ehsm_handle_t	922
4.69.2.4	ehsm_int16_t	922
4.69.2.5	ehsm_int32_t	922
4.69.2.6	ehsm_int64_t	922
4.69.2.7	ehsm_int8_t	923
4.69.2.8	ehsm_uint16_t	923
4.69.2.9	ehsm_uint32_t	923
4.69.2.10	ehsm_uint64_t	923
4.69.2.11	ehsm_uint8_t	923
4.70	Hsm_Hal.c File Reference	923
4.70.1	Detailed Description	926
4.70.2	Function Documentation	926
4.70.2.1	HSM_Hal_AesCipher()	926
4.70.2.2	HSM_Hal_CipherMac()	927
4.70.2.3	HSM_Hal_DebugAuth()	927
4.70.2.4	HSM_Hal_Deinit()	928
4.70.2.5	HSM_Hal_DeriveKey()	928
4.70.2.6	HSM_Hal_DisableSecureBoot()	929
4.70.2.7	HSM_Hal_EccSign()	929
4.70.2.8	HSM_Hal_EnableSecureBoot()	930
4.70.2.9	HSM_Hal_GenerateDHKey()	930
4.70.2.10	HSM_Hal_GenerateKey()	931
4.70.2.11	HSM_Hal_GetChallenge()	931
4.70.2.12	HSM_Hal_GetHsmFwVersion()	932
4.70.2.13	HSM_Hal_GetKeyStatus()	932
4.70.2.14	HSM_Hal_GetLifeCycle()	932
4.70.2.15	HSM_Hal_GetLockState()	933
4.70.2.16	HSM_Hal_GetPubKeyFromPrvKey()	933
4.70.2.17	HSM_Hal_GetRnd()	933
4.70.2.18	HSM_Hal_GetRndKey()	934

4.70.2.19 HSM_Hal_GetSecretkey()	935
4.70.2.20 HSM_Hal_Hash()	935
4.70.2.21 HSM_Hal_HashMac()	936
4.70.2.22 HSM_Hal_HostImageSecureUpgrade()	936
4.70.2.23 HSM_Hal_HostImageSecureVerify()	937
4.70.2.24 HSM_Hal_Init()	937
4.70.2.25 HSM_Hal_InstallCallback()	937
4.70.2.26 HSM_Hal_Lock()	938
4.70.2.27 HSM_Hal_OtpRead()	938
4.70.2.28 HSM_Hal_OtpWrite()	939
4.70.2.29 HSM_Hal_RemoveKey()	939
4.70.2.30 HSM_Hal_RsaCipher()	940
4.70.2.31 HSM_Hal_RsaSign()	940
4.70.2.32 HSM_Hal_SetImageSecureUpgradeAlgo()	941
4.70.2.33 HSM_Hal_SetImageSecureVerifyAlgo()	941
4.70.2.34 HSM_Hal_SetLifeCycle()	942
4.70.2.35 HSM_Hal_SetOtpExternalKey()	942
4.70.2.36 HSM_Hal_SetOtpKeyCipherAlgo()	943
4.70.2.37 HSM_Hal_SetOtpRndKey()	943
4.70.2.38 HSM_Hal_SetPlainKey()	944
4.70.2.39 HSM_Hal_SetSecretKey()	944
4.70.2.40 HSM_Hal_Sm2Cipher()	945
4.70.2.41 HSM_Hal_Sm2Sign()	945
4.70.2.42 HSM_Hal_Sm4Cipher()	946
4.70.2.43 HSM_Hal_Unlock()	946
4.70.2.44 ISR()	947
4.71 Hsm_Hal.h File Reference	947
4.71.1 Detailed Description	954
4.71.2 Macro Definition Documentation	954
4.71.2.1 AES128_CMAC_WITH_AES128_CBC_UPGRADE	954
4.71.2.2 AES128_CMAC_WITH_AES128_CBC_VERIFY	954

4.71.2.3 HSM_DEBUG_MODE	954
4.71.2.4 HSM_DESTROY_MODE	954
4.71.2.5 HSM_DEVELOP_MODE	954
4.71.2.6 HSM_DISABLE	955
4.71.2.7 HSM_FLASH_KEY_BASE	955
4.71.2.8 HSM_FLASH_KEY_MAX_SLOT_NUM	955
4.71.2.9 HSM_FLASH_KEY_OFFSET	955
4.71.2.10 HSM_FLASH_KEY_PAGE_NUM	955
4.71.2.11 HSM_FLASH_KEY_SLOT_LEN	955
4.71.2.12 HSM_FLASH_PAGE_NUM	956
4.71.2.13 HSM_FLASH_PAGE_NUM_OFFSET	956
4.71.2.14 HSM_FLASH_PAGE_REVERSE_LEN	956
4.71.2.15 HSM_FLASH_PAGE_VALID_LEN	956
4.71.2.16 HSM_KEY_DATA_VALID_TAG	956
4.71.2.17 HSM_MANU_MODE	956
4.71.2.18 HSM_PAGE_VALID_TAG	957
4.71.2.19 HSM_RAM_KEY_ECC_MAX_NUM	957
4.71.2.20 HSM_RAM_KEY_MAX_NUM	957
4.71.2.21 HSM_RAM_KEY_MEM_SIZE	957
4.71.2.22 HSM_RAM_KEY_RSA_MAX_NUM	957
4.71.2.23 HSM_RAM_KEY_SYM_MAX_NUM	957
4.71.2.24 HSM_SECURE_BOOT_DISABLE	958
4.71.2.25 HSM_SECURE_BOOT_ENABLE	958
4.71.2.26 HSM_SLOT_INVALID_TAG	958
4.71.2.27 HSM_SUCCESS	958
4.71.2.28 HSM_TEST_MODE	958
4.71.2.29 HSM_UNNORMAL_MODE	958
4.71.2.30 HSM_USER_MODE	959
4.71.2.31 OTP_KEY_CIPHER_AES_CIPHER	959
4.71.2.32 OTP_KEY_CIPHER_SM4_CIPHER	959
4.71.2.33 RSA_2048_WITH_AES128_CBC_UPGRADE	959

4.71.2.34	RSA_2048_WITH_AES128_CBC_VERIFY	959
4.71.2.35	SM2_WITH_SM4_CBC_UPGRADE	959
4.71.2.36	SM2_WITH_SM4_CBC_VERIFY	960
4.71.2.37	SM4_CMAC_WITH_SM4_CBC_UPGRADE	960
4.71.2.38	SM4_CMAC_WITH_SM4_CBC_VERIFY	960
4.71.3	Typedef Documentation	960
4.71.3.1	HSM_DhPubKeyType	960
4.71.3.2	HSM_ImageVerifyType	960
4.71.3.3	HSM_PriKeyDataType	960
4.71.3.4	HSM_PubKeyDataType	961
4.71.3.5	HSM_RsaPrvCrtType	961
4.71.3.6	HSM_SecureUpgradeType	961
4.71.4	Enumeration Type Documentation	961
4.71.4.1	HSM_AsymAlgoType	961
4.71.4.2	HSM_ChallengeType	961
4.71.4.3	HSM_CipherDirection	962
4.71.4.4	HSM_CipherMode	962
4.71.4.5	HSM_DebugAuthAlgoType	962
4.71.4.6	HSM_GenKeyAlgo	963
4.71.4.7	HSM_HashAlgoType	963
4.71.4.8	HSM_ImageType	964
4.71.4.9	HSM_IrqNum	964
4.71.4.10	HSM_KdfType	965
4.71.4.11	HSM_KeyAlgoType	965
4.71.4.12	HSM_KeyId	965
4.71.4.13	HSM_KeyStorageType	967
4.71.4.14	HSM_LifeCycleType	968
4.71.4.15	HSM_MacDirection	968
4.71.4.16	HSM_OtpCtrlAlgoType	968
4.71.4.17	HSM_OtpKeyCipherAlgo	969
4.71.4.18	HSM_OtpKeyId	969

4.71.4.19 HSM_OtpKeyLevel	969
4.71.4.20 HSM_PaddingType	970
4.71.4.21 HSM_ProcessMode	970
4.71.4.22 HSM_RndAlgo	970
4.71.4.23 HSM_RndOtpKeyType	971
4.71.4.24 HSM_SignDirection	971
4.71.4.25 HSM_SymAlgoType	971
4.71.5 Function Documentation	972
4.71.5.1 HSM_Hal_AesCipher()	972
4.71.5.2 HSM_Hal_CipherMac()	972
4.71.5.3 HSM_Hal_DebugAuth()	973
4.71.5.4 HSM_Hal_Deinit()	973
4.71.5.5 HSM_Hal_DeriveKey()	974
4.71.5.6 HSM_Hal_DisableSecureBoot()	974
4.71.5.7 HSM_Hal_EccSign()	975
4.71.5.8 HSM_Hal_EnableSecureBoot()	975
4.71.5.9 HSM_Hal_GenerateDHKey()	975
4.71.5.10 HSM_Hal_GenerateKey()	976
4.71.5.11 HSM_Hal_GetChallenge()	976
4.71.5.12 HSM_Hal_GetHsmFwVersion()	977
4.71.5.13 HSM_Hal_GetKeyStatus()	977
4.71.5.14 HSM_Hal_GetLifeCycle()	978
4.71.5.15 HSM_Hal_GetLockState()	978
4.71.5.16 HSM_Hal_GetPubKeyFromPrvKey()	978
4.71.5.17 HSM_Hal_GetRnd()	979
4.71.5.18 HSM_Hal_GetRndKey()	979
4.71.5.19 HSM_Hal_GetSecretkey()	980
4.71.5.20 HSM_Hal_Hash()	980
4.71.5.21 HSM_Hal_HashMac()	981
4.71.5.22 HSM_Hal_HostImageSecureUpgrade()	981
4.71.5.23 HSM_Hal_HostImageSecureVerify()	982

4.71.5.24 HSM_Hal_Init()	983
4.71.5.25 HSM_Hal_InstallCallback()	983
4.71.5.26 HSM_Hal_Lock()	984
4.71.5.27 HSM_Hal_OtpRead()	984
4.71.5.28 HSM_Hal_OtpWrite()	984
4.71.5.29 HSM_Hal_RemoveKey()	985
4.71.5.30 HSM_Hal_RsaCipher()	985
4.71.5.31 HSM_Hal_RsaSign()	986
4.71.5.32 HSM_Hal_SetImageSecureUpgradeAlgo()	986
4.71.5.33 HSM_Hal_SetImageSecureVerifyAlgo()	987
4.71.5.34 HSM_Hal_SetLifeCycle()	988
4.71.5.35 HSM_Hal_SetOtpExternalKey()	988
4.71.5.36 HSM_Hal_SetOtpKeyCipherAlgo()	989
4.71.5.37 HSM_Hal_SetOtpRndKey()	989
4.71.5.38 HSM_Hal_SetPlainKey()	990
4.71.5.39 HSM_Hal_SetSecretKey()	990
4.71.5.40 HSM_Hal_SetSecureBootCfg()	991
4.71.5.41 HSM_Hal_Sm2Cipher()	991
4.71.5.42 HSM_Hal_Sm2Sign()	992
4.71.5.43 HSM_Hal_Sm4Cipher()	993
4.71.5.44 HSM_Hal_Unlock()	993
4.72 test_int_config.h File Reference	994
4.72.1 Macro Definition Documentation	994
4.72.1.1 CONFIG_EHSM_UNIT_TEST_EVITA_SYM	994
4.73 utest_vecs_st.h File Reference	994
4.73.1 Macro Definition Documentation	994
4.73.1.1 MAX_TAP	994

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

aead_testvec	8
akcipher_testvec	11
bitmap	13
cipher_session_st	14
cipher_testvec	19
counter_value_64_t	21
Crypto_AlgorithmInfoType	22
crypto_copy_key_dh_key_info	23
crypto_copy_key_info	25
crypto_create_evita_key_info	26
crypto_evita_key_info	27
crypto_exported_key	28
crypto_import_evita_key_info	30
Crypto_JobInfoType	32
Crypto_JobPrimitiveInfoType	33
Crypto_JobPrimitiveInputOutputType	34
Crypto_JobRedirectionInfoType	38
Crypto_JobType	40
crypto_key_derive_info	42
crypto_key_element_type_info_st	43
crypto_key_export_info	45
crypto_key_status_info	47
crypto_object	48
Crypto_PrimitiveInfoType	50
crypto_she_key	51
CryptoDriverObject	52
CryptoKey	53
CryptoKeyElementType	54
CryptoKeyType	56
CryptoPrimitive	57
dlist_head	58
ecies_testvec	59
ehsm_aead_data_ptr	63
ehsm_certificate_verify_st	64
ehsm_change_control_field_cmd	65
ehsm_change_control_field_st	66
ehsm_change_lifecycle_cmd	67
ehsm_close_debug_cmd	68

ehsm_cmd	68
ehsm_cmd_aead_ptr_st	
GCM/CCM data structure. It's the content of ehsm_cmd_cipher_st.input_addr/output_addr	69
ehsm_cmd_cipher_st	70
ehsm_cmd_hdr_eccp_keygen_st	
Header for sm2 and eccp key generation	73
ehsm_cmd_hdr_ecise_st	
Header for ecies	74
ehsm_cmd_hdr_pke_st	
Header for sm2/rsa/ecdsa	76
ehsm_cmd_hdr_rng_st	
Header for RNG	78
ehsm_cmd_hdr_rsa_keygen_st	
Header for rsa key generation	79
ehsm_cmd_hdr_ske_st	
Header for Ske/Aead/Mac/Hash	80
ehsm_cmd_hdr_sm9_st	
Header for SM9 ciphert and signature	82
ehsm_cmd_req	84
ehsm_cmd_req_buffer	87
ehsm_cmd_sm9_sig_vry_output_ptr_st	
SM9 signature and verification structure. It's the content of ehsm_cmd_cipher_st.output_addr	88
ehsm_copy_key_cmd	88
ehsm_create_dh_key_cmd	90
ehsm_create_dh_key_param	94
ehsm_create_dh_sm2_ext_param	97
ehsm_create_evita_key_param	99
ehsm_create_random_key_param	100
ehsm_crypto_key	103
ehsm_crypto_randomgenerate_param	104
ehsm_ctx_block_mgr	105
ehsm_ctx_session_st	106
ehsm_debug_auth_st	107
ehsm_debug_authentication_cmd	108
ehsm_derive_key_cmd	110
ehsm_dh_param	114
ehsm_dh_param_size_info	114
ehsm_dh_prikey	115
ehsm_dh_pubkey	116
ehsm_ecc_key_size_	117
ehsm_ecc_pubkey	118
ehsm_emu_status_st	118
ehsm_evita_key_export	119
ehsm_evita_key_import_st	122
ehsm_evita_memory_info_st	124
ehsm_exchange_sm9_key_param	126
ehsm_export_key_cmd	129
ehsm_export_pub_key_	131
ehsm_external_key_	133
ehsm_fast_cmac_st	133
ehsm_fw_encrypt_key	136
ehsm_fw_encrypt_key_cmd	137
ehsm_fw_get_random_key_cmd	138
ehsm_fw_random_key	139
ehsm_gen_dh_key_param_st	140
ehsm_gen_key_cmd	142
ehsm_gen_key_param_st	145
ehsm_gen_sm9_key_param	146
ehsm_gen_sm9_master_key_param	148
ehsm_gen_sm9_userpriv_key_cmd	148
ehsm_gen_sm9_userpriv_key_param	150

ehsm_get_challenge_cmd	152
ehsm_get_challenge_st	152
ehsm_get_emu_cmd	153
ehsm_get_emu_status_param_st	155
ehsm_get_pub_from_priv_cmd	156
ehsm_get_pub_from_priv_param	158
ehsm_get_she_id_cmd	160
ehsm_image	161
ehsm_image_upgrade_cmd	163
ehsm_image_verfiy_cmd	166
ehsm_image_verify_st	169
ehsm_import_key_cmd	171
ehsm_internal_key_	173
ehsm_key_attr_data_	175
ehsm_key_copy_param	176
ehsm_key_derived_param	178
ehsm_key_flags_element_st	181
ehsm_key_remove_cmd	182
ehsm_key_remove_param	183
ehsm_key_signature_	184
ehsm_key_status_	186
ehsm_key_status_cmd	188
ehsm_key_status_param	190
ehsm_key_usages_st	191
ehsm_keyexchange_key_info	194
ehsm_low_power_cmd	194
ehsm_mailbox_req	195
ehsm_mbox_cancel_channel_req	203
ehsm_mbox_cancel_channel_rps	204
ehsm_mbox_mgr_channel_req	206
ehsm_module_status_cmd	207
ehsm_module_status_st	
The structure to store the information of ehsm status. The parameter alg, key_handle, key_auth_size, key_auth_value, sign_size and sign are only valid when type is not EHSM_GET_STATUS_SHE	
	210
ehsm_otp_read_cmd	212
ehsm_otp_read_param_st	213
ehsm_otp_write_cmd	214
ehsm_otp_write_param_st	215
ehsm_prikey_data_	216
ehsm_pub_key_	217
ehsm_pubkey_data_	218
ehsm_rng_generate_cmd	219
ehsm_rsa crt_param_	222
ehsm_rsa_dh_key_size_	223
ehsm_rsa_key_size_	224
ehsm_rsa_pubkey	225
ehsm_se_key_	226
ehsm_secure_boot_st	228
ehsm_self_test_cmd	231
ehsm_sensor_init_param_st	231
ehsm_sensor_resp_init_cmd	232
ehsm_service	233
ehsm_service_info	234
ehsm_set_baudrate_cmd	235
ehsm_she_get_id_param_st	236
ehsm_she_key_host_param	237
ehsm_she_key_param	239
ehsm_she_key_st	240
ehsm_she_load_export_key_cmd	242
ehsm_she_load_plain_key_cmd	243
ehsm_she_plain_key_host_param	244

ehsm_she_plain_key_param	244
ehsm_sm9_exchg_gen_usertmp_cmd	245
ehsm_sm9_exchg_key_cmd	247
ehsm_sm9_exchg_key_cmd_child	249
ehsm_sm9_exkey_gen_tmpkey_param	251
ehsm_sm9_export_key_cmd	252
ehsm_sm9_gen_mast_pubkey	254
ehsm_sm9_gen_tmp_pubkey_param	255
ehsm_sm9_get_mast_pubkey_cmd	256
ehsm_sm9_get_tmp_pubkey_cmd	257
ehsm_sm9_import_key_cmd	259
ehsm_sm9_inexport_key_param	261
ehsm_sm9_remove_key_cmd	262
ehsm_sm9_unwrap_key_cmd	263
ehsm_sm9_unwrap_key_param	265
ehsm_sm9_wrap_key_cmd	267
ehsm_sm9_wrap_key_param	270
ehsm_soc_image_verify_cmd	271
ehsm_soc_image_verify_st	275
ehsm_soc_secure_boot_status_st	278
ehsm_storage_area_param_st	278
ehsm_sym_key_size_	279
ehsm_tick_value	280
ehsm_uart_cmd	
Struct for command of getting challenge, the command is from uart or JTAG channel of mailbox	281
hash_hmac_t	282
hash_testvec	282
HSM_AsymCfgType	
Specifies the asymmetric algo config struct	285
HSM_BootCfgType	
Specifies the secureboot information	287
HSM_CMacCfgType	
Specifies the Cipher Mac algo config struct	291
HSM_DebugAuthConfigType	
Deug auth config struct	292
HSM_DeriveKeyCfgType	
Derive Key config struct	294
HSM_DhParamType	
Dh algo param	296
HSM_DhPriKeyType	
Dh private key struct	297
Hsm_DhPubKeyType_	
Dh pubkey struct	297
HSM_EccPubKeyType	
Ecc pubkey struct	298
HSM_FlashKeyPageType	
Specifies the flash Key content,size is DFLASH_PAGE_SIZE	299
HSM_FlashKeyType	
Specifies the Key info content,size is 56Bytes	300
HSM_GenKeyCfgType	
Generate random Key config struct by random generator in hsm	301
HSM_HMacCfgType	
Specifies the Hash Mac algo config struct	303
HSM_ImageVerifyType__	
Host image verify information	304
HSM_InOutMacType	
Input and output struct about gen/ver Mac interface	307
HSM_InOutSignType	
Input and output struct about gen/ver signature interface	309
HSM_InOutType	
Basic Input and output struct	310

HSM_KeyActUseFlagsType	
Key active attribute config struct	311
HSM_KeyFlagsElementType	
Key flags	313
HSM_KeyHandleInfoType	
Specifies the Key handle info content,size is 40Bytes	315
HSM_KeyIndexInfoType	
Specifies the Key index info content,size is 8Bytes	316
HSM_KeySlotInfoType	
Specifies the Key slot info content,size is 8Bytes	317
HSM_KeyStatusType	
Key status in hsm	318
HSM_KeyUsagesType	
Key attribute config struct	319
HSM_PlainKeyCfgType	
Set plain key config struct	321
Hsm_PriKeyDataType_	
Private key struct for all algo	325
Hsm_PubKeyDataType_	
Public key struct for all algo	327
HSM_RamKeyHeadType	
Specifies the Ram Key space header	328
HSM_RamKeyInfoType	
Specifies the Ram Key info	329
Hsm_RsaCrtType_	
Rsa private key struct when CRT	330
HSM_RsaPubKeyType	
Rsa pubkey struct	331
HSM_SecretKeyCfgType	
Secret Key config struct for key import or export by secret	332
HSM_SecureUpgradeType_	
Host image secure upgrade information	334
HSM_SymCfgType	
Specifies the symmetric algo config struct	339
kdf_testvec	341
key_act_use_flags_t	343
key_info_st	346
kpp_testvec	
Test struct for shared secret	347
mac_t	351
mailbox_channel	352
ehsm_cmd_cipher_st::osr_cmd_hdr_u	354
rsacipher_testvec	356
signature_t	361
sm2_ext_param	362
sm9cipher_testvec	364
soc_image_upgrade_info	367
soc_image_upgrade_input	372
soc_image_verify_info	378
soc_image_verify_input	381

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_HSM.pdf	385
AC784xx_Hsm_Reg.h	
This file provides HSM hardware integration functions	385
Asr_Standard_Types.h	391
eHSM_Com_Struct_lp.h	408
eHSM_Compt_Bitmap.c	432
eHSM_Compt_Bitmap.h	435
eHSM_Compt_List.h	438
eHSM_Config_lp.h	440
eHSM_Debug_lp.h	485
eHSM_Dspt_CryObj_lp.c	496
eHSM_Dspt_CryObj_lp.h	498
eHSM_Dspt_lp.c	502
eHSM_Dspt_lp.h	504
eHSM_Err_Code_lp.h	506
eHSM_Exclusive_Area.h	543
eHSM_If_Asr_Cipher_lp.c	544
eHSM_If_Asr_Cipher_lp.h	548
eHSM_If_Asr_ErrCode_lp.c	551
eHSM_If_Asr_ErrCode_lp.h	552
eHSM_If_Asr_lp.h	552
eHSM_If_Asr_Job_lp.c	566
eHSM_If_Asr_Job_lp.h	568
eHSM_If_Asr_Key_lp.c	569
eHSM_If_Asr_Key_lp.h	569
eHSM_If_Asr_KeyCfg_lp.c	571
eHSM_If_Asr_KeyCfg_lp.h	580
eHSM_If_Asr_Types_lp.h	602
eHSM_If_Evita_AsymCper_lp.c	619
eHSM_If_Evita_Counter_lp.c	624
eHSM_If_Evita_ErrCode_lp.c	624
eHSM_If_Evita_ErrCode_lp.h	626
eHSM_If_Evita_Hash_lp.c	628
eHSM_If_Evita_lp.h	632
eHSM_If_Evita_Key_lp.c	664
eHSM_If_Evita_Rng_lp.c	674
eHSM_If_Evita_SymCper_lp.c	676
eHSM_If_Evita_Timer_lp.c	684

eHSM_Ic_Evita_Types_Ip.h	685
eHSM_Ic_Ext_Ip.h	713
eHSM_Ic_Ext_Sm9_Ip.c	737
eHSM_Ic_Ext_SysMgr_Ip.c	738
eHSM_Ic_Ext_Types_Ip.h	753
eHSM_Ic_She_ErrCode_Ip.c	755
eHSM_Ic_She_ErrCode_Ip.h	756
eHSM_Ic_She_Ip.c	756
eHSM_Ic_She_Ip.h	771
eHSM_Ic_She_Types_Ip.h	784
eHSM_IntCfg_Ip.h	791
eHSM_Mailbox_CmdId_Ip.h	824
eHSM_Mailbox_Ip.c	844
eHSM_Mailbox_Ip.h	847
eHSM_Mailbox_Prtcl_Ip.h	849
eHSM_Mailbox_Reg_Ip.h	866
eHSM_Mgr_Ctx_Ip.c	876
eHSM_Mgr_Ctx_Ip.h	879
ehsm_register_addr.h	883
eHSM_Srv_AsymCper_Ip.c	883
eHSM_Srv_Ciper_Ip.c	883
eHSM_Srv_Cipher_Ip.h	885
eHSM_Srv_CmdReq_Ip.h	886
eHSM_Srv_Counter_Ip.c	889
eHSM_Srv_Ext_Ip.c	889
eHSM_Srv_Hash_Ip.c	897
eHSM_Srv_Key_Ip.c	898
eHSM_Srv_Mgr_Ip.c	903
eHSM_Srv_Mgr_Ip.h	914
eHSM_Srv_Rng_Ip.c	919
eHSM_Srv_Timer_Ip.c	920
eHSM_Types_Ip.h	920
Hsm_Hal.c	
This file provides HSM integration functions	923
Hsm_Hal.h	
This file provides hsm integration functions interface	947
test_int_config.h	994
utest_vecs_st.h	994

Chapter 3

Class Documentation

3.1 aead_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- const char * [key](#)
- const char * [iv](#)
- const char * [ptext](#)
- const char * [assoc](#)
- const char * [ctext](#)
- unsigned char [novrfy](#)
- unsigned char [wk](#)
- unsigned char [klen](#)
- unsigned int [plen](#)
- unsigned int [clen](#)
- unsigned int [alen](#)
- unsigned int [ivlen](#)
- int [setkey_error](#)
- int [setauthsize_error](#)
- int [crypt_error](#)

3.1.1 Detailed Description

Definition at line 94 of file `utest_vecs_st.h`.

3.1.2 Member Data Documentation

3.1.2.1 alen

```
unsigned int aead_testvec::alen
```

Definition at line 105 of file `utest_vecs_st.h`.

3.1.2.2 assoc

```
const char* aead_testvec::assoc
```

Definition at line 98 of file `utest_vecs_st.h`.

3.1.2.3 clen

```
unsigned int aead_testvec::clen
```

Definition at line 104 of file `utest_vecs_st.h`.

3.1.2.4 crypt_error

```
int aead_testvec::crypt_error
```

Definition at line 109 of file `utest_vecs_st.h`.

3.1.2.5 ctext

```
const char* aead_testvec::ctext
```

Definition at line 99 of file `utest_vecs_st.h`.

3.1.2.6 iv

```
const char* aead_testvec::iv
```

Definition at line 96 of file `utest_vecs_st.h`.

3.1.2.7 ivlen

```
unsigned int aead_testvec::ivlen
```

Definition at line 106 of file `utest_vecs_st.h`.

3.1.2.8 key

```
const char* aead_testvec::key
```

Definition at line 95 of file `utest_vecs_st.h`.

3.1.2.9 klen

```
unsigned char aead_testvec::klen
```

Definition at line 102 of file `utest_vecs_st.h`.

3.1.2.10 novrfy

```
unsigned char aead_testvec::novrfy
```

Definition at line 100 of file `utest_vecs_st.h`.

3.1.2.11 plen

```
unsigned int aead_testvec::plen
```

Definition at line 103 of file `utest_vecs_st.h`.

3.1.2.12 ptext

```
const char* aead_testvec::ptext
```

Definition at line 97 of file `utest_vecs_st.h`.

3.1.2.13 setauthsize_error

```
int aead_testvec::setauthsize_error
```

Definition at line 108 of file `utest_vecs_st.h`.

3.1.2.14 setkey_error

```
int aead_testvec::setkey_error
```

Definition at line 107 of file `utest_vecs_st.h`.

3.1.2.15 wk

```
unsigned char aead_testvec::wk
```

Definition at line 101 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.2 akcipher_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- const unsigned char * [key](#)
- const unsigned char * [params](#)
- const unsigned char * [m](#)
- const unsigned char * [c](#)
- unsigned int [key_len](#)
- unsigned int [param_len](#)
- unsigned int [m_size](#)
- unsigned int [c_size](#)
- unsigned char [public_key_vec](#)
- unsigned char [siggen_sigver_test](#)
- unsigned char [hash_alg](#)

3.2.1 Detailed Description

Parameters

<i>key</i>	Public key only or contain private key. Depending on <code>public_key_vec</code> , if <code>public_key_vec</code> is true. This is only the public key, otherwise its the public key and private key (private key is first). The keys are just in raw data. No '0x04' is put for public key. If the code need '0x04', please add it in test code. : Random generated number for signature. : Original message. : Signature in (r, s) mode. : Size of key in bytes. : Size of in bytes. : Size of message in bytes. : Size of signature in bytes. : Is the key only contain public key. : Test for signature and verification if ture. Otherwise, test for encryption and decryption. : Hash algorithm.
------------	--

Definition at line 130 of file `utest_vecs_st.h`.

3.2.2 Member Data Documentation

3.2.2.1 c

```
const unsigned char* akcipher_testvec::c
```

Definition at line 134 of file `utest_vecs_st.h`.

3.2.2.2 c_size

```
unsigned int akcipher_testvec::c_size
```

Definition at line 138 of file `utest_vecs_st.h`.

3.2.2.3 hash_alg

```
unsigned char akcipher_testvec::hash_alg
```

Definition at line 141 of file `utest_vecs_st.h`.

3.2.2.4 key

```
const unsigned char* akcipher_testvec::key
```

Definition at line 131 of file `utest_vecs_st.h`.

3.2.2.5 key_len

```
unsigned int akcipher_testvec::key_len
```

Definition at line 135 of file `utest_vecs_st.h`.

3.2.2.6 m

```
const unsigned char* akcipher_testvec::m
```

Definition at line 133 of file `utest_vecs_st.h`.

3.2.2.7 m_size

```
unsigned int akcipher_testvec::m_size
```

Definition at line 137 of file `utest_vecs_st.h`.

3.2.2.8 param_len

```
unsigned int akcipher_testvec::param_len
```

Definition at line 136 of file `utest_vecs_st.h`.

3.2.2.9 params

```
const unsigned char* akcipher_testvec::params
```

Definition at line 132 of file `utest_vecs_st.h`.

3.2.2.10 public_key_vec

```
unsigned char akcipher_testvec::public_key_vec
```

Definition at line 139 of file `utest_vecs_st.h`.

3.2.2.11 siggen_sigver_test

```
unsigned char akcipher_testvec::siggen_sigver_test
```

Definition at line 140 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.3 bitmap Struct Reference

```
#include <eHSM_Compt_Bitmap.h>
```


Public Attributes

- [ehsm_uint32_t bit_max](#)
- [ehsm_uint32_t bit_bytes](#)
- [ehsm_uint8_t bit_map \[0\]](#)

3.3.1 Detailed Description

Definition at line 30 of file eHSM_Compt_Bitmap.h.

3.3.2 Member Data Documentation

3.3.2.1 bit_bytes

```
ehsm_uint32_t bitmap::bit_bytes
```

Definition at line 32 of file eHSM_Compt_Bitmap.h.

3.3.2.2 bit_map

```
ehsm_uint8_t bitmap::bit_map[0]
```

Definition at line 33 of file eHSM_Compt_Bitmap.h.

3.3.2.3 bit_max

```
ehsm_uint32_t bitmap::bit_max
```

Definition at line 31 of file eHSM_Compt_Bitmap.h.

The documentation for this struct was generated from the following file:

- [eHSM_Compt_Bitmap.h](#)

3.4 cipher_session_st Struct Reference

```
#include <eHSM_Mgr_Ctx_Ip.h>
```

Public Attributes

- [ehsm_uint32_t cmd_id](#)
- [ehsm_uint8_t direction](#)
- [ehsm_uint8_t padding](#)
- [ehsm_uint8_t algorithm](#)
- [ehsm_uint8_t cipher_mode](#)
- [ehsm_uint8_t time_stamp](#)
- [ehsm_uint8_t is_hmac](#)
- [ehsm_uint8_t mac_bytes](#)
- [ehsm_uint8_t rsa_crt_mode](#)
- [ehsm_uint8_t key_type](#)
- [ehsm_uint32_t key_handle](#)
- [ehsm_uint32_t key_auth_addr](#)
- [ehsm_uint32_t key_auth_size](#)
- [ehsm_uint32_t signature_addr](#)
- [ehsm_uint32_t signature_size](#)
- [ehsm_uint32_t job_id](#)
- [ehsm_aead_data_ptr_st aead_data](#)
- [ehsm_utc_time_t utc_time](#)
- [ehsm_uint32_t status](#)
- [ehsm_uint8_t ctx \[EHSM_CONTEXT_SIZE\]](#)
- [ehsm_ctx_block_mgr_st ctx_block_mgr](#)

3.4.1 Detailed Description

Definition at line 52 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2 Member Data Documentation

3.4.2.1 aead_data

[ehsm_aead_data_ptr_st](#) cipher_session_st::aead_data

Definition at line 70 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.2 algorithm

[ehsm_uint8_t](#) cipher_session_st::algorithm

Definition at line 57 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.3 cipher_mode

`ehsm_uint8_t cipher_session_st::cipher_mode`

Definition at line 58 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.4 cmd_id

`ehsm_uint32_t cipher_session_st::cmd_id`

Definition at line 54 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.5 ctx

`ehsm_uint8_t cipher_session_st::ctx[EHSM_CONTEXT_SIZE]`

Definition at line 73 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.6 ctx_block_mgr

`ehsm_ctx_block_mgr_st cipher_session_st::ctx_block_mgr`

Definition at line 74 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.7 direction

`ehsm_uint8_t cipher_session_st::direction`

Definition at line 55 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.8 is_hmac

`ehsm_uint8_t cipher_session_st::is_hmac`

Definition at line 60 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.9 job_id

`ehsm_uint32_t cipher_session_st::job_id`

Definition at line 69 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.10 key_auth_addr

`ehsm_uint32_t cipher_session_st::key_auth_addr`

Definition at line 65 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.11 key_auth_size

`ehsm_uint32_t cipher_session_st::key_auth_size`

Definition at line 66 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.12 key_handle

`ehsm_uint32_t cipher_session_st::key_handle`

Definition at line 64 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.13 key_type

`ehsm_uint8_t cipher_session_st::key_type`

Definition at line 63 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.14 mac_bytes

`ehsm_uint8_t cipher_session_st::mac_bytes`

Definition at line 61 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.15 padding

`ehsm_uint8_t cipher_session_st::padding`

Definition at line 56 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.16 rsa_cert_mode

`ehsm_uint8_t cipher_session_st::rsa_cert_mode`

Definition at line 62 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.17 signature_addr

`ehsm_uint32_t cipher_session_st::signature_addr`

Definition at line 67 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.18 signature_size

`ehsm_uint32_t cipher_session_st::signature_size`

Definition at line 68 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.19 status

`ehsm_uint32_t cipher_session_st::status`

Definition at line 72 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.20 time_stamp

`ehsm_uint8_t cipher_session_st::time_stamp`

Definition at line 59 of file eHSM_Mgr_Ctx_Ip.h.

3.4.2.21 utc_time

```
ehsm_utc_time_t cipher_session_st::utc_time
```

Definition at line 71 of file eHSM_Mgr_Ctx_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mgr_Ctx_Ip.h](#)

3.5 cipher_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- const char * [key](#)
- const char * [iv](#)
- const char * [iv_out](#)
- const char * [ptext](#)
- const char * [ctext](#)
- unsigned char [wk](#)
- unsigned short [klen](#)
- unsigned int [len](#)
- int [fips_skip](#)
- int [setkey_error](#)
- int [crypt_error](#)
- unsigned int [clen](#)

3.5.1 Detailed Description

Definition at line 51 of file utest_vecs_st.h.

3.5.2 Member Data Documentation

3.5.2.1 clen

```
unsigned int cipher_testvec::clen
```

Definition at line 65 of file utest_vecs_st.h.

3.5.2.2 crypt_error

```
int cipher_testvec::crypt_error
```

Definition at line 63 of file utest_vecs_st.h.

3.5.2.3 ctext

```
const char* cipher_testvec::ctext
```

Definition at line 56 of file utest_vecs_st.h.

3.5.2.4 fips_skip

```
int cipher_testvec::fips_skip
```

Definition at line 61 of file utest_vecs_st.h.

3.5.2.5 iv

```
const char* cipher_testvec::iv
```

Definition at line 53 of file utest_vecs_st.h.

3.5.2.6 iv_out

```
const char* cipher_testvec::iv_out
```

Definition at line 54 of file utest_vecs_st.h.

3.5.2.7 key

```
const char* cipher_testvec::key
```

Definition at line 52 of file utest_vecs_st.h.

3.5.2.8 klen

```
unsigned short cipher_testvec::klen
```

Definition at line 59 of file `utest_vecs_st.h`.

3.5.2.9 len

```
unsigned int cipher_testvec::len
```

Definition at line 60 of file `utest_vecs_st.h`.

3.5.2.10 ptext

```
const char* cipher_testvec::ptext
```

Definition at line 55 of file `utest_vecs_st.h`.

3.5.2.11 setkey_error

```
int cipher_testvec::setkey_error
```

Definition at line 62 of file `utest_vecs_st.h`.

3.5.2.12 wk

```
unsigned char cipher_testvec::wk
```

Definition at line 58 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.6 counter_value_64_t Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```


Public Attributes

- [ehsm_uint32_t high_word](#)
- [ehsm_uint32_t low_word](#)

3.6.1 Detailed Description

Definition at line 419 of file eHSM_If_Evita_Types_Ip.h.

3.6.2 Member Data Documentation

3.6.2.1 high_word

[ehsm_uint32_t](#) counter_value_64_t::high_word

Definition at line 420 of file eHSM_If_Evita_Types_Ip.h.

3.6.2.2 low_word

[ehsm_uint32_t](#) counter_value_64_t::low_word

Definition at line 421 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.7 Crypto_AlgorithmInfoType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- [Crypto_AlgorithmFamilyType](#) family
- [Crypto_AlgorithmFamilyType](#) secondaryFamily
- [ehsm_uint32_t](#) keyLength
- [Crypto_AlgorithmModeType](#) mode

3.7.1 Detailed Description

Definition at line 315 of file Asr_Standard_Types.h.

3.7.2 Member Data Documentation

3.7.2.1 family

[Crypto_AlgorithmFamilyType](#) `Crypto_AlgorithmInfoType::family`

Definition at line 317 of file `Asr_Standard_Types.h`.

3.7.2.2 keyLength

[ehsm_uint32_t](#) `Crypto_AlgorithmInfoType::keyLength`

Definition at line 319 of file `Asr_Standard_Types.h`.

3.7.2.3 mode

[Crypto_AlgorithmModeType](#) `Crypto_AlgorithmInfoType::mode`

Definition at line 320 of file `Asr_Standard_Types.h`.

3.7.2.4 secondaryFamily

[Crypto_AlgorithmFamilyType](#) `Crypto_AlgorithmInfoType::secondaryFamily`

Definition at line 318 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.8 crypto_copy_key_dh_key_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * p
- [ehsm_uint32_t](#) p_size
- [ehsm_uint8_t](#) * q
- [ehsm_uint32_t](#) q_size
- [ehsm_uint8_t](#) * g
- [ehsm_uint32_t](#) g_size

3.8.1 Detailed Description

Definition at line 337 of file eHSM_Lf_Asr_Types_lp.h.

3.8.2 Member Data Documentation

3.8.2.1 g

`ehsm_uint8_t* crypto_copy_key_dh_key_info::g`

Definition at line 343 of file eHSM_Lf_Asr_Types_lp.h.

3.8.2.2 g_size

`ehsm_uint32_t crypto_copy_key_dh_key_info::g_size`

Definition at line 344 of file eHSM_Lf_Asr_Types_lp.h.

3.8.2.3 p

`ehsm_uint8_t* crypto_copy_key_dh_key_info::p`

Definition at line 339 of file eHSM_Lf_Asr_Types_lp.h.

3.8.2.4 p_size

`ehsm_uint32_t crypto_copy_key_dh_key_info::p_size`

Definition at line 340 of file eHSM_Lf_Asr_Types_lp.h.

3.8.2.5 q

`ehsm_uint8_t* crypto_copy_key_dh_key_info::q`

Definition at line 341 of file eHSM_Lf_Asr_Types_lp.h.

3.8.2.6 q_size

`ehsm_uint32_t crypto_copy_key_dh_key_info::q_size`

Definition at line 342 of file `eHSM_If_Asr_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.9 crypto_copy_key_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t key_handle](#)
- [ehsm_uint32_t auth_size](#)
- [ehsm_uint8_t * auth_value](#)
- [ehsm_uint32_t element_size](#)
- [ehsm_key_flags_element_st element_data \[CRYPTO_MAX_KEY_FLAG_ELEMENT\]](#)

3.9.1 Detailed Description

Definition at line 324 of file `eHSM_If_Asr_Types_Ip.h`.

3.9.2 Member Data Documentation

3.9.2.1 auth_size

`ehsm_uint32_t crypto_copy_key_info::auth_size`

Definition at line 329 of file `eHSM_If_Asr_Types_Ip.h`.

3.9.2.2 auth_value

`ehsm_uint8_t* crypto_copy_key_info::auth_value`

Definition at line 331 of file `eHSM_If_Asr_Types_Ip.h`.

3.9.2.3 element_data

```
ehsm_key_flags_element_st crypto_copy_key_info::element_data[CRYPTO_MAX_KEY_FLAG_ELEMENT]
```

Definition at line 333 of file eHSM_If_Asr_Types_Ip.h.

3.9.2.4 element_size

```
ehsm_uint32_t crypto_copy_key_info::element_size
```

Definition at line 332 of file eHSM_If_Asr_Types_Ip.h.

3.9.2.5 key_handle

```
ehsm_uint32_t crypto_copy_key_info::key_handle
```

Definition at line 327 of file eHSM_If_Asr_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.10 crypto_create_evita_key_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_size
- [ehsm_uint32_t](#) valid_until
- [ehsm_uint8_t](#) type
- [ehsm_uint32_t](#) element_size
- [ehsm_key_flags_element_st](#) element_data [CRYPTO_MAX_KEY_FLAG_ELEMENT]

3.10.1 Detailed Description

Definition at line 217 of file eHSM_If_Asr_Types_Ip.h.

3.10.2 Member Data Documentation

3.10.2.1 element_data

[ehsm_key_flags_element_st](#) [crypto_create_evita_key_info::element_data](#)[[CRYPTO_MAX_KEY_FLAG_ELEMENT](#)]

Definition at line 226 of file [eHSM_If_Asr_Types_Ip.h](#).

3.10.2.2 element_size

[ehsm_uint32_t](#) [crypto_create_evita_key_info::element_size](#)

Definition at line 225 of file [eHSM_If_Asr_Types_Ip.h](#).

3.10.2.3 key_size

[ehsm_uint32_t](#) [crypto_create_evita_key_info::key_size](#)

Definition at line 220 of file [eHSM_If_Asr_Types_Ip.h](#).

3.10.2.4 type

[ehsm_uint8_t](#) [crypto_create_evita_key_info::type](#)

Definition at line 224 of file [eHSM_If_Asr_Types_Ip.h](#).

3.10.2.5 valid_until

[ehsm_uint32_t](#) [crypto_create_evita_key_info::valid_until](#)

Definition at line 222 of file [eHSM_If_Asr_Types_Ip.h](#).

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.11 crypto_evita_key_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) auth_size
- [ehsm_uint8_t](#) * auth_value

3.11.1 Detailed Description

Definition at line 231 of file eHSM_If_Asr_Types_Ip.h.

3.11.2 Member Data Documentation

3.11.2.1 auth_size

```
ehsm_uint32_t crypto_evita_key_info::auth_size
```

Definition at line 236 of file eHSM_If_Asr_Types_Ip.h.

3.11.2.2 auth_value

```
ehsm_uint8_t* crypto_evita_key_info::auth_value
```

Definition at line 238 of file eHSM_If_Asr_Types_Ip.h.

3.11.2.3 key_handle

```
ehsm_uint32_t crypto_evita_key_info::key_handle
```

Definition at line 234 of file eHSM_If_Asr_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.12 crypto_exported_key Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * encrypted_key
- [ehsm_uint32_t](#) encrypted_key_buffer_szie
- [ehsm_uint32_t](#) encrypted_key_size
- [ehsm_uint8_t](#) * key_auth_code
- [ehsm_uint32_t](#) key_auth_code_buffer_size
- [ehsm_uint32_t](#) key_auth_code_size

3.12.1 Detailed Description

Definition at line 280 of file eHSM_If_Asr_Types_lp.h.

3.12.2 Member Data Documentation

3.12.2.1 encrypted_key

[ehsm_uint8_t](#)* crypto_exported_key::encrypted_key

Definition at line 283 of file eHSM_If_Asr_Types_lp.h.

3.12.2.2 encrypted_key_buffer_szie

[ehsm_uint32_t](#) crypto_exported_key::encrypted_key_buffer_szie

Definition at line 285 of file eHSM_If_Asr_Types_lp.h.

3.12.2.3 encrypted_key_size

[ehsm_uint32_t](#) crypto_exported_key::encrypted_key_size

Definition at line 287 of file eHSM_If_Asr_Types_lp.h.

3.12.2.4 key_auth_code

[ehsm_uint8_t](#)* crypto_exported_key::key_auth_code

Definition at line 289 of file eHSM_If_Asr_Types_lp.h.

3.12.2.5 key_auth_code_buffer_size

[ehsm_uint32_t](#) crypto_exported_key::key_auth_code_buffer_size

Definition at line 291 of file eHSM_If_Asr_Types_Ip.h.

3.12.2.6 key_auth_code_size

[ehsm_uint32_t](#) crypto_exported_key::key_auth_code_size

Definition at line 293 of file eHSM_If_Asr_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.13 crypto_import_evita_key_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) transport_key_handle
- [ehsm_uint32_t](#) transport_key_authorization_size
- [ehsm_uint8_t](#) * transport_key_authorization
- [ehsm_uint32_t](#) authenticity_key_handle
- [ehsm_uint32_t](#) authenticity_key_authorization_size
- [ehsm_uint8_t](#) * authenticity_key_authorization
- [ehsm_key_mem_type_e](#) type
- [ehsm_uint32_t](#) encrypted_key_size
- [ehsm_uint8_t](#) * encrypted_key
- [ehsm_uint32_t](#) key_authenticity_code_size
- [ehsm_uint8_t](#) * key_authenticity_code

3.13.1 Detailed Description

Definition at line 242 of file eHSM_If_Asr_Types_Ip.h.

3.13.2 Member Data Documentation

3.13.2.1 authenticity_key_authorization

`ehsm_uint8_t*` `crypto_import_evita_key_info::authenticity_key_authorization`

Definition at line 249 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.2 authenticity_key_authorization_size

`ehsm_uint32_t` `crypto_import_evita_key_info::authenticity_key_authorization_size`

Definition at line 248 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.3 authenticity_key_handle

`ehsm_uint32_t` `crypto_import_evita_key_info::authenticity_key_handle`

Definition at line 247 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.4 encrypted_key

`ehsm_uint8_t*` `crypto_import_evita_key_info::encrypted_key`

Definition at line 252 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.5 encrypted_key_size

`ehsm_uint32_t` `crypto_import_evita_key_info::encrypted_key_size`

Definition at line 251 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.6 key_authenticity_code

`ehsm_uint8_t*` `crypto_import_evita_key_info::key_authenticity_code`

Definition at line 254 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.7 key_authenticity_code_size

`ehsm_uint32_t` `crypto_import_evita_key_info::key_authenticity_code_size`

Definition at line 253 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.8 transport_key_authorization

`ehsm_uint8_t*` `crypto_import_evita_key_info::transport_key_authorization`

Definition at line 246 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.9 transport_key_authorization_size

`ehsm_uint32_t` `crypto_import_evita_key_info::transport_key_authorization_size`

Definition at line 245 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.10 transport_key_handle

`ehsm_uint32_t` `crypto_import_evita_key_info::transport_key_handle`

Definition at line 244 of file `eHSM_If_Asr_Types_Ip.h`.

3.13.2.11 type

`ehsm_key_mem_type_e` `crypto_import_evita_key_info::type`

Definition at line 250 of file `eHSM_If_Asr_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.14 Crypto_JobInfoType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- const [ehsm_uint32_t](#) jobId
- const [ehsm_uint32_t](#) jobPriority

3.14.1 Detailed Description

Definition at line 157 of file Asr_Standard_Types.h.

3.14.2 Member Data Documentation

3.14.2.1 jobId

```
const ehsm\_uint32\_t Crypto_JobInfoType::jobId
```

Definition at line 159 of file Asr_Standard_Types.h.

3.14.2.2 jobPriority

```
const ehsm\_uint32\_t Crypto_JobInfoType::jobPriority
```

Definition at line 160 of file Asr_Standard_Types.h.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.15 Crypto_JobPrimitiveInfoType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- [ehsm_uint32_t](#) callbackId
- const [Crypto_PrimitiveInfoType](#) * primitiveInfo
- [ehsm_uint32_t](#) crylfKeyId
- [Crypto_ProcessingType](#) processingType
- [ehsm_bool_t](#) callbackUpdateNotification

3.15.1 Detailed Description

Definition at line 345 of file Asr_Standard_Types.h.

3.15.2 Member Data Documentation

3.15.2.1 callbackId

`ehsm_uint32_t` `Crypto_JobPrimitiveInfoType::callbackId`

Definition at line 347 of file `Asr_Standard_Types.h`.

3.15.2.2 callbackUpdateNotification

`ehsm_bool_t` `Crypto_JobPrimitiveInfoType::callbackUpdateNotification`

Definition at line 355 of file `Asr_Standard_Types.h`.

3.15.2.3 cryIfKeyId

`ehsm_uint32_t` `Crypto_JobPrimitiveInfoType::cryIfKeyId`

Definition at line 353 of file `Asr_Standard_Types.h`.

3.15.2.4 primitiveInfo

`const` `Crypto_PrimitiveInfoType*` `Crypto_JobPrimitiveInfoType::primitiveInfo`

Definition at line 351 of file `Asr_Standard_Types.h`.

3.15.2.5 processingType

`Crypto_ProcessingType` `Crypto_JobPrimitiveInfoType::processingType`

Definition at line 354 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.16 Crypto_JobPrimitiveInputOutputType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- const ehsm_uint8_t * inputPtr
- ehsm_uint32_t inputLength
- const ehsm_uint8_t * secondaryInputPtr
- ehsm_uint32_t secondaryInputLength
- const ehsm_uint8_t * tertiaryInputPtr
- ehsm_uint32_t tertiaryInputLength
- ehsm_uint8_t * outputPtr
- ehsm_uint32_t * outputLengthPtr
- ehsm_uint8_t * secondaryOutputPtr
- ehsm_uint32_t * secondaryOutputLengthPtr
- ehsm_uint64_t input64
- Crypto_VerifyResultType * verifyPtr
- ehsm_uint64_t * output64Ptr
- Crypto_OperationModeType mode
- ehsm_uint32_t cryIfKeyId
- ehsm_uint32_t targetCryIfKeyId

3.16.1 Detailed Description

Definition at line 294 of file Asr_Standard_Types.h.

3.16.2 Member Data Documentation

3.16.2.1 cryIfKeyId

`ehsm_uint32_t` Crypto_JobPrimitiveInputOutputType::cryIfKeyId

Definition at line 310 of file Asr_Standard_Types.h.

3.16.2.2 input64

`ehsm_uint64_t` Crypto_JobPrimitiveInputOutputType::input64

Definition at line 306 of file Asr_Standard_Types.h.

3.16.2.3 inputLength

`ehsm_uint32_t` Crypto_JobPrimitiveInputOutputType::inputLength

Definition at line 297 of file Asr_Standard_Types.h.

3.16.2.4 inputPtr

```
const ehsm_uint8_t* Crypto_JobPrimitiveInputOutputType::inputPtr
```

Definition at line 296 of file Asr_Standard_Types.h.

3.16.2.5 mode

```
Crypto_OperationModeType Crypto_JobPrimitiveInputOutputType::mode
```

Definition at line 309 of file Asr_Standard_Types.h.

3.16.2.6 output64Ptr

```
ehsm_uint64_t* Crypto_JobPrimitiveInputOutputType::output64Ptr
```

Definition at line 308 of file Asr_Standard_Types.h.

3.16.2.7 outputLengthPtr

```
ehsm_uint32_t* Crypto_JobPrimitiveInputOutputType::outputLengthPtr
```

Definition at line 303 of file Asr_Standard_Types.h.

3.16.2.8 outputPtr

```
ehsm_uint8_t* Crypto_JobPrimitiveInputOutputType::outputPtr
```

Definition at line 302 of file Asr_Standard_Types.h.

3.16.2.9 secondaryInputLength

```
ehsm_uint32_t Crypto_JobPrimitiveInputOutputType::secondaryInputLength
```

Definition at line 299 of file Asr_Standard_Types.h.

3.16.2.10 secondaryInputPtr

```
const ehsm_uint8_t* Crypto_JobPrimitiveInputOutputType::secondaryInputPtr
```

Definition at line 298 of file Asr_Standard_Types.h.

3.16.2.11 secondaryOutputLengthPtr

```
ehsm_uint32_t* Crypto_JobPrimitiveInputOutputType::secondaryOutputLengthPtr
```

Definition at line 305 of file Asr_Standard_Types.h.

3.16.2.12 secondaryOutputPtr

```
ehsm_uint8_t* Crypto_JobPrimitiveInputOutputType::secondaryOutputPtr
```

Definition at line 304 of file Asr_Standard_Types.h.

3.16.2.13 targetCryIfKeyId

```
ehsm_uint32_t Crypto_JobPrimitiveInputOutputType::targetCryIfKeyId
```

Definition at line 311 of file Asr_Standard_Types.h.

3.16.2.14 tertiaryInputLength

```
ehsm_uint32_t Crypto_JobPrimitiveInputOutputType::tertiaryInputLength
```

Definition at line 301 of file Asr_Standard_Types.h.

3.16.2.15 tertiaryInputPtr

```
const ehsm_uint8_t* Crypto_JobPrimitiveInputOutputType::tertiaryInputPtr
```

Definition at line 300 of file Asr_Standard_Types.h.

3.16.2.16 verifyPtr

`Crypto_VerifyResultType* Crypto_JobPrimitiveInputOutputType::verifyPtr`

Definition at line 307 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.17 Crypto_JobRedirectionInfoType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- `ehsm_uint8_t` redirectionConfig
- `ehsm_uint32_t` inputKeyId
- `ehsm_uint32_t` inputKeyElementId
- `ehsm_uint32_t` secondaryInputKeyId
- `ehsm_uint32_t` secondaryInputKeyElementId
- `ehsm_uint32_t` tertiaryInputKeyId
- `ehsm_uint32_t` tertiaryInputKeyElementId
- `ehsm_uint32_t` outputKeyId
- `ehsm_uint32_t` outputKeyElementId
- `ehsm_uint32_t` secondaryOutputKeyId
- `ehsm_uint32_t` secondaryOutputKeyElementId

3.17.1 Detailed Description

Definition at line 369 of file `Asr_Standard_Types.h`.

3.17.2 Member Data Documentation

3.17.2.1 inputKeyElementId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::inputKeyElementId`

Definition at line 373 of file `Asr_Standard_Types.h`.

3.17.2.2 inputKeyId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::inputKeyId`

Definition at line 372 of file `Asr_Standard_Types.h`.

3.17.2.3 outputKeyElementId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::outputKeyElementId`

Definition at line 379 of file `Asr_Standard_Types.h`.

3.17.2.4 outputKeyId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::outputKeyId`

Definition at line 378 of file `Asr_Standard_Types.h`.

3.17.2.5 redirectionConfig

`ehsm_uint8_t` `Crypto_JobRedirectionInfoType::redirectionConfig`

Definition at line 371 of file `Asr_Standard_Types.h`.

3.17.2.6 secondaryInputKeyElementId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::secondaryInputKeyElementId`

Definition at line 375 of file `Asr_Standard_Types.h`.

3.17.2.7 secondaryInputKeyId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::secondaryInputKeyId`

Definition at line 374 of file `Asr_Standard_Types.h`.

3.17.2.8 secondaryOutputKeyElementId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::secondaryOutputKeyElementId`

Definition at line 381 of file `Asr_Standard_Types.h`.

3.17.2.9 secondaryOutputKeyId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::secondaryOutputKeyId`

Definition at line 380 of file `Asr_Standard_Types.h`.

3.17.2.10 tertiaryInputKeyElementId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::tertiaryInputKeyElementId`

Definition at line 377 of file `Asr_Standard_Types.h`.

3.17.2.11 tertiaryInputKeyId

`ehsm_uint32_t` `Crypto_JobRedirectionInfoType::tertiaryInputKeyId`

Definition at line 376 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.18 Crypto_JobType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- `ehsm_uint32_t` `jobId`
- `Crypto_JobStateType` `jobState`
- `Crypto_JobPrimitiveInputOutputType` `jobPrimitiveInputOutput`
- `const` `Crypto_JobPrimitiveInfoType` * `jobPrimitiveInfo`
- `const` `Crypto_JobInfoType` * `jobInfo`
- `Crypto_JobRedirectionInfoType` * `jobRedirectionInfoRef`
- `ehsm_uint32_t` `cryptoKeyId`

3.18.1 Detailed Description

Definition at line 385 of file Asr_Standard_Types.h.

3.18.2 Member Data Documentation

3.18.2.1 cryptoKeyId

`ehsm_uint32_t` Crypto_JobType::cryptoKeyId

Definition at line 398 of file Asr_Standard_Types.h.

3.18.2.2 jobId

`ehsm_uint32_t` Crypto_JobType::jobId

Definition at line 387 of file Asr_Standard_Types.h.

3.18.2.3 jobInfo

`const Crypto_JobInfoType*` Crypto_JobType::jobInfo

Definition at line 395 of file Asr_Standard_Types.h.

3.18.2.4 jobPrimitiveInfo

`const Crypto_JobPrimitiveInfoType*` Crypto_JobType::jobPrimitiveInfo

Definition at line 394 of file Asr_Standard_Types.h.

3.18.2.5 jobPrimitiveInputOutput

`Crypto_JobPrimitiveInputOutputType` Crypto_JobType::jobPrimitiveInputOutput

Definition at line 389 of file Asr_Standard_Types.h.

3.18.2.6 jobRedirectionInfoRef

[Crypto_JobRedirectionInfoType*](#) [Crypto_JobType::jobRedirectionInfoRef](#)

Definition at line 397 of file [Asr_Standard_Types.h](#).

3.18.2.7 jobState

[Crypto_JobStateType](#) [Crypto_JobType::jobState](#)

Definition at line 388 of file [Asr_Standard_Types.h](#).

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.19 crypto_key_derive_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [crypto_key_derive_type_e](#) [derive_type](#)
- [crypto_evita_key_info_st](#) [key_info](#)
- [ehsm_uint8_t](#) * [passwd](#)
- [ehsm_uint32_t](#) [passwd_size](#)
- [ehsm_uint32_t](#) [itera_times](#)

3.19.1 Detailed Description

Definition at line 348 of file [eHSM_If_Asr_Types_Ip.h](#).

3.19.2 Member Data Documentation

3.19.2.1 derive_type

[crypto_key_derive_type_e](#) [crypto_key_derive_info::derive_type](#)

Definition at line 350 of file [eHSM_If_Asr_Types_Ip.h](#).

3.19.2.2 itera_times

[ehsm_uint32_t](#) crypto_key_derive_info::itera_times

Definition at line 357 of file eHSM_If_Asr_Types_Ip.h.

3.19.2.3 key_info

[crypto_evita_key_info_st](#) crypto_key_derive_info::key_info

Definition at line 352 of file eHSM_If_Asr_Types_Ip.h.

3.19.2.4 passwd

[ehsm_uint8_t*](#) crypto_key_derive_info::passwd

Definition at line 354 of file eHSM_If_Asr_Types_Ip.h.

3.19.2.5 passwd_size

[ehsm_uint32_t](#) crypto_key_derive_info::passwd_size

Definition at line 355 of file eHSM_If_Asr_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.20 crypto_key_element_type_info_st Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) CryptoKeyId
- [ehsm_bool_t](#) CryptoKeyElementAllowPartialAccess
- [CryptoKeyFormatType](#) CryptoKeyFormat
- [ehsm_bool_t](#) CryptoKeyElementPersist
- [Crypto_KeyElementReadAccessType](#) CryptoKeyElementReadAccess
- [ehsm_uint32_t](#) CryptoKeyElementMaxSize
- [Crypto_KeyElementWriteAccessType](#) CryptoKeyElementWriteAccess

3.20.1 Detailed Description

Definition at line 179 of file eHSM_If_Asr_Types_lp.h.

3.20.2 Member Data Documentation

3.20.2.1 CryptoKeyElementAllowPartialAccess

```
ehsm_bool_t crypto_key_element_type_info_st::CryptoKeyElementAllowPartialAccess
```

Definition at line 183 of file eHSM_If_Asr_Types_lp.h.

3.20.2.2 CryptoKeyElementId

```
ehsm_uint32_t crypto_key_element_type_info_st::CryptoKeyElementId
```

Definition at line 182 of file eHSM_If_Asr_Types_lp.h.

3.20.2.3 CryptoKeyElementMaxSize

```
ehsm_uint32_t crypto_key_element_type_info_st::CryptoKeyElementMaxSize
```

Definition at line 187 of file eHSM_If_Asr_Types_lp.h.

3.20.2.4 CryptoKeyElementPersist

```
ehsm_bool_t crypto_key_element_type_info_st::CryptoKeyElementPersist
```

Definition at line 185 of file eHSM_If_Asr_Types_lp.h.

3.20.2.5 CryptoKeyElementReadAccess

```
Crypto_KeyElementReadAccessType crypto_key_element_type_info_st::CryptoKeyElementReadAccess
```

Definition at line 186 of file eHSM_If_Asr_Types_lp.h.

3.20.2.6 CryptoKeyElementWriteAccess

[CryptoKeyElementWriteAccessType](#) crypto_key_element_type_info_st::CryptoKeyElementWriteAccess

Definition at line 188 of file eHSM_If_Asr_Types_Ip.h.

3.20.2.7 CryptoKeyFormat

[CryptoKeyFormatType](#) crypto_key_element_type_info_st::CryptoKeyFormat

Definition at line 184 of file eHSM_If_Asr_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.21 crypto_key_export_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [key_act_use_flags_t](#) use_flags
- [ehsm_uint32_t](#) transport_key_handle
- [ehsm_uint32_t](#) transport_key_authorization_size
- [ehsm_uint8_t](#) * transport_key_authorization
- [ehsm_uint32_t](#) authenticity_key_handle
- [ehsm_uint32_t](#) authenticity_key_authorization_size
- [ehsm_uint8_t](#) * authenticity_key_authorization

3.21.1 Detailed Description

Definition at line 258 of file eHSM_If_Asr_Types_Ip.h.

3.21.2 Member Data Documentation

3.21.2.1 authenticity_key_authorization

[ehsm_uint8_t](#)* crypto_key_export_info::authenticity_key_authorization

Definition at line 276 of file eHSM_If_Asr_Types_Ip.h.

3.21.2.2 authenticity_key_authorization_size

`ehsm_uint32_t` `crypto_key_export_info::authenticity_key_authorization_size`

Definition at line 274 of file `eHSM_If_Asr_Types_Ip.h`.

3.21.2.3 authenticity_key_handle

`ehsm_uint32_t` `crypto_key_export_info::authenticity_key_handle`

Definition at line 272 of file `eHSM_If_Asr_Types_Ip.h`.

3.21.2.4 key_handle

`ehsm_uint32_t` `crypto_key_export_info::key_handle`

Definition at line 261 of file `eHSM_If_Asr_Types_Ip.h`.

3.21.2.5 transport_key_authorization

`ehsm_uint8_t*` `crypto_key_export_info::transport_key_authorization`

Definition at line 269 of file `eHSM_If_Asr_Types_Ip.h`.

3.21.2.6 transport_key_authorization_size

`ehsm_uint32_t` `crypto_key_export_info::transport_key_authorization_size`

Definition at line 267 of file `eHSM_If_Asr_Types_Ip.h`.

3.21.2.7 transport_key_handle

`ehsm_uint32_t` `crypto_key_export_info::transport_key_handle`

Definition at line 265 of file `eHSM_If_Asr_Types_Ip.h`.

3.21.2.8 use_flags

`key_act_use_flags_t` `crypto_key_export_info::use_flags`

Definition at line 263 of file `eHSM_If_Asr_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.22 crypto_key_status_info Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) `key_handle`
- [ehsm_uint32_t](#) `certification_key_handle`
- [ehsm_uint32_t](#) `certification_key_auth_size`
- [ehsm_uint8_t](#) * `certification_key_auth_value`

3.22.1 Detailed Description

Definition at line 297 of file `eHSM_If_Asr_Types_Ip.h`.

3.22.2 Member Data Documentation

3.22.2.1 certification_key_auth_size

[ehsm_uint32_t](#) `crypto_key_status_info::certification_key_auth_size`

Definition at line 305 of file `eHSM_If_Asr_Types_Ip.h`.

3.22.2.2 certification_key_auth_value

[ehsm_uint8_t](#)* `crypto_key_status_info::certification_key_auth_value`

Definition at line 307 of file `eHSM_If_Asr_Types_Ip.h`.

3.22.2.3 certification_key_handle

`ehsm_uint32_t crypto_key_status_info::certification_key_handle`

Definition at line 303 of file `eHSM_If_Asr_Types_Ip.h`.

3.22.2.4 key_handle

`ehsm_uint32_t crypto_key_status_info::key_handle`

Definition at line 300 of file `eHSM_If_Asr_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.23 crypto_object Struct Reference

```
#include <eHSM_Dspt_CryObj_Ip.h>
```

Public Attributes

- `crypto_object_type_e` type
- `ehsm_uint8_t * name`
- `crypto_object_state_e` state
- `ehsm_uint32_t cmd_limit`
- `ehsm_uint32_t cmd_sent_num`
- `ehsm_uint32_t queue_capacity`
- struct `dlist_head` cmd_list
- struct `dlist_head` cmd_sent

3.23.1 Detailed Description

Definition at line 30 of file `eHSM_Dspt_CryObj_Ip.h`.

3.23.2 Member Data Documentation

3.23.2.1 cmd_limit

`ehsm_uint32_t crypto_object::cmd_limit`

Definition at line 39 of file `eHSM_Dspt_CryObj_Ip.h`.

3.23.2.2 cmd_list

```
struct dlist_head crypto_object::cmd_list
```

Definition at line 46 of file eHSM_Dspt_CryObj_Ip.h.

3.23.2.3 cmd_sent

```
struct dlist_head crypto_object::cmd_sent
```

Definition at line 48 of file eHSM_Dspt_CryObj_Ip.h.

3.23.2.4 cmd_sent_num

```
ehsm_uint32_t crypto_object::cmd_sent_num
```

Definition at line 42 of file eHSM_Dspt_CryObj_Ip.h.

3.23.2.5 name

```
ehsm_uint8_t* crypto_object::name
```

dispatcher name such as "SKE" "PKE" "TRNG" "KEY MANAGER" or "HASH"

Definition at line 35 of file eHSM_Dspt_CryObj_Ip.h.

3.23.2.6 queue_capacity

```
ehsm_uint32_t crypto_object::queue_capacity
```

Definition at line 44 of file eHSM_Dspt_CryObj_Ip.h.

3.23.2.7 state

```
crypto_object_state_e crypto_object::state
```

Definition at line 37 of file eHSM_Dspt_CryObj_Ip.h.

3.23.2.8 type

```
crypto_object_type_e crypto_object::type
```

crypto object type

Definition at line 33 of file eHSM_Dspt_CryObj_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Dspt_CryObj_Ip.h](#)

3.24 Crypto_PrimitiveInfoType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- const [ehsm_uint32_t](#) resultLength
- const [Crypto_ServiceInfoType](#) service
- const [Crypto_AlgorithmInfoType](#) algorithm

3.24.1 Detailed Description

Definition at line 324 of file Asr_Standard_Types.h.

3.24.2 Member Data Documentation

3.24.2.1 algorithm

```
const Crypto\_AlgorithmInfoType Crypto_PrimitiveInfoType::algorithm
```

Definition at line 333 of file Asr_Standard_Types.h.

3.24.2.2 resultLength

```
const ehsm\_uint32\_t Crypto_PrimitiveInfoType::resultLength
```

Definition at line 331 of file Asr_Standard_Types.h.

3.24.2.3 service

```
const Crypto\_ServiceInfoType Crypto_PrimitiveInfoType::service
```

Definition at line 332 of file [Asr_Standard_Types.h](#).

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.25 crypto_she_key Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * m1
- [ehsm_uint8_t](#) * m2
- [ehsm_uint8_t](#) * m3
- [ehsm_uint8_t](#) * m4
- [ehsm_uint8_t](#) * m5
- [ehsm_bool_t](#) she_ext_flag

3.25.1 Detailed Description

Definition at line 311 of file [eHSM_If_Asr_Types_Ip.h](#).

3.25.2 Member Data Documentation

3.25.2.1 m1

```
ehsm\_uint8\_t* crypto_she_key::m1
```

Definition at line 313 of file [eHSM_If_Asr_Types_Ip.h](#).

3.25.2.2 m2

```
ehsm\_uint8\_t* crypto_she_key::m2
```

Definition at line 314 of file [eHSM_If_Asr_Types_Ip.h](#).

3.25.2.3 m3

```
ehsm_uint8_t* crypto_she_key::m3
```

Definition at line 315 of file eHSM_If_Asr_Types_lp.h.

3.25.2.4 m4

```
ehsm_uint8_t* crypto_she_key::m4
```

Definition at line 316 of file eHSM_If_Asr_Types_lp.h.

3.25.2.5 m5

```
ehsm_uint8_t* crypto_she_key::m5
```

Definition at line 317 of file eHSM_If_Asr_Types_lp.h.

3.25.2.6 she_ext_flag

```
ehsm_bool_t crypto_she_key::she_ext_flag
```

Definition at line 319 of file eHSM_If_Asr_Types_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_lp.h](#)

3.26 CryptoDriverObject Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- [ehsm_uint32_t](#) CryptoDriverObjectId
- [ehsm_uint32_t](#) CryptoQueueSize
- [ehsm_uint32_t](#) CryptoDriverObjectEcucPartitionRef
- [CryptoPrimitive](#) * [CryptoPrimitiveRef](#)
- [ehsm_uint32_t](#) CryptoPrimitiveNum

3.26.1 Detailed Description

Definition at line 474 of file Asr_Standard_Types.h.

3.26.2 Member Data Documentation

3.26.2.1 CryptoDriverObjectEcucPartitionRef

`ehsm_uint32_t` `CryptoDriverObject::CryptoDriverObjectEcucPartitionRef`

Definition at line 478 of file `Asr_Standard_Types.h`.

3.26.2.2 CryptoDriverObjectId

`ehsm_uint32_t` `CryptoDriverObject::CryptoDriverObjectId`

Definition at line 476 of file `Asr_Standard_Types.h`.

3.26.2.3 CryptoPrimitiveNum

`ehsm_uint32_t` `CryptoDriverObject::CryptoPrimitiveNum`

Definition at line 480 of file `Asr_Standard_Types.h`.

3.26.2.4 CryptoPrimitiveRef

`CryptoPrimitive*` `CryptoDriverObject::CryptoPrimitiveRef`

Definition at line 479 of file `Asr_Standard_Types.h`.

3.26.2.5 CryptoQueueSize

`ehsm_uint32_t` `CryptoDriverObject::CryptoQueueSize`

Definition at line 477 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.27 CryptoKey Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```


Public Attributes

- [ehsm_uint32_t](#) [TypeId](#)
- [CryptoKeyType](#) [KeyType](#)

3.27.1 Detailed Description

Definition at line 198 of file `eHSM_If_Asr_Types_lp.h`.

3.27.2 Member Data Documentation

3.27.2.1 KeyType

[CryptoKeyType](#) `CryptoKey::KeyType`

Definition at line 201 of file `eHSM_If_Asr_Types_lp.h`.

3.27.2.2 TypeId

[ehsm_uint32_t](#) `CryptoKey::TypeId`

Definition at line 200 of file `eHSM_If_Asr_Types_lp.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_lp.h](#)

3.28 CryptoKeyElementType Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- [ehsm_uint32_t](#) [CryptoKeyElementId](#)
- [ehsm_bool_t](#) [CryptoKeyElementAllowPartialAccess](#)
- [CryptoKeyFormatType](#) [CryptoKeyFormat](#)
- [ehsm_bool_t](#) [CryptoKeyElementPersist](#)
- [Crypto_KeyElementReadAccessType](#) [CryptoKeyElementReadAccess](#)
- [ehsm_uint32_t](#) [CryptoKeyElementMaxSize](#)
- [Crypto_KeyElementWriteAccessType](#) [CryptoKeyElementWriteAccess](#)
- [ehsm_uint32_t](#) [CryptoElementActualSize](#)
- [ehsm_uint8_t](#) * [CryptoElementArray](#)

3.28.1 Detailed Description

Definition at line 450 of file Asr_Standard_Types.h.

3.28.2 Member Data Documentation

3.28.2.1 CryptoElementActualSize

`ehsm_uint32_t` CryptoKeyElementType::CryptoElementActualSize

Definition at line 460 of file Asr_Standard_Types.h.

3.28.2.2 CryptoElementArray

`ehsm_uint8_t*` CryptoKeyElementType::CryptoElementArray

Definition at line 461 of file Asr_Standard_Types.h.

3.28.2.3 CryptoKeyElementAllowPartialAccess

`ehsm_bool_t` CryptoKeyElementType::CryptoKeyElementAllowPartialAccess

Definition at line 454 of file Asr_Standard_Types.h.

3.28.2.4 CryptoKeyElementId

`ehsm_uint32_t` CryptoKeyElementType::CryptoKeyElementId

Definition at line 453 of file Asr_Standard_Types.h.

3.28.2.5 CryptoKeyElementMaxSize

`ehsm_uint32_t` CryptoKeyElementType::CryptoKeyElementMaxSize

Definition at line 458 of file Asr_Standard_Types.h.

3.28.2.6 CryptoKeyElementPersist

`ehsm_bool_t` `CryptoKeyElementType::CryptoKeyElementPersist`

Definition at line 456 of file `Asr_Standard_Types.h`.

3.28.2.7 CryptoKeyElementReadAccess

`Crypto_KeyElementReadAccessType` `CryptoKeyElementType::CryptoKeyElementReadAccess`

Definition at line 457 of file `Asr_Standard_Types.h`.

3.28.2.8 CryptoKeyElementWriteAccess

`Crypto_KeyElementWriteAccessType` `CryptoKeyElementType::CryptoKeyElementWriteAccess`

Definition at line 459 of file `Asr_Standard_Types.h`.

3.28.2.9 CryptoKeyFormat

`CryptoKeyFormatType` `CryptoKeyElementType::CryptoKeyFormat`

Definition at line 455 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.29 CryptoKeyType Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- `ehsm_uint32_t` `keyelement_num`
- `crypto_key_element_type_info_st` * `element_info`
- `CryptoKeyElementType` * `keyelement_arr`

3.29.1 Detailed Description

Definition at line 191 of file `eHSM_If_Asr_Types_Ip.h`.

3.29.2 Member Data Documentation

3.29.2.1 element_info

[crypto_key_element_type_info_st](#)* [CryptoKeyType::element_info](#)

Definition at line 194 of file [eHSM_If_Asr_Types_Ip.h](#).

3.29.2.2 keyelement_arr

[CryptoKeyElementType](#)* [CryptoKeyType::keyelement_arr](#)

Definition at line 195 of file [eHSM_If_Asr_Types_Ip.h](#).

3.29.2.3 keyelement_num

[ehsm_uint32_t](#) [CryptoKeyType::keyelement_num](#)

Definition at line 193 of file [eHSM_If_Asr_Types_Ip.h](#).

The documentation for this struct was generated from the following file:

- [eHSM_If_Asr_Types_Ip.h](#)

3.30 CryptoPrimitive Struct Reference

```
#include <Asr_Standard_Types.h>
```

Public Attributes

- [Crypto_ServiceInfoType](#) [CryptoPrimitiveService](#)
- [Crypto_AlgorithmFamilyType](#) [CryptoPrimitiveAlgorithmFamily](#)
- [Crypto_AlgorithmModeType](#) [CryptoPrimitiveAlgorithmMode](#)
- [Crypto_AlgorithmFamilyType](#) [CryptoPrimitiveAlgorithmSecondaryFamily](#)

3.30.1 Detailed Description

Definition at line 465 of file [Asr_Standard_Types.h](#).

3.30.2 Member Data Documentation

3.30.2.1 CryptoPrimitiveAlgorithmFamily

[Crypto_AlgorithmFamilyType](#) `CryptoPrimitive::CryptoPrimitiveAlgorithmFamily`

Definition at line 468 of file `Asr_Standard_Types.h`.

3.30.2.2 CryptoPrimitiveAlgorithmMode

[Crypto_AlgorithmModeType](#) `CryptoPrimitive::CryptoPrimitiveAlgorithmMode`

Definition at line 469 of file `Asr_Standard_Types.h`.

3.30.2.3 CryptoPrimitiveAlgorithmSecondaryFamily

[Crypto_AlgorithmFamilyType](#) `CryptoPrimitive::CryptoPrimitiveAlgorithmSecondaryFamily`

Definition at line 470 of file `Asr_Standard_Types.h`.

3.30.2.4 CryptoPrimitiveService

[Crypto_ServiceInfoType](#) `CryptoPrimitive::CryptoPrimitiveService`

Definition at line 467 of file `Asr_Standard_Types.h`.

The documentation for this struct was generated from the following file:

- [Asr_Standard_Types.h](#)

3.31 dlist_head Struct Reference

```
#include <eHSM_Compt_List.h>
```

Public Attributes

- struct [dlist_head](#) * [next](#)
- struct [dlist_head](#) * [prev](#)

3.31.1 Detailed Description

Definition at line 37 of file eHSM_Compt_List.h.

3.31.2 Member Data Documentation

3.31.2.1 next

```
struct dlist_head* dlist_head::next
```

Definition at line 38 of file eHSM_Compt_List.h.

3.31.2.2 prev

```
struct dlist_head * dlist_head::prev
```

Definition at line 38 of file eHSM_Compt_List.h.

The documentation for this struct was generated from the following file:

- [eHSM_Compt_List.h](#)

3.32 ecies_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- unsigned char [curve_id](#)
- unsigned char * [msg](#)
- unsigned int [msg_bytes](#)
- unsigned char * [shared_info1](#)
- unsigned int [shared_info1_bytes](#)
- unsigned char * [shared_info2](#)
- unsigned int [shared_info2_bytes](#)
- const unsigned char * [receiver_pri_key](#)
- unsigned int [receiver_pri_key_sz](#)
- const unsigned char * [receiver_pub_key](#)
- unsigned int [receiver_pub_key_sz](#)
- unsigned char [point_form](#)
- const unsigned char * [sender_tmp_pri_key](#)
- unsigned int [sender_tmp_pri_key_sz](#)
- unsigned char [kdf_hash_alg](#)
- unsigned char [mac_hash_alg](#)
- unsigned int [mac_k_bytes](#)
- unsigned char * [cipher_part1](#)
- unsigned char * [cipher_part2_part3_without_s1_s2](#)
- unsigned char * [cipher_part2_part3_with_s1](#)
- unsigned char * [cipher_part2_part3_with_s2](#)
- unsigned char * [cipher_part2_part3_with_s1_s2](#)

3.32.1 Detailed Description

Definition at line 232 of file `utest_vecs_st.h`.

3.32.2 Member Data Documentation

3.32.2.1 cipher_part1

```
unsigned char* ecies_testvec::cipher_part1
```

Definition at line 251 of file `utest_vecs_st.h`.

3.32.2.2 cipher_part2_part3_with_s1

```
unsigned char* ecies_testvec::cipher_part2_part3_with_s1
```

Definition at line 253 of file `utest_vecs_st.h`.

3.32.2.3 cipher_part2_part3_with_s1_s2

```
unsigned char* ecies_testvec::cipher_part2_part3_with_s1_s2
```

Definition at line 255 of file `utest_vecs_st.h`.

3.32.2.4 cipher_part2_part3_with_s2

```
unsigned char* ecies_testvec::cipher_part2_part3_with_s2
```

Definition at line 254 of file `utest_vecs_st.h`.

3.32.2.5 cipher_part2_part3_without_s1_s2

```
unsigned char* ecies_testvec::cipher_part2_part3_without_s1_s2
```

Definition at line 252 of file `utest_vecs_st.h`.

3.32.2.6 curve_id

```
unsigned char ecies_testvec::curve_id
```

Definition at line 234 of file utest_vecs_st.h.

3.32.2.7 kdf_hash_alg

```
unsigned char ecies_testvec::kdf_hash_alg
```

Definition at line 248 of file utest_vecs_st.h.

3.32.2.8 mac_hash_alg

```
unsigned char ecies_testvec::mac_hash_alg
```

Definition at line 249 of file utest_vecs_st.h.

3.32.2.9 mac_k_bytes

```
unsigned int ecies_testvec::mac_k_bytes
```

Definition at line 250 of file utest_vecs_st.h.

3.32.2.10 msg

```
unsigned char* ecies_testvec::msg
```

Definition at line 235 of file utest_vecs_st.h.

3.32.2.11 msg_bytes

```
unsigned int ecies_testvec::msg_bytes
```

Definition at line 236 of file utest_vecs_st.h.

3.32.2.12 point_form

```
unsigned char ecies_testvec::point_form
```

Definition at line 245 of file `utest_vecs_st.h`.

3.32.2.13 receiver_pri_key

```
const unsigned char* ecies_testvec::receiver_pri_key
```

Definition at line 241 of file `utest_vecs_st.h`.

3.32.2.14 receiver_pri_key_sz

```
unsigned int ecies_testvec::receiver_pri_key_sz
```

Definition at line 242 of file `utest_vecs_st.h`.

3.32.2.15 receiver_pub_key

```
const unsigned char* ecies_testvec::receiver_pub_key
```

Definition at line 243 of file `utest_vecs_st.h`.

3.32.2.16 receiver_pub_key_sz

```
unsigned int ecies_testvec::receiver_pub_key_sz
```

Definition at line 244 of file `utest_vecs_st.h`.

3.32.2.17 sender_tmp_pri_key

```
const unsigned char* ecies_testvec::sender_tmp_pri_key
```

Definition at line 246 of file `utest_vecs_st.h`.

3.32.2.18 sender_tmp_pri_key_sz

```
unsigned int ecies_testvec::sender_tmp_pri_key_sz
```

Definition at line 247 of file `utest_vecs_st.h`.

3.32.2.19 shared_info1

```
unsigned char* ecies_testvec::shared_info1
```

Definition at line 237 of file `utest_vecs_st.h`.

3.32.2.20 shared_info1_bytes

```
unsigned int ecies_testvec::shared_info1_bytes
```

Definition at line 238 of file `utest_vecs_st.h`.

3.32.2.21 shared_info2

```
unsigned char* ecies_testvec::shared_info2
```

Definition at line 239 of file `utest_vecs_st.h`.

3.32.2.22 shared_info2_bytes

```
unsigned int ecies_testvec::shared_info2_bytes
```

Definition at line 240 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.33 ehsm_aead_data_ptr Struct Reference

```
#include <eHSM_Mgr_Ctx_Ip.h>
```

Public Attributes

- [ehsm_cmd_aead_ptr_st input_data](#)
- [ehsm_cmd_aead_ptr_st output_data](#)

3.33.1 Detailed Description

Definition at line 37 of file eHSM_Mgr_Ctx_Ip.h.

3.33.2 Member Data Documentation

3.33.2.1 input_data

[ehsm_cmd_aead_ptr_st](#) ehsm_aead_data_ptr::input_data

Definition at line 39 of file eHSM_Mgr_Ctx_Ip.h.

3.33.2.2 output_data

[ehsm_cmd_aead_ptr_st](#) ehsm_aead_data_ptr::output_data

Definition at line 40 of file eHSM_Mgr_Ctx_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mgr_Ctx_Ip.h](#)

3.34 ehsm_certificate_verify_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * certificate_info
- [ehsm_uint32_t](#) * verify

3.34.1 Detailed Description

Definition at line 1028 of file eHSM_Com_Struct_Ip.h.

3.34.2 Member Data Documentation

3.34.2.1 certificate_info

`ehsm_uint8_t*` ehsm_certificate_verify_st::certificate_info

Definition at line 1030 of file eHSM_Com_Struct_Ip.h.

3.34.2.2 verify

`ehsm_uint32_t*` ehsm_certificate_verify_st::verify

Definition at line 1031 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.35 ehsm_change_control_field_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) reserved [1]
- [ehsm_uint16_t](#) size
- [ehsm_uint8_t](#) value_addr [HOST_ADDRESS_SIZE]

3.35.1 Detailed Description

Definition at line 594 of file eHSM_Mailbox_Prtcl_Ip.h.

3.35.2 Member Data Documentation

3.35.2.1 reserved

`ehsm_uint8_t` ehsm_change_control_field_cmd::reserved[1]

Definition at line 596 of file eHSM_Mailbox_Prtcl_Ip.h.

3.35.2.2 size

```
ehsm_uint16_t ehsm_change_control_field_cmd::size
```

Definition at line 597 of file eHSM_Mailbox_Prtcl_Ip.h.

3.35.2.3 type

```
ehsm_uint8_t ehsm_change_control_field_cmd::type
```

Definition at line 595 of file eHSM_Mailbox_Prtcl_Ip.h.

3.35.2.4 value_addr

```
ehsm_uint8_t ehsm_change_control_field_cmd::value_addr[HOST_ADDRESS_SIZE]
```

Definition at line 598 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.36 ehsm_change_control_field_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) rev [1]
- [ehsm_uint16_t](#) size
- [ehsm_uint8_t](#) * value

3.36.1 Detailed Description

Definition at line 956 of file eHSM_Com_Struct_Ip.h.

3.36.2 Member Data Documentation

3.36.2.1 rev

```
ehsm_uint8_t ehsm_change_control_field_st::rev[1]
```

Definition at line 958 of file eHSM_Com_Struct_Ip.h.

3.36.2.2 size

```
ehsm_uint16_t ehsm_change_control_field_st::size
```

Definition at line 959 of file eHSM_Com_Struct_Ip.h.

3.36.2.3 type

```
ehsm_uint8_t ehsm_change_control_field_st::type
```

Definition at line 957 of file eHSM_Com_Struct_Ip.h.

3.36.2.4 value

```
ehsm_uint8_t* ehsm_change_control_field_st::value
```

Definition at line 960 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.37 ehsm_change_lifecycle_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) type

3.37.1 Detailed Description

Definition at line 590 of file eHSM_Mailbox_Prtcl_Ip.h.

3.37.2 Member Data Documentation

3.37.2.1 type

[ehsm_uint32_t](#) ehsm_change_lifecycle_cmd::type

Definition at line 591 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.38 ehsm_close_debug_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type

3.38.1 Detailed Description

Definition at line 138 of file eHSM_Mailbox_Prtcl_Ip.h.

3.38.2 Member Data Documentation

3.38.2.1 type

[ehsm_uint8_t](#) ehsm_close_debug_cmd::type

Definition at line 140 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.39 ehsm_cmd Struct Reference

```
#include <eHSM_Srv_Cipher_Ip.h>
```

Public Attributes

- [ehsm_cmd_cipher_st req_cipher](#)
- [ehsm_uint32_t output_size](#)

3.39.1 Detailed Description

Definition at line 24 of file eHSM_Srv_Cipher_Ip.h.

3.39.2 Member Data Documentation

3.39.2.1 output_size

[ehsm_uint32_t](#) ehsm_cmd::output_size

Definition at line 26 of file eHSM_Srv_Cipher_Ip.h.

3.39.2.2 req_cipher

[ehsm_cmd_cipher_st](#) ehsm_cmd::req_cipher

Definition at line 25 of file eHSM_Srv_Cipher_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Srv_Cipher_Ip.h](#)

3.40 ehsm_cmd_aead_ptr_st Struct Reference

GCM/CCM data structure. It's the content of [ehsm_cmd_cipher_st.input_addr/output_addr](#).

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_addr_t aad_ptr](#)
- [ehsm_addr_t data_ptr](#)
- [ehsm_addr_t tag_ptr](#)

3.40.1 Detailed Description

GCM/CCM data structure. It's the content of [ehsm_cmd_cipher_st.input_addr/output_addr](#).

Definition at line 856 of file eHSM_Mailbox_Prtcl_Ip.h.

3.40.2 Member Data Documentation

3.40.2.1 aad_ptr

[ehsm_addr_t](#) ehsm_cmd_aead_ptr_st::aad_ptr

Definition at line 858 of file eHSM_Mailbox_Prtcl_Ip.h.

3.40.2.2 data_ptr

[ehsm_addr_t](#) ehsm_cmd_aead_ptr_st::data_ptr

Definition at line 859 of file eHSM_Mailbox_Prtcl_Ip.h.

3.40.2.3 tag_ptr

[ehsm_addr_t](#) ehsm_cmd_aead_ptr_st::tag_ptr

Definition at line 860 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.41 ehsm_cmd_cipher_st Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Classes

- union [osr_cmd_hdr_u](#)

Public Attributes

- [ehsm_uint32_t](#) cmd_id
- [ehsm_uint32_t](#) rev1 [2]
- union [ehsm_cmd_cipher_st::osr_cmd_hdr_u](#) u_hdr
- [ehsm_uint32_t](#) key_handle
- [ehsm_addr_t](#) key_addr
- [ehsm_uint32_t](#) key_size
- [ehsm_addr_t](#) input_addr
- [ehsm_uint32_t](#) input_size
- [ehsm_addr_t](#) output_addr
- [ehsm_uint32_t](#) output_size
- [ehsm_addr_t](#) sec_input_addr
- [ehsm_uint32_t](#) sec_input_size
- [ehsm_addr_t](#) context_addr
- [ehsm_uint32_t](#) context_size

3.41.1 Detailed Description

Definition at line 874 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2 Member Data Documentation

3.41.2.1 cmd_id

```
ehsm_uint32_t ehsm_cmd_cipher_st::cmd_id
```

Definition at line 878 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.2 context_addr

```
ehsm_addr_t ehsm_cmd_cipher_st::context_addr
```

Definition at line 930 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.3 context_size

```
ehsm_uint32_t ehsm_cmd_cipher_st::context_size
```

Definition at line 931 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.4 input_addr

```
ehsm_addr_t ehsm_cmd_cipher_st::input_addr
```

Definition at line 909 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.5 input_size

```
ehsm_uint32_t ehsm_cmd_cipher_st::input_size
```

Definition at line 910 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.6 key_addr

`ehsm_addr_t ehsm_cmd_cipher_st::key_addr`

Definition at line 900 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.7 key_handle

`ehsm_uint32_t ehsm_cmd_cipher_st::key_handle`

Definition at line 899 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.8 key_size

`ehsm_uint32_t ehsm_cmd_cipher_st::key_size`

Definition at line 901 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.9 output_addr

`ehsm_addr_t ehsm_cmd_cipher_st::output_addr`

Definition at line 917 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.10 output_size

`ehsm_uint32_t ehsm_cmd_cipher_st::output_size`

Definition at line 918 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.11 rev1

`ehsm_uint32_t ehsm_cmd_cipher_st::rev1[2]`

Definition at line 882 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.12 sec_input_addr

[ehsm_addr_t](#) ehsm_cmd_cipher_st::sec_input_addr

Definition at line 925 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.13 sec_input_size

[ehsm_uint32_t](#) ehsm_cmd_cipher_st::sec_input_size

Definition at line 926 of file eHSM_Mailbox_Prtcl_Ip.h.

3.41.2.14 u_hdr

union [ehsm_cmd_cipher_st::osr_cmd_hdr_u](#) ehsm_cmd_cipher_st::u_hdr

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.42 ehsm_cmd_hdr_eccp_keygen_st Struct Reference

Header for sm2 and eccp key generation.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) curve_id
- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) hdr_rev1 [6]

3.42.1 Detailed Description

Header for sm2 and eccp key generation.

Definition at line 810 of file eHSM_Mailbox_Prtcl_Ip.h.

3.42.2 Member Data Documentation

3.42.2.1 curve_id

[ehsm_uint8_t](#) ehsm_cmd_hdr_eccp_keygen_st::curve_id

Definition at line 812 of file eHSM_Mailbox_Prtcl_Ip.h.

3.42.2.2 hdr_rev1

[ehsm_uint8_t](#) ehsm_cmd_hdr_eccp_keygen_st::hdr_rev1[6]

Definition at line 814 of file eHSM_Mailbox_Prtcl_Ip.h.

3.42.2.3 type

[ehsm_uint8_t](#) ehsm_cmd_hdr_eccp_keygen_st::type

Definition at line 813 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.43 ehsm_cmd_hdr_ecise_st Struct Reference

Header for ecies.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) hdr_rev1 [1]
- [ehsm_uint8_t](#) direction
- [ehsm_uint8_t](#) hdr_rev2 [1]
- [ehsm_uint8_t](#) mac_k_byte
- [ehsm_uint8_t](#) curve_id
- [ehsm_uint8_t](#) kdf_alg
- [ehsm_uint8_t](#) cipher_alg
- [ehsm_uint8_t](#) mac_alg

3.43.1 Detailed Description

Header for ecies.

Definition at line 795 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2 Member Data Documentation

3.43.2.1 cipher_alg

`ehsm_uint8_t ehsm_cmd_hdr_ecise_st::cipher_alg`

Definition at line 803 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.2 curve_id

`ehsm_uint8_t ehsm_cmd_hdr_ecise_st::curve_id`

Definition at line 801 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.3 direction

`ehsm_uint8_t ehsm_cmd_hdr_ecise_st::direction`

Definition at line 798 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.4 hdr_rev1

`ehsm_uint8_t ehsm_cmd_hdr_ecise_st::hdr_rev1[1]`

Definition at line 797 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.5 hdr_rev2

`ehsm_uint8_t ehsm_cmd_hdr_ecise_st::hdr_rev2[1]`

Definition at line 799 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.6 kdf_alg

`ehsm_uint8_t ehsm_cmd_hdr_ecise_st::kdf_alg`

Definition at line 802 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.7 mac_alg

[ehsm_uint8_t](#) ehsm_cmd_hdr_ecise_st::mac_alg

Definition at line 804 of file eHSM_Mailbox_Prtcl_Ip.h.

3.43.2.8 mac_k_byte

[ehsm_uint8_t](#) ehsm_cmd_hdr_ecise_st::mac_k_byte

Definition at line 800 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.44 ehsm_cmd_hdr_pke_st Struct Reference

Header for sm2/rsa/ecdsa.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) process_mode
- [ehsm_uint8_t](#) direction
- [ehsm_uint8_t](#) padding
- [ehsm_uint8_t](#) algorithm
- [ehsm_uint8_t](#) rsa crt_mode
- [ehsm_uint8_t](#) time_stamp
- [ehsm_uint8_t](#) hdr_rev1 [2]

3.44.1 Detailed Description

Header for sm2/rsa/ecdsa.

Definition at line 777 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2 Member Data Documentation

3.44.2.1 algorithm

[ehsm_uint8_t](#) ehsm_cmd_hdr_pke_st::algorithm

Definition at line 786 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2.2 direction

`ehsm_uint8_t ehsm_cmd_hdr_pke_st::direction`

Definition at line 782 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2.3 hdr_rev1

`ehsm_uint8_t ehsm_cmd_hdr_pke_st::hdr_rev1[2]`

Definition at line 789 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2.4 padding

`ehsm_uint8_t ehsm_cmd_hdr_pke_st::padding`

Definition at line 784 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2.5 process_mode

`ehsm_uint8_t ehsm_cmd_hdr_pke_st::process_mode`

Definition at line 780 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2.6 rsa crt_mode

`ehsm_uint8_t ehsm_cmd_hdr_pke_st::rsa crt_mode`

Definition at line 787 of file eHSM_Mailbox_Prtcl_Ip.h.

3.44.2.7 time_stamp

`ehsm_uint8_t ehsm_cmd_hdr_pke_st::time_stamp`

Definition at line 788 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.45 ehsm_cmd_hdr_rng_st Struct Reference

Header for RNG.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) `hdr_rev1` [3]
- [ehsm_uint8_t](#) `algorithm`
- [ehsm_uint8_t](#) `hdr_rev2` [4]

3.45.1 Detailed Description

Header for RNG.

Definition at line 866 of file `eHSM_Mailbox_Prtcl_Ip.h`.

3.45.2 Member Data Documentation

3.45.2.1 algorithm

```
ehsm\_uint8\_t ehsm_cmd_hdr_rng_st::algorithm
```

Definition at line 869 of file `eHSM_Mailbox_Prtcl_Ip.h`.

3.45.2.2 hdr_rev1

```
ehsm\_uint8\_t ehsm_cmd_hdr_rng_st::hdr_rev1 [3]
```

Definition at line 868 of file `eHSM_Mailbox_Prtcl_Ip.h`.

3.45.2.3 hdr_rev2

```
ehsm\_uint8\_t ehsm_cmd_hdr_rng_st::hdr_rev2 [4]
```

Definition at line 870 of file `eHSM_Mailbox_Prtcl_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.46 ehsm_cmd_hdr_rsa_keygen_st Struct Reference

Header for rsa key generation.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t is crt](#)
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t hdr_rev1](#) [2]
- [ehsm_uint8_t e_bit_size](#) [2]
- [ehsm_uint8_t n_bit_size](#) [2]

3.46.1 Detailed Description

Header for rsa key generation.

Definition at line 820 of file eHSM_Mailbox_Prtcl_Ip.h.

3.46.2 Member Data Documentation

3.46.2.1 e_bit_size

```
ehsm_uint8_t ehsm_cmd_hdr_rsa_keygen_st::e_bit_size[2]
```

Definition at line 826 of file eHSM_Mailbox_Prtcl_Ip.h.

3.46.2.2 hdr_rev1

```
ehsm_uint8_t ehsm_cmd_hdr_rsa_keygen_st::hdr_rev1[2]
```

Definition at line 825 of file eHSM_Mailbox_Prtcl_Ip.h.

3.46.2.3 is crt

```
ehsm_uint8_t ehsm_cmd_hdr_rsa_keygen_st::is crt
```

Definition at line 823 of file eHSM_Mailbox_Prtcl_Ip.h.

3.46.2.4 n_bit_size

[ehsm_uint8_t](#) ehsm_cmd_hdr_rsa_keygen_st::n_bit_size[2]

Definition at line 827 of file eHSM_Mailbox_Prtcl_Ip.h.

3.46.2.5 type

[ehsm_uint8_t](#) ehsm_cmd_hdr_rsa_keygen_st::type

Definition at line 824 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.47 ehsm_cmd_hdr_ske_st Struct Reference

Header for Ske/Aead/Mac/Hash.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) process_mode
- [ehsm_uint8_t](#) direction
- [ehsm_uint8_t](#) padding
- [ehsm_uint8_t](#) algorithm
- [ehsm_uint8_t](#) cipher_mode
- [ehsm_uint8_t](#) time_stamp
- [ehsm_uint8_t](#) tag_size
- [ehsm_uint8_t](#) key_type

3.47.1 Detailed Description

Header for Ske/Aead/Mac/Hash.

Definition at line 754 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2 Member Data Documentation

3.47.2.1 algorithm

`ehsm_uint8_t ehsm_cmd_hdr_ske_st::algorithm`

Definition at line 763 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.2 cipher_mode

`ehsm_uint8_t ehsm_cmd_hdr_ske_st::cipher_mode`

Definition at line 765 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.3 direction

`ehsm_uint8_t ehsm_cmd_hdr_ske_st::direction`

Definition at line 759 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.4 key_type

`ehsm_uint8_t ehsm_cmd_hdr_ske_st::key_type`

Definition at line 771 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.5 padding

`ehsm_uint8_t ehsm_cmd_hdr_ske_st::padding`

Definition at line 761 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.6 process_mode

`ehsm_uint8_t ehsm_cmd_hdr_ske_st::process_mode`

Definition at line 757 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.7 tag_size

[ehsm_uint8_t](#) ehsm_cmd_hdr_ske_st::tag_size

Definition at line 769 of file eHSM_Mailbox_Prtcl_Ip.h.

3.47.2.8 time_stamp

[ehsm_uint8_t](#) ehsm_cmd_hdr_ske_st::time_stamp

Definition at line 767 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.48 ehsm_cmd_hdr_sm9_st Struct Reference

Header for SM9 ciphert and signature.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) enc_type
- [ehsm_uint8_t](#) direction
- [ehsm_uint8_t](#) padding
- [ehsm_uint8_t](#) key2_size
- [ehsm_uint8_t](#) hdr_rev1 [2]
- [ehsm_uint8_t](#) hid
- [ehsm_uint8_t](#) hdr_rev2 [1]

3.48.1 Detailed Description

Header for SM9 ciphert and signature.

Definition at line 833 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2 Member Data Documentation

3.48.2.1 direction

[ehsm_uint8_t](#) ehsm_cmd_hdr_sm9_st::direction

Definition at line 836 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2.2 enc_type

`ehsm_uint8_t ehsm_cmd_hdr_sm9_st::enc_type`

Definition at line 835 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2.3 hdr_rev1

`ehsm_uint8_t ehsm_cmd_hdr_sm9_st::hdr_rev1[2]`

Definition at line 839 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2.4 hdr_rev2

`ehsm_uint8_t ehsm_cmd_hdr_sm9_st::hdr_rev2[1]`

Definition at line 841 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2.5 hid

`ehsm_uint8_t ehsm_cmd_hdr_sm9_st::hid`

Definition at line 840 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2.6 key2_size

`ehsm_uint8_t ehsm_cmd_hdr_sm9_st::key2_size`

Definition at line 838 of file eHSM_Mailbox_Prtcl_Ip.h.

3.48.2.7 padding

`ehsm_uint8_t ehsm_cmd_hdr_sm9_st::padding`

Definition at line 837 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.49 ehsm_cmd_req Struct Reference

```
#include <eHSM_Srv_CmdReq_Ip.h>
```

Public Attributes

- [ehsm_uint32_t cmd_id](#)
- [ehsm_cmd_req_state_e cmd_state](#)
- [ehsm_cmd_req_type_e req_type](#)
- [ehsm_uint32_t priority](#)
- [cmd_req_cb req_cb](#)
- [cmd_release_cb release_cb](#)
- [crypto_object_type_e object_type](#)
- [ehsm_api_type_e api_type](#)
- [struct dlist_head list](#)
- [ehsm_uint32_t error_code](#)
- [void * req_ctx](#)
- [ehsm_uint8_t rps_data \[MAX_RESPONSE_DATA_SIZE\]](#)
- [ehsm_uint32_t cmd_size](#)
- [mailbox_channel_e channel](#)
- [ehsm_uint32_t timeout](#)
- [ehsm_uint8_t cmd_data \[MAILBOX_CMD_MAX_SIZE\]](#)

3.49.1 Detailed Description

Definition at line 50 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2 Member Data Documentation

3.49.2.1 api_type

[ehsm_api_type_e](#) ehsm_cmd_req::api_type

Definition at line 66 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.2 channel

[mailbox_channel_e](#) ehsm_cmd_req::channel

Definition at line 76 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.3 cmd_data

```
ehsm_uint8_t ehsm_cmd_req::cmd_data[MAILBOX_CMD_MAX_SIZE]
```

Definition at line 80 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.4 cmd_id

```
ehsm_uint32_t ehsm_cmd_req::cmd_id
```

Definition at line 52 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.5 cmd_size

```
ehsm_uint32_t ehsm_cmd_req::cmd_size
```

Definition at line 74 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.6 cmd_state

```
ehsm_cmd_req_state_e ehsm_cmd_req::cmd_state
```

Definition at line 54 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.7 error_code

```
ehsm_uint32_t ehsm_cmd_req::error_code
```

Definition at line 69 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.8 list

```
struct dlist_head ehsm_cmd_req::list
```

Definition at line 67 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.9 object_type

`crypto_object_type_e` ehsm_cmd_req::object_type

Definition at line 64 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.10 priority

`ehsm_uint32_t` ehsm_cmd_req::priority

Definition at line 58 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.11 release_cb

`cmd_release_cb` ehsm_cmd_req::release_cb

Definition at line 62 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.12 req_cb

`cmd_req_cb` ehsm_cmd_req::req_cb

Definition at line 60 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.13 req_ctx

`void*` ehsm_cmd_req::req_ctx

Definition at line 71 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.14 req_type

`ehsm_cmd_req_type_e` ehsm_cmd_req::req_type

request type, synchronous or asynchronous

Definition at line 56 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.15 rps_data

```
ehsm_uint8_t ehsm_cmd_req::rps_data[MAX_RESPONSE_DATA_SIZE]
```

Definition at line 72 of file eHSM_Srv_CmdReq_Ip.h.

3.49.2.16 timeout

```
ehsm_uint32_t ehsm_cmd_req::timeout
```

Definition at line 78 of file eHSM_Srv_CmdReq_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Srv_CmdReq_Ip.h](#)

3.50 ehsm_cmd_req_buffer Struct Reference

Public Attributes

- [ehsm_cmd_req_st cmd_req](#) [COMMAND_REQ_QUANTITY]
- [bitmap_st](#) * [bitmap](#)

3.50.1 Detailed Description

Definition at line 102 of file eHSM_Srv_Mgr_Ip.c.

3.50.2 Member Data Documentation

3.50.2.1 bitmap

```
bitmap_st* ehsm_cmd_req_buffer::bitmap
```

Definition at line 105 of file eHSM_Srv_Mgr_Ip.c.

3.50.2.2 cmd_req

```
ehsm_cmd_req_st ehsm_cmd_req_buffer::cmd_req[COMMAND_REQ_QUANTITY]
```

Definition at line 104 of file eHSM_Srv_Mgr_Ip.c.

The documentation for this struct was generated from the following file:

- [eHSM_Srv_Mgr_Ip.c](#)

3.51 ehsm_cmd_sm9_sig_vry_output_ptr_st Struct Reference

SM9 signature and verification structure. It's the content of [ehsm_cmd_cipher_st.output_addr](#).

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t h](#) [32]
- [ehsm_uint8_t Sig](#) [65]

3.51.1 Detailed Description

SM9 signature and verification structure. It's the content of [ehsm_cmd_cipher_st.output_addr](#).

Definition at line 847 of file eHSM_Mailbox_Prtcl_Ip.h.

3.51.2 Member Data Documentation

3.51.2.1 h

```
ehsm_uint8_t ehsm_cmd_sm9_sig_vry_output_ptr_st::h[32]
```

Definition at line 849 of file eHSM_Mailbox_Prtcl_Ip.h.

3.51.2.2 Sig

```
ehsm_uint8_t ehsm_cmd_sm9_sig_vry_output_ptr_st::Sig[65]
```

Definition at line 850 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.52 ehsm_copy_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t reserved1](#) [16]
- [ehsm_uint8_t key_usage](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_usage_size](#) [4]
- [ehsm_uint8_t key_handle](#) [4]
- [ehsm_uint8_t key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_auth_size](#) [4]

3.52.1 Detailed Description

Definition at line 378 of file eHSM_Mailbox_Prtcl_Ip.h.

3.52.2 Member Data Documentation

3.52.2.1 key_auth_size

```
ehsm_uint8_t ehsm_copy_key_cmd::key_auth_size[4]
```

Definition at line 385 of file eHSM_Mailbox_Prtcl_Ip.h.

3.52.2.2 key_auth_value

```
ehsm_uint8_t ehsm_copy_key_cmd::key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 384 of file eHSM_Mailbox_Prtcl_Ip.h.

3.52.2.3 key_handle

```
ehsm_uint8_t ehsm_copy_key_cmd::key_handle[4]
```

Definition at line 383 of file eHSM_Mailbox_Prtcl_Ip.h.

3.52.2.4 key_usage

```
ehsm_uint8_t ehsm_copy_key_cmd::key_usage[HOST_ADDRESS_SIZE]
```

Definition at line 381 of file eHSM_Mailbox_Prtcl_Ip.h.

3.52.2.5 key_usage_size

```
ehsm_uint8_t ehsm_copy_key_cmd::key_usage_size[4]
```

Definition at line 382 of file eHSM_Mailbox_Prtcl_Ip.h.

3.52.2.6 reserved1

```
ehsm_uint8_t ehsm_copy_key_cmd::reserved1[16]
```

Definition at line 380 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.53 ehsm_create_dh_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t remote_key_handle](#) [4]
- [ehsm_uint8_t key_size](#) [4]
- [ehsm_uint8_t valid_until](#) [4]
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t parent_alg](#)
- [ehsm_uint8_t reserved2](#) [2]
- [ehsm_uint8_t key_usage](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_usage_size](#) [4]
- [ehsm_uint8_t local_key_handle](#) [4]
- [ehsm_uint8_t local_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t local_key_auth_size](#) [4]
- [ehsm_uint8_t remote_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t remote_key_auth_size](#) [4]
- [ehsm_uint8_t algorithm](#)
- [ehsm_uint8_t dh_mode](#)
- [ehsm_uint8_t reserved3](#) [2]
- [ehsm_uint8_t ss_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t sm2_ext_param](#) [HOST_ADDRESS_SIZE]

3.53.1 Detailed Description

Definition at line 324 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2 Member Data Documentation

3.53.2.1 algorithm

`ehsm_uint8_t ehsm_create_dh_key_cmd::algorithm`

Definition at line 338 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.2 dh_mode

`ehsm_uint8_t ehsm_create_dh_key_cmd::dh_mode`

Definition at line 339 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.3 key_size

`ehsm_uint8_t ehsm_create_dh_key_cmd::key_size[4]`

Definition at line 326 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.4 key_usage

`ehsm_uint8_t ehsm_create_dh_key_cmd::key_usage[HOST_ADDRESS_SIZE]`

Definition at line 331 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.5 key_usage_size

`ehsm_uint8_t ehsm_create_dh_key_cmd::key_usage_size[4]`

Definition at line 332 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.6 local_key_auth_size

`ehsm_uint8_t ehsm_create_dh_key_cmd::local_key_auth_size[4]`

Definition at line 335 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.7 local_key_auth_value

```
ehsm_uint8_t ehsm_create_dh_key_cmd::local_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 334 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.8 local_key_handle

```
ehsm_uint8_t ehsm_create_dh_key_cmd::local_key_handle[4]
```

Definition at line 333 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.9 parent_alg

```
ehsm_uint8_t ehsm_create_dh_key_cmd::parent_alg
```

Definition at line 329 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.10 remote_key_auth_size

```
ehsm_uint8_t ehsm_create_dh_key_cmd::remote_key_auth_size[4]
```

Definition at line 337 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.11 remote_key_auth_value

```
ehsm_uint8_t ehsm_create_dh_key_cmd::remote_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 336 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.12 remote_key_handle

```
ehsm_uint8_t ehsm_create_dh_key_cmd::remote_key_handle[4]
```

Definition at line 325 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.13 reserved2

`ehsm_uint8_t ehsm_create_dh_key_cmd::reserved2[2]`

Definition at line 330 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.14 reserved3

`ehsm_uint8_t ehsm_create_dh_key_cmd::reserved3[2]`

Definition at line 340 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.15 sm2_ext_param

`ehsm_uint8_t ehsm_create_dh_key_cmd::sm2_ext_param[HOST_ADDRESS_SIZE]`

Definition at line 343 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.16 ss_addr

`ehsm_uint8_t ehsm_create_dh_key_cmd::ss_addr[HOST_ADDRESS_SIZE]`

Definition at line 341 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.17 type

`ehsm_uint8_t ehsm_create_dh_key_cmd::type`

Definition at line 328 of file eHSM_Mailbox_Prtcl_Ip.h.

3.53.2.18 valid_until

`ehsm_uint8_t ehsm_create_dh_key_cmd::valid_until[4]`

Definition at line 327 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.54 ehsm_create_dh_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) target_algorithm_identifier
- [ehsm_uint8_t](#) dh_mode
- [ehsm_uint8_t](#) parent_alg
- [ehsm_uint32_t](#) key_size
- [ehsm_key_mem_type_e](#) type
- [ehsm_uint32_t](#) key_element_size
- [ehsm_key_flags_element_st](#) * key_element_data
- [ehsm_uint32_t](#) local_key_handle
- [ehsm_uint32_t](#) local_key_auth_size
- [ehsm_uint8_t](#) * local_key_auth_value
- [ehsm_uint32_t](#) remote_key_handle
- [ehsm_uint32_t](#) remote_key_auth_or_pub_key_size
- [ehsm_uint8_t](#) * remote_key_auth_value_or_pub_key
- [ehsm_uint32_t](#) key_handle
- [ehsm_uint8_t](#) * ss_addr
- [sm2_ext_param_st](#) sm2_ext_para

3.54.1 Detailed Description

Definition at line 336 of file eHSM_Com_Struct_Ip.h.

3.54.2 Member Data Documentation

3.54.2.1 dh_mode

[ehsm_uint8_t](#) ehsm_create_dh_key_param::dh_mode

Definition at line 342 of file eHSM_Com_Struct_Ip.h.

3.54.2.2 key_element_data

[ehsm_key_flags_element_st](#)* ehsm_create_dh_key_param::key_element_data

Definition at line 356 of file eHSM_Com_Struct_Ip.h.

3.54.2.3 key_element_size

`ehsm_uint32_t ehsm_create_dh_key_param::key_element_size`

Definition at line 354 of file eHSM_Com_Struct_Ip.h.

3.54.2.4 key_handle

`ehsm_uint32_t ehsm_create_dh_key_param::key_handle`

Definition at line 371 of file eHSM_Com_Struct_Ip.h.

3.54.2.5 key_size

`ehsm_uint32_t ehsm_create_dh_key_param::key_size`

Definition at line 346 of file eHSM_Com_Struct_Ip.h.

3.54.2.6 local_key_auth_size

`ehsm_uint32_t ehsm_create_dh_key_param::local_key_auth_size`

Definition at line 360 of file eHSM_Com_Struct_Ip.h.

3.54.2.7 local_key_auth_value

`ehsm_uint8_t* ehsm_create_dh_key_param::local_key_auth_value`

Definition at line 362 of file eHSM_Com_Struct_Ip.h.

3.54.2.8 local_key_handle

`ehsm_uint32_t ehsm_create_dh_key_param::local_key_handle`

Definition at line 358 of file eHSM_Com_Struct_Ip.h.

3.54.2.9 parent_alg

`ehsm_uint8_t ehsm_create_dh_key_param::parent_alg`

Definition at line 344 of file eHSM_Com_Struct_lp.h.

3.54.2.10 remote_key_auth_or_pub_key_size

`ehsm_uint32_t ehsm_create_dh_key_param::remote_key_auth_or_pub_key_size`

Definition at line 366 of file eHSM_Com_Struct_lp.h.

3.54.2.11 remote_key_auth_value_or_pub_key

`ehsm_uint8_t* ehsm_create_dh_key_param::remote_key_auth_value_or_pub_key`

Definition at line 368 of file eHSM_Com_Struct_lp.h.

3.54.2.12 remote_key_handle

`ehsm_uint32_t ehsm_create_dh_key_param::remote_key_handle`

Definition at line 364 of file eHSM_Com_Struct_lp.h.

3.54.2.13 sm2_ext_para

`sm2_ext_param_st ehsm_create_dh_key_param::sm2_ext_para`

Definition at line 373 of file eHSM_Com_Struct_lp.h.

3.54.2.14 ss_addr

`ehsm_uint8_t* ehsm_create_dh_key_param::ss_addr`

Definition at line 372 of file eHSM_Com_Struct_lp.h.

3.54.2.15 target_algorithm_identifier

[ehsm_uint32_t](#) ehsm_create_dh_key_param::target_algorithm_identifier

Definition at line 340 of file eHSM_Com_Struct_lp.h.

3.54.2.16 type

[ehsm_key_mem_type_e](#) ehsm_create_dh_key_param::type

Definition at line 352 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.55 ehsm_create_dh_sm2_ext_param Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) sm2_role
- [ehsm_uint8_t](#) reserved2 [3]
- [ehsm_uint8_t](#) local_tmp_key_handle [4]
- [ehsm_uint8_t](#) local_tmp_key_auth_value [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) local_tmp_key_auth_size [4]
- [ehsm_uint8_t](#) s1_s2_value [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) sa_sb_value [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) peer_temp_pubkey [HOST_ADDRESS_SIZE]

3.55.1 Detailed Description

Definition at line 313 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2 Member Data Documentation

3.55.2.1 local_tmp_key_auth_size

[ehsm_uint8_t](#) ehsm_create_dh_sm2_ext_param::local_tmp_key_auth_size[4]

Definition at line 318 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.2 local_tmp_key_auth_value

```
ehsm_uint8_t ehsm_create_dh_sm2_ext_param::local_tmp_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 317 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.3 local_tmp_key_handle

```
ehsm_uint8_t ehsm_create_dh_sm2_ext_param::local_tmp_key_handle[4]
```

Definition at line 316 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.4 peer_temp_pubkey

```
ehsm_uint8_t ehsm_create_dh_sm2_ext_param::peer_temp_pubkey[HOST_ADDRESS_SIZE]
```

Definition at line 321 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.5 reserved2

```
ehsm_uint8_t ehsm_create_dh_sm2_ext_param::reserved2[3]
```

Definition at line 315 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.6 s1_s2_value

```
ehsm_uint8_t ehsm_create_dh_sm2_ext_param::s1_s2_value[HOST_ADDRESS_SIZE]
```

Definition at line 319 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.7 sa_sb_value

```
ehsm_uint8_t ehsm_create_dh_sm2_ext_param::sa_sb_value[HOST_ADDRESS_SIZE]
```

Definition at line 320 of file eHSM_Mailbox_Prtcl_Ip.h.

3.55.2.8 sm2_role

`ehsm_uint8_t ehsm_create_dh_sm2_ext_param::sm2_role`

Definition at line 314 of file `eHSM_Mailbox_Prtcl_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.56 ehsm_create_evita_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_gen_key_param_st](#) `gen_key_param`
- [ehsm_uint16_t](#) `key_size`
- [ehsm_key_mem_type_e](#) `type`
- [ehsm_uint32_t](#) `key_element_size`
- [ehsm_key_flags_element_st](#) * `key_element_data`
- [ehsm_uint32_t](#) `key_handle`

3.56.1 Detailed Description

Definition at line 751 of file `eHSM_Com_Struct_Ip.h`.

3.56.2 Member Data Documentation

3.56.2.1 gen_key_param

`ehsm_gen_key_param_st ehsm_create_evita_key_param::gen_key_param`

Definition at line 754 of file `eHSM_Com_Struct_Ip.h`.

3.56.2.2 key_element_data

`ehsm_key_flags_element_st* ehsm_create_evita_key_param::key_element_data`

Definition at line 766 of file `eHSM_Com_Struct_Ip.h`.

3.56.2.3 key_element_size

[ehsm_uint32_t](#) ehsm_create_evita_key_param::key_element_size

Definition at line 764 of file eHSM_Com_Struct_Ip.h.

3.56.2.4 key_handle

[ehsm_uint32_t](#) ehsm_create_evita_key_param::key_handle

Definition at line 768 of file eHSM_Com_Struct_Ip.h.

3.56.2.5 key_size

[ehsm_uint16_t](#) ehsm_create_evita_key_param::key_size

Definition at line 756 of file eHSM_Com_Struct_Ip.h.

3.56.2.6 type

[ehsm_key_mem_type_e](#) ehsm_create_evita_key_param::type

Definition at line 762 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.57 ehsm_create_random_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) target_algorithm_identifier
- [ehsm_uint32_t](#) key_size
- [ehsm_key_mem_type_e](#) type
- [ehsm_uint32_t](#) key_element_size
- [ehsm_key_flags_element_st](#) * key_element_data
- [ehsm_uint8_t](#) * p
- [ehsm_uint32_t](#) p_size
- [ehsm_uint8_t](#) * q
- [ehsm_uint32_t](#) q_size
- [ehsm_uint8_t](#) * g
- [ehsm_uint32_t](#) g_size
- [ehsm_uint32_t](#) key_handle

3.57.1 Detailed Description

Definition at line 193 of file eHSM_Com_Struct_lp.h.

3.57.2 Member Data Documentation

3.57.2.1 g

`ehsm_uint8_t* ehsm_create_random_key_param::g`

Definition at line 217 of file eHSM_Com_Struct_lp.h.

3.57.2.2 g_size

`ehsm_uint32_t ehsm_create_random_key_param::g_size`

Definition at line 218 of file eHSM_Com_Struct_lp.h.

3.57.2.3 key_element_data

`ehsm_key_flags_element_st* ehsm_create_random_key_param::key_element_data`

Definition at line 211 of file eHSM_Com_Struct_lp.h.

3.57.2.4 key_element_size

`ehsm_uint32_t ehsm_create_random_key_param::key_element_size`

Definition at line 209 of file eHSM_Com_Struct_lp.h.

3.57.2.5 key_handle

`ehsm_uint32_t ehsm_create_random_key_param::key_handle`

Definition at line 220 of file eHSM_Com_Struct_lp.h.

3.57.2.6 key_size

`ehsm_uint32_t ehsm_create_random_key_param::key_size`

Definition at line 201 of file eHSM_Com_Struct_Ip.h.

3.57.2.7 p

`ehsm_uint8_t* ehsm_create_random_key_param::p`

Definition at line 213 of file eHSM_Com_Struct_Ip.h.

3.57.2.8 p_size

`ehsm_uint32_t ehsm_create_random_key_param::p_size`

Definition at line 214 of file eHSM_Com_Struct_Ip.h.

3.57.2.9 q

`ehsm_uint8_t* ehsm_create_random_key_param::q`

Definition at line 215 of file eHSM_Com_Struct_Ip.h.

3.57.2.10 q_size

`ehsm_uint32_t ehsm_create_random_key_param::q_size`

Definition at line 216 of file eHSM_Com_Struct_Ip.h.

3.57.2.11 target_algorithm_identifier

`ehsm_uint32_t ehsm_create_random_key_param::target_algorithm_identifier`

reference to target algorithm for key generation/usage based on RNG outputs, otherwise = 0 (cf. hardware interface data structures)

Definition at line 199 of file eHSM_Com_Struct_Ip.h.

3.57.2.12 type

[ehsm_key_mem_type_e](#) ehsm_create_random_key_param::type

Definition at line 207 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.58 ehsm_crypto_key Struct Reference

```
#include <eHSM_If_Asr_Types_Ip.h>
```

Public Attributes

- [ehsm_key_type_e](#) ehsm_key_type
- [ehsm_bool_t](#) valid
- [CryptoKey](#) * crypto_key

3.58.1 Detailed Description

Definition at line 205 of file eHSM_If_Asr_Types_Ip.h.

3.58.2 Member Data Documentation

3.58.2.1 crypto_key

[CryptoKey](#)* ehsm_crypto_key::crypto_key

Definition at line 212 of file eHSM_If_Asr_Types_Ip.h.

3.58.2.2 ehsm_key_type

[ehsm_key_type_e](#) ehsm_crypto_key::ehsm_key_type

Definition at line 208 of file eHSM_If_Asr_Types_Ip.h.

3.58.2.3 valid

`ehsm_bool_t ehsm_crypto_key::valid`

Definition at line 210 of file `eHSM_If_Asr_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- `eHSM_If_Asr_Types_Ip.h`

3.59 ehsm_crypto_randomgenerate_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- `ehsm_uint8_t algorithm`
- `ehsm_uint32_t random_data_addr`
- `ehsm_uint32_t request_size`

3.59.1 Detailed Description

Definition at line 476 of file `eHSM_Com_Struct_Ip.h`.

3.59.2 Member Data Documentation

3.59.2.1 algorithm

`ehsm_uint8_t ehsm_crypto_randomgenerate_param::algorithm`

Definition at line 477 of file `eHSM_Com_Struct_Ip.h`.

3.59.2.2 random_data_addr

`ehsm_uint32_t ehsm_crypto_randomgenerate_param::random_data_addr`

Definition at line 478 of file `eHSM_Com_Struct_Ip.h`.

3.59.2.3 request_size

```
ehsm_uint32_t ehsm_crypto_randomgenerate_param::request_size
```

Definition at line 479 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.60 ehsm_ctx_block_mgr Struct Reference

```
#include <eHSM_Mgr_Ctx_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) block_sz
- [ehsm_uint32_t](#) remain_data_sz
- [ehsm_uint8_t](#) block_buf [144+4]

3.60.1 Detailed Description

Definition at line 30 of file eHSM_Mgr_Ctx_Ip.h.

3.60.2 Member Data Documentation

3.60.2.1 block_buf

```
ehsm_uint8_t ehsm_ctx_block_mgr::block_buf[144+4]
```

Definition at line 34 of file eHSM_Mgr_Ctx_Ip.h.

3.60.2.2 block_sz

```
ehsm_uint32_t ehsm_ctx_block_mgr::block_sz
```

Definition at line 32 of file eHSM_Mgr_Ctx_Ip.h.

3.60.2.3 remain_data_sz

[ehsm_uint32_t](#) ehsm_ctx_block_mgr::remain_data_sz

Definition at line 33 of file eHSM_Mgr_Ctx_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mgr_Ctx_Ip.h](#)

3.61 ehsm_ctx_session_st Struct Reference

```
#include <eHSM_Mgr_Ctx_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) max_chunk_size
- [ehsm_uint32_t](#) block_sz
- [ehsm_uint32_t](#) session_id

3.61.1 Detailed Description

Definition at line 45 of file eHSM_Mgr_Ctx_Ip.h.

3.61.2 Member Data Documentation

3.61.2.1 block_sz

[ehsm_uint32_t](#) ehsm_ctx_session_st::block_sz

Definition at line 48 of file eHSM_Mgr_Ctx_Ip.h.

3.61.2.2 max_chunk_size

[ehsm_uint32_t](#) ehsm_ctx_session_st::max_chunk_size

Definition at line 47 of file eHSM_Mgr_Ctx_Ip.h.

3.61.2.3 session_id

`ehsm_uint32_t ehsm_ctx_session_st::session_id`

Definition at line 49 of file eHSM_Mgr_Ctx_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mgr_Ctx_Ip.h](#)

3.62 ehsm_debug_auth_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_challenge_type_e](#) type
- [ehsm_uint32_t](#) signature_size
- [ehsm_uint8_t](#) * signature
- [ehsm_debug_auth_alg_e](#) alg
- [ehsm_uint32_t](#) public_key_size
- [ehsm_uint8_t](#) * public_key

3.62.1 Detailed Description

Definition at line 508 of file eHSM_Com_Struct_Ip.h.

3.62.2 Member Data Documentation

3.62.2.1 alg

`ehsm_debug_auth_alg_e ehsm_debug_auth_st::alg`

Definition at line 513 of file eHSM_Com_Struct_Ip.h.

3.62.2.2 public_key

`ehsm_uint8_t* ehsm_debug_auth_st::public_key`

Definition at line 515 of file eHSM_Com_Struct_Ip.h.

3.62.2.3 public_key_size

`ehsm_uint32_t ehsm_debug_auth_st::public_key_size`

Definition at line 514 of file eHSM_Com_Struct_Ip.h.

3.62.2.4 signature

`ehsm_uint8_t* ehsm_debug_auth_st::signature`

Definition at line 512 of file eHSM_Com_Struct_Ip.h.

3.62.2.5 signature_size

`ehsm_uint32_t ehsm_debug_auth_st::signature_size`

Definition at line 511 of file eHSM_Com_Struct_Ip.h.

3.62.2.6 type

`ehsm_challenge_type_e ehsm_debug_auth_st::type`

Definition at line 510 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.63 ehsm_debug_authentication_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) algrprthm
- [ehsm_uint8_t](#) rev1 [6]
- [ehsm_uint8_t](#) rev2 [4]
- [ehsm_uint8_t](#) sign_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) sign_size [4]
- [ehsm_uint8_t](#) pub_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) pub_size [4]

3.63.1 Detailed Description

Definition at line 120 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2 Member Data Documentation

3.63.2.1 algrprithm

```
ehsm_uint8_t ehsm_debug_authentication_cmd::algrprithm
```

Definition at line 124 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.2 pub_addr

```
ehsm_uint8_t ehsm_debug_authentication_cmd::pub_addr[HOST_ADDRESS_SIZE]
```

Definition at line 132 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.3 pub_size

```
ehsm_uint8_t ehsm_debug_authentication_cmd::pub_size[4]
```

Definition at line 134 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.4 rev1

```
ehsm_uint8_t ehsm_debug_authentication_cmd::rev1[6]
```

Definition at line 125 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.5 rev2

```
ehsm_uint8_t ehsm_debug_authentication_cmd::rev2[4]
```

Definition at line 126 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.6 sign_addr

```
ehsm_uint8_t ehsm_debug_authentication_cmd::sign_addr[HOST_ADDRESS_SIZE]
```

Definition at line 128 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.7 sign_size

```
ehsm_uint8_t ehsm_debug_authentication_cmd::sign_size[4]
```

Definition at line 130 of file eHSM_Mailbox_Prtcl_Ip.h.

3.63.2.8 type

```
ehsm_uint8_t ehsm_debug_authentication_cmd::type
```

Definition at line 122 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.64 ehsm_derive_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t key_deriv_func](#)
- [ehsm_uint8_t reserved1](#) [3]
- [ehsm_uint8_t key_size](#) [4]
- [ehsm_uint8_t valid_until](#) [4]
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t derive_type](#)
- [ehsm_uint8_t reserved2](#) [2]
- [ehsm_uint8_t key_usage](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_usage_size](#) [4]
- [ehsm_uint8_t parent_key_handle](#) [4]
- [ehsm_uint8_t parent_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t parent_key_auth_size](#) [4]
- [ehsm_uint8_t salt_size](#) [4]
- [ehsm_uint8_t salt_data](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t pw_size](#) [4]
- [ehsm_uint8_t pw_data](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t itera_times](#) [4]

3.64.1 Detailed Description

Definition at line 250 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2 Member Data Documentation

3.64.2.1 derive_type

```
ehsm_uint8_t ehsm_derive_key_cmd::derive_type
```

Definition at line 256 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.2 itera_times

```
ehsm_uint8_t ehsm_derive_key_cmd::itera_times[4]
```

Definition at line 267 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.3 key_deriv_func

```
ehsm_uint8_t ehsm_derive_key_cmd::key_deriv_func
```

Definition at line 251 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.4 key_size

```
ehsm_uint8_t ehsm_derive_key_cmd::key_size[4]
```

Definition at line 253 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.5 key_usage

```
ehsm_uint8_t ehsm_derive_key_cmd::key_usage[HOST_ADDRESS_SIZE]
```

Definition at line 258 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.6 key_usage_size

```
ehsm_uint8_t ehsm_derive_key_cmd::key_usage_size[4]
```

Definition at line 259 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.7 parent_key_auth_size

```
ehsm_uint8_t ehsm_derive_key_cmd::parent_key_auth_size[4]
```

Definition at line 262 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.8 parent_key_auth_value

```
ehsm_uint8_t ehsm_derive_key_cmd::parent_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 261 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.9 parent_key_handle

```
ehsm_uint8_t ehsm_derive_key_cmd::parent_key_handle[4]
```

Definition at line 260 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.10 pw_data

```
ehsm_uint8_t ehsm_derive_key_cmd::pw_data[HOST_ADDRESS_SIZE]
```

Definition at line 266 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.11 pw_size

```
ehsm_uint8_t ehsm_derive_key_cmd::pw_size[4]
```

Definition at line 265 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.12 reserved1

```
ehsm_uint8_t ehsm_derive_key_cmd::reserved1[3]
```

Definition at line 252 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.13 reserved2

```
ehsm_uint8_t ehsm_derive_key_cmd::reserved2[2]
```

Definition at line 257 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.14 salt_data

```
ehsm_uint8_t ehsm_derive_key_cmd::salt_data[HOST_ADDRESS_SIZE]
```

Definition at line 264 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.15 salt_size

```
ehsm_uint8_t ehsm_derive_key_cmd::salt_size[4]
```

Definition at line 263 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.16 type

```
ehsm_uint8_t ehsm_derive_key_cmd::type
```

Definition at line 255 of file eHSM_Mailbox_Prtcl_Ip.h.

3.64.2.17 valid_until

```
ehsm_uint8_t ehsm_derive_key_cmd::valid_until[4]
```

Definition at line 254 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.65 ehsm_dh_param Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t p](#) [512]
- [ehsm_uint8_t q](#) [512]
- [ehsm_uint8_t g](#) [512]

3.65.1 Detailed Description

Definition at line 283 of file eHSM_If_Evita_Types_Ip.h.

3.65.2 Member Data Documentation

3.65.2.1 g

```
ehsm_uint8_t ehsm_dh_param::g[512]
```

Definition at line 287 of file eHSM_If_Evita_Types_Ip.h.

3.65.2.2 p

```
ehsm_uint8_t ehsm_dh_param::p[512]
```

Definition at line 285 of file eHSM_If_Evita_Types_Ip.h.

3.65.2.3 q

```
ehsm_uint8_t ehsm_dh_param::q[512]
```

Definition at line 286 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.66 ehsm_dh_param_size_info Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint16_t p_size](#)
- [ehsm_uint16_t q_size](#)
- [ehsm_uint16_t g_size](#)

3.66.1 Detailed Description

Definition at line 220 of file eHSM_If_Evita_Types_Ip.h.

3.66.2 Member Data Documentation

3.66.2.1 g_size

```
ehsm_uint16_t ehsm_dh_param_size_info::g_size
```

Definition at line 224 of file eHSM_If_Evita_Types_Ip.h.

3.66.2.2 p_size

```
ehsm_uint16_t ehsm_dh_param_size_info::p_size
```

Definition at line 222 of file eHSM_If_Evita_Types_Ip.h.

3.66.2.3 q_size

```
ehsm_uint16_t ehsm_dh_param_size_info::q_size
```

Definition at line 223 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.67 ehsm_dh_prikey Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t priv](#) [512]

3.67.1 Detailed Description

Definition at line 303 of file eHSM_If_Evita_Types_Ip.h.

3.67.2 Member Data Documentation

3.67.2.1 priv

```
ehsm_uint8_t ehsm_dh_prikey::priv[512]
```

Definition at line 305 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.68 ehsm_dh_pubkey Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t pub](#) [512]
- [ehsm_dh_param_st dh_param](#)

3.68.1 Detailed Description

Definition at line 290 of file eHSM_If_Evita_Types_Ip.h.

3.68.2 Member Data Documentation

3.68.2.1 dh_param

```
ehsm_dh_param_st ehsm_dh_pubkey::dh_param
```

Definition at line 293 of file eHSM_If_Evita_Types_Ip.h.

3.68.2.2 pub

```
ehsm_uint8_t ehsm_dh_pubkey::pub[512]
```

Definition at line 292 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.69 ehsm_ecc_key_size_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint16_t pub_key_size](#)
- [ehsm_uint16_t priv_key_size](#)

3.69.1 Detailed Description

Definition at line 374 of file eHSM_If_Evita_Types_Ip.h.

3.69.2 Member Data Documentation

3.69.2.1 priv_key_size

```
ehsm_uint16_t ehsm_ecc_key_size::priv_key_size
```

Definition at line 377 of file eHSM_If_Evita_Types_Ip.h.

3.69.2.2 pub_key_size

```
ehsm_uint16_t ehsm_ecc_key_size::pub_key_size
```

Definition at line 376 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.70 ehsm_ecc_pubkey Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t p](#) [132]

3.70.1 Detailed Description

Definition at line 272 of file eHSM_If_Evita_Types_Ip.h.

3.70.2 Member Data Documentation

3.70.2.1 p

```
ehsm_uint8_t ehsm_ecc_pubkey::p[132]
```

Definition at line 274 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.71 ehsm_emu_status_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t o_hsm_status](#) [2]
- [ehsm_uint32_t o_hsm_err_sensor](#)
- [ehsm_uint32_t o_hsm_err_hw](#) [2]
- [ehsm_uint32_t o_hsm_err_fw](#) [2]

3.71.1 Detailed Description

Definition at line 980 of file eHSM_Com_Struct_Ip.h.

3.71.2 Member Data Documentation

3.71.2.1 o_hsm_err_fw

[ehsm_uint32_t](#) ehsm_emu_status_st_::o_hsm_err_fw[2]

Definition at line 988 of file eHSM_Com_Struct_Ip.h.

3.71.2.2 o_hsm_err_hw

[ehsm_uint32_t](#) ehsm_emu_status_st_::o_hsm_err_hw[2]

Definition at line 986 of file eHSM_Com_Struct_Ip.h.

3.71.2.3 o_hsm_err_sensor

[ehsm_uint32_t](#) ehsm_emu_status_st_::o_hsm_err_sensor

Definition at line 984 of file eHSM_Com_Struct_Ip.h.

3.71.2.4 o_hsm_status

[ehsm_uint32_t](#) ehsm_emu_status_st_::o_hsm_status[2]

Definition at line 982 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.72 ehsm_evita_key_export Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) use_flags
- [ehsm_uint32_t](#) transport_key_handle
- [ehsm_uint32_t](#) transport_key_author_size
- [ehsm_uint8_t](#) * transport_key_author_value
- [ehsm_uint32_t](#) authenticity_key_handle
- [ehsm_uint32_t](#) authenticity_key_author_size
- [ehsm_uint8_t](#) * authenticity_key_author_value
- [ehsm_uint32_t](#) encrypted_key_size
- [ehsm_uint8_t](#) * encrypted_key
- [ehsm_uint32_t](#) key_auth_code_size
- [ehsm_uint8_t](#) * key_auth_code

3.72.1 Detailed Description

Definition at line 293 of file eHSM_Com_Struct_lp.h.

3.72.2 Member Data Documentation

3.72.2.1 authenticity_key_author_size

```
ehsm_uint32_t ehsm_evita_key_export::authenticity_key_author_size
```

Definition at line 309 of file eHSM_Com_Struct_lp.h.

3.72.2.2 authenticity_key_author_value

```
ehsm_uint8_t* ehsm_evita_key_export::authenticity_key_author_value
```

Definition at line 311 of file eHSM_Com_Struct_lp.h.

3.72.2.3 authenticity_key_handle

```
ehsm_uint32_t ehsm_evita_key_export::authenticity_key_handle
```

Definition at line 307 of file eHSM_Com_Struct_lp.h.

3.72.2.4 encrypted_key

```
ehsm_uint8_t* ehsm_evita_key_export::encrypted_key
```

Definition at line 315 of file eHSM_Com_Struct_lp.h.

3.72.2.5 encrypted_key_size

```
ehsm_uint32_t ehsm_evita_key_export::encrypted_key_size
```

Definition at line 313 of file eHSM_Com_Struct_lp.h.

3.72.2.6 key_auth_code

`ehsm_uint8_t* ehsm_evita_key_export::key_auth_code`

Definition at line 320 of file eHSM_Com_Struct_Ip.h.

3.72.2.7 key_auth_code_size

`ehsm_uint32_t ehsm_evita_key_export::key_auth_code_size`

Definition at line 318 of file eHSM_Com_Struct_Ip.h.

3.72.2.8 key_handle

`ehsm_uint32_t ehsm_evita_key_export::key_handle`

Definition at line 296 of file eHSM_Com_Struct_Ip.h.

3.72.2.9 transport_key_author_size

`ehsm_uint32_t ehsm_evita_key_export::transport_key_author_size`

Definition at line 302 of file eHSM_Com_Struct_Ip.h.

3.72.2.10 transport_key_author_value

`ehsm_uint8_t* ehsm_evita_key_export::transport_key_author_value`

Definition at line 304 of file eHSM_Com_Struct_Ip.h.

3.72.2.11 transport_key_handle

`ehsm_uint32_t ehsm_evita_key_export::transport_key_handle`

Definition at line 300 of file eHSM_Com_Struct_Ip.h.

3.72.2.12 use_flags

```
ehsm_uint32_t ehsm_evita_key_export::use_flags
```

Definition at line 298 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.73 ehsm_evita_key_import_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) transport_key_handle
- [ehsm_uint32_t](#) transport_key_author_size
- [ehsm_uint8_t](#) * transport_key_author_value
- [ehsm_uint32_t](#) authenticity_key_handle
- [ehsm_uint32_t](#) authenticity_key_author_size
- [ehsm_uint8_t](#) * authenticity_key_author_value
- [ehsm_key_mem_type_e](#) type
- [ehsm_uint32_t](#) encrypted_key_size
- [ehsm_uint8_t](#) * encrypted_key
- [ehsm_uint32_t](#) key_auth_code_size
- [ehsm_uint8_t](#) * key_auth_code
- [ehsm_uint32_t](#) key_handle

3.73.1 Detailed Description

Definition at line 263 of file eHSM_Com_Struct_Ip.h.

3.73.2 Member Data Documentation

3.73.2.1 authenticity_key_author_size

```
ehsm_uint32_t ehsm_evita_key_import_st::authenticity_key_author_size
```

Definition at line 275 of file eHSM_Com_Struct_Ip.h.

3.73.2.2 authenticity_key_author_value

`ehsm_uint8_t* ehsm_evita_key_import_st::authenticity_key_author_value`

Definition at line 277 of file eHSM_Com_Struct_lp.h.

3.73.2.3 authenticity_key_handle

`ehsm_uint32_t ehsm_evita_key_import_st::authenticity_key_handle`

Definition at line 273 of file eHSM_Com_Struct_lp.h.

3.73.2.4 encrypted_key

`ehsm_uint8_t* ehsm_evita_key_import_st::encrypted_key`

Definition at line 283 of file eHSM_Com_Struct_lp.h.

3.73.2.5 encrypted_key_size

`ehsm_uint32_t ehsm_evita_key_import_st::encrypted_key_size`

Definition at line 281 of file eHSM_Com_Struct_lp.h.

3.73.2.6 key_auth_code

`ehsm_uint8_t* ehsm_evita_key_import_st::key_auth_code`

Definition at line 288 of file eHSM_Com_Struct_lp.h.

3.73.2.7 key_auth_code_size

`ehsm_uint32_t ehsm_evita_key_import_st::key_auth_code_size`

Definition at line 286 of file eHSM_Com_Struct_lp.h.

3.73.2.8 key_handle

`ehsm_uint32_t ehsm_evita_key_import_st::key_handle`

Definition at line 290 of file eHSM_Com_Struct_Ip.h.

3.73.2.9 transport_key_author_size

`ehsm_uint32_t ehsm_evita_key_import_st::transport_key_author_size`

Definition at line 268 of file eHSM_Com_Struct_Ip.h.

3.73.2.10 transport_key_author_value

`ehsm_uint8_t* ehsm_evita_key_import_st::transport_key_author_value`

Definition at line 270 of file eHSM_Com_Struct_Ip.h.

3.73.2.11 transport_key_handle

`ehsm_uint32_t ehsm_evita_key_import_st::transport_key_handle`

Definition at line 266 of file eHSM_Com_Struct_Ip.h.

3.73.2.12 type

`ehsm_key_mem_type_e ehsm_evita_key_import_st::type`

Definition at line 279 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.74 ehsm_evita_memory_info_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) nvm_total_size
- [ehsm_uint32_t](#) nvm_free_size
- [ehsm_uint32_t](#) ram_total_size
- [ehsm_uint32_t](#) ram_free_size

3.74.1 Detailed Description

Definition at line 991 of file eHSM_Com_Struct_lp.h.

3.74.2 Member Data Documentation

3.74.2.1 nvm_free_size

[ehsm_uint32_t](#) ehsm_evita_memory_info_st::nvm_free_size

Definition at line 993 of file eHSM_Com_Struct_lp.h.

3.74.2.2 nvm_total_size

[ehsm_uint32_t](#) ehsm_evita_memory_info_st::nvm_total_size

Definition at line 992 of file eHSM_Com_Struct_lp.h.

3.74.2.3 ram_free_size

[ehsm_uint32_t](#) ehsm_evita_memory_info_st::ram_free_size

Definition at line 995 of file eHSM_Com_Struct_lp.h.

3.74.2.4 ram_total_size

[ehsm_uint32_t](#) ehsm_evita_memory_info_st::ram_total_size

Definition at line 994 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.75 ehsm_exchange_sm9_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_key_mem_type_e](#) type
- [ehsm_SM9_exchg_key_role_e](#) role
- [ehsm_uint32_t](#) key_size
- [ehsm_uint32_t](#) user_tmp_key_handle
- [ehsm_uint32_t](#) user_priv_key_handle
- [const ehsm_uint8_t *](#) peer_tmp_pub
- [const ehsm_uint8_t *](#) kgc_pub_key
- [const ehsm_uint8_t *](#) fp12g
- [const ehsm_uint8_t *](#) self_id
- [const ehsm_uint8_t *](#) peer_id
- [ehsm_uint32_t](#) self_id_size
- [ehsm_uint32_t](#) peer_id_size
- [ehsm_uint8_t *](#) s1_s2
- [ehsm_uint8_t *](#) sa_sb
- [ehsm_uint32_t](#) key_handle

3.75.1 Detailed Description

Definition at line 833 of file eHSM_Com_Struct_Ip.h.

3.75.2 Member Data Documentation

3.75.2.1 fp12g

```
const ehsm\_uint8\_t\* ehsm_exchange_sm9_key_param::fp12g
```

Definition at line 842 of file eHSM_Com_Struct_Ip.h.

3.75.2.2 key_handle

```
ehsm\_uint32\_t ehsm_exchange_sm9_key_param::key_handle
```

Definition at line 849 of file eHSM_Com_Struct_Ip.h.

3.75.2.3 key_size

`ehsm_uint32_t ehsm_exchange_sm9_key_param::key_size`

Definition at line 837 of file eHSM_Com_Struct_Ip.h.

3.75.2.4 kgc_pub_key

`const ehsm_uint8_t* ehsm_exchange_sm9_key_param::kgc_pub_key`

Definition at line 841 of file eHSM_Com_Struct_Ip.h.

3.75.2.5 peer_id

`const ehsm_uint8_t* ehsm_exchange_sm9_key_param::peer_id`

Definition at line 844 of file eHSM_Com_Struct_Ip.h.

3.75.2.6 peer_id_size

`ehsm_uint32_t ehsm_exchange_sm9_key_param::peer_id_size`

Definition at line 846 of file eHSM_Com_Struct_Ip.h.

3.75.2.7 peer_tmp_pub

`const ehsm_uint8_t* ehsm_exchange_sm9_key_param::peer_tmp_pub`

Definition at line 840 of file eHSM_Com_Struct_Ip.h.

3.75.2.8 role

`ehsm_SM9_exchg_key_role_e ehsm_exchange_sm9_key_param::role`

Definition at line 836 of file eHSM_Com_Struct_Ip.h.

3.75.2.9 s1_s2

`ehsm_uint8_t*` ehsm_exchange_sm9_key_param::s1_s2

Definition at line 847 of file eHSM_Com_Struct_lp.h.

3.75.2.10 sa_sb

`ehsm_uint8_t*` ehsm_exchange_sm9_key_param::sa_sb

Definition at line 848 of file eHSM_Com_Struct_lp.h.

3.75.2.11 self_id

`const ehsm_uint8_t*` ehsm_exchange_sm9_key_param::self_id

Definition at line 843 of file eHSM_Com_Struct_lp.h.

3.75.2.12 self_id_size

`ehsm_uint32_t` ehsm_exchange_sm9_key_param::self_id_size

Definition at line 845 of file eHSM_Com_Struct_lp.h.

3.75.2.13 type

`ehsm_key_mem_type_e` ehsm_exchange_sm9_key_param::type

Definition at line 835 of file eHSM_Com_Struct_lp.h.

3.75.2.14 user_priv_key_handle

`ehsm_uint32_t` ehsm_exchange_sm9_key_param::user_priv_key_handle

Definition at line 839 of file eHSM_Com_Struct_lp.h.

3.75.2.15 user_tmp_key_handle

```
ehsm_uint32_t ehsm_exchange_sm9_key_param::user_tmp_key_handle
```

Definition at line 838 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.76 ehsm_export_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t key_handle](#) [4]
- [ehsm_uint8_t use_flags](#) [4]
- [ehsm_uint8_t transport_key_handle](#) [4]
- [ehsm_uint8_t transport_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t transport_key_auth_size](#) [4]
- [ehsm_uint8_t authenticity_key_handle](#) [4]
- [ehsm_uint8_t authenticity_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t authenticity_key_auth_size](#) [4]
- [ehsm_uint8_t encrypted_key](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t encrypted_key_size](#) [4]
- [ehsm_uint8_t key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_auth_size](#) [4]

3.76.1 Detailed Description

Definition at line 298 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2 Member Data Documentation

3.76.2.1 authenticity_key_auth_size

```
ehsm_uint8_t ehsm_export_key_cmd::authenticity_key_auth_size[4]
```

Definition at line 306 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.2 authenticity_key_auth_value

```
ehsm_uint8_t ehsm_export_key_cmd::authenticity_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 305 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.3 authenticity_key_handle

```
ehsm_uint8_t ehsm_export_key_cmd::authenticity_key_handle[4]
```

Definition at line 304 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.4 encrypted_key

```
ehsm_uint8_t ehsm_export_key_cmd::encrypted_key[HOST_ADDRESS_SIZE]
```

Definition at line 307 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.5 encrypted_key_size

```
ehsm_uint8_t ehsm_export_key_cmd::encrypted_key_size[4]
```

Definition at line 308 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.6 key_auth_size

```
ehsm_uint8_t ehsm_export_key_cmd::key_auth_size[4]
```

Definition at line 310 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.7 key_auth_value

```
ehsm_uint8_t ehsm_export_key_cmd::key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 309 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.8 key_handle

```
ehsm_uint8_t ehsm_export_key_cmd::key_handle[4]
```

Definition at line 299 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.9 transport_key_auth_size

```
ehsm_uint8_t ehsm_export_key_cmd::transport_key_auth_size[4]
```

Definition at line 303 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.10 transport_key_auth_value

```
ehsm_uint8_t ehsm_export_key_cmd::transport_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 302 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.11 transport_key_handle

```
ehsm_uint8_t ehsm_export_key_cmd::transport_key_handle[4]
```

Definition at line 301 of file eHSM_Mailbox_Prtcl_Ip.h.

3.76.2.12 use_flags

```
ehsm_uint8_t ehsm_export_key_cmd::use_flags[4]
```

Definition at line 300 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.77 ehsm_export_pub_key_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) algo_id
- union {
 - [ehsm_uint16_t](#) rsa_e_bytes_size
 - [ehsm_uint16_t](#) dh_pubkey_bytes_size
- } size_info
- [ehsm_pubkey_data_st](#) key

3.77.1 Detailed Description

Definition at line 358 of file eHSM_If_Evita_Types_Ip.h.

3.77.2 Member Data Documentation

3.77.2.1 algo_id

[ehsm_uint32_t](#) ehsm_export_pub_key_::algo_id

Definition at line 360 of file eHSM_If_Evita_Types_Ip.h.

3.77.2.2 dh_pubkey_bytes_size

[ehsm_uint16_t](#) ehsm_export_pub_key_::dh_pubkey_bytes_size

Definition at line 363 of file eHSM_If_Evita_Types_Ip.h.

3.77.2.3 key

[ehsm_pubkey_data_st](#) ehsm_export_pub_key_::key

Definition at line 365 of file eHSM_If_Evita_Types_Ip.h.

3.77.2.4 rsa_e_bytes_size

[ehsm_uint16_t](#) ehsm_export_pub_key_::rsa_e_bytes_size

Definition at line 362 of file eHSM_If_Evita_Types_Ip.h.

3.77.2.5 size_info

```
union { ... } ehsm_export_pub_key_::size_info
```

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.78 ehsm_external_key_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- `ehsm_internal_key_st` [evita_internal_key](#)
- `ehsm_key_signatrue_st` [auth_sign_data](#)

3.78.1 Detailed Description

Definition at line 352 of file `eHSM_If_Evita_Types_Ip.h`.

3.78.2 Member Data Documentation

3.78.2.1 auth_sign_data

```
ehsm\_key\_signatrue\_st ehsm_external_key_::auth_sign_data
```

Definition at line 355 of file `eHSM_If_Evita_Types_Ip.h`.

3.78.2.2 evita_internal_key

```
ehsm_internal_key_st ehsm_external_key_::evita_internal_key
```

Definition at line 354 of file `eHSM_If_Evita_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.79 ehsm_fast_cmac_st Struct Reference

```
#include <eHSM_If_Ext_Ip.h>
```


Public Attributes

- [ehsm_uint32_t](#) key_type
- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) key_auth_size
- [ehsm_uint8_t](#) * key_auth_value
- [ehsm_uint32_t](#) size
- [ehsm_uint8_t](#) * in
- [ehsm_uint8_t](#) direction
- [ehsm_uint8_t](#) algorithm
- [ehsm_uint8_t](#) * mac

3.79.1 Detailed Description

Definition at line 43 of file eHSM_If_Ext_Ip.h.

3.79.2 Member Data Documentation

3.79.2.1 algorithm

```
ehsm\_uint8\_t ehsm_fast_cmac_st::algorithm
```

Definition at line 58 of file eHSM_If_Ext_Ip.h.

3.79.2.2 direction

```
ehsm\_uint8\_t ehsm_fast_cmac_st::direction
```

Mac direction.0: Mac generation 1: Mac verification

Definition at line 56 of file eHSM_If_Ext_Ip.h.

3.79.2.3 in

```
ehsm\_uint8\_t* ehsm_fast_cmac_st::in
```

Definition at line 54 of file eHSM_If_Ext_Ip.h.

3.79.2.4 key_auth_size

[ehsm_uint32_t](#) ehsm_fast_cmac_st::key_auth_size

Definition at line 50 of file eHSM_If_Ext_Ip.h.

3.79.2.5 key_auth_value

[ehsm_uint8_t*](#) ehsm_fast_cmac_st::key_auth_value

Definition at line 51 of file eHSM_If_Ext_Ip.h.

3.79.2.6 key_handle

[ehsm_uint32_t](#) ehsm_fast_cmac_st::key_handle

Definition at line 48 of file eHSM_If_Ext_Ip.h.

3.79.2.7 key_type

[ehsm_uint32_t](#) ehsm_fast_cmac_st::key_type

Definition at line 46 of file eHSM_If_Ext_Ip.h.

3.79.2.8 mac

[ehsm_uint8_t*](#) ehsm_fast_cmac_st::mac

Definition at line 61 of file eHSM_If_Ext_Ip.h.

3.79.2.9 size

[ehsm_uint32_t](#) ehsm_fast_cmac_st::size

Definition at line 53 of file eHSM_If_Ext_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Ext_Ip.h](#)

3.80 ehsm_fw_encrypt_key Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_fw_encrypt_key_type_e](#) key_type
- [ehsm_fw_encrypt_key_slot_e](#) key_slot
- [ehsm_uint8_t](#) * key_data
- [ehsm_uint32_t](#) key_size

3.80.1 Detailed Description

Definition at line 555 of file eHSM_Com_Struct_Ip.h.

3.80.2 Member Data Documentation

3.80.2.1 key_data

[ehsm_uint8_t](#)* ehsm_fw_encrypt_key::key_data

Definition at line 559 of file eHSM_Com_Struct_Ip.h.

3.80.2.2 key_size

[ehsm_uint32_t](#) ehsm_fw_encrypt_key::key_size

Definition at line 560 of file eHSM_Com_Struct_Ip.h.

3.80.2.3 key_slot

[ehsm_fw_encrypt_key_slot_e](#) ehsm_fw_encrypt_key::key_slot

Definition at line 558 of file eHSM_Com_Struct_Ip.h.

3.80.2.4 key_type

[ehsm_fw_encrypt_key_type_e](#) ehsm_fw_encrypt_key::key_type

Definition at line 557 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.81 ehsm_fw_encrypt_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) rev1
- [ehsm_uint8_t](#) key_type
- [ehsm_uint8_t](#) key_slot
- [ehsm_uint8_t](#) rev2 [5]
- [ehsm_uint8_t](#) rev3 [16]
- [ehsm_uint8_t](#) input_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) input_size [4]

3.81.1 Detailed Description

Definition at line 183 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2 Member Data Documentation

3.81.2.1 input_addr

[ehsm_uint8_t](#) ehsm_fw_encrypt_key_cmd::input_addr [HOST_ADDRESS_SIZE]

Definition at line 191 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2.2 input_size

[ehsm_uint8_t](#) ehsm_fw_encrypt_key_cmd::input_size [4]

Definition at line 192 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2.3 key_slot

`ehsm_uint8_t ehsm_fw_encrypt_key_cmd::key_slot`

Definition at line 187 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2.4 key_type

`ehsm_uint8_t ehsm_fw_encrypt_key_cmd::key_type`

Definition at line 186 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2.5 rev1

`ehsm_uint8_t ehsm_fw_encrypt_key_cmd::rev1`

Definition at line 185 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2.6 rev2

`ehsm_uint8_t ehsm_fw_encrypt_key_cmd::rev2[5]`

Definition at line 188 of file eHSM_Mailbox_Prtcl_Ip.h.

3.81.2.7 rev3

`ehsm_uint8_t ehsm_fw_encrypt_key_cmd::rev3[16]`

Definition at line 190 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.82 ehsm_fw_get_random_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) reserved1
- [ehsm_uint8_t](#) key_type
- [ehsm_uint8_t](#) key_slot

3.82.1 Detailed Description

Definition at line 176 of file eHSM_Mailbox_Prtcl_Ip.h.

3.82.2 Member Data Documentation

3.82.2.1 key_slot

[ehsm_uint8_t](#) ehsm_fw_get_random_key_cmd::key_slot

Definition at line 180 of file eHSM_Mailbox_Prtcl_Ip.h.

3.82.2.2 key_type

[ehsm_uint8_t](#) ehsm_fw_get_random_key_cmd::key_type

Definition at line 179 of file eHSM_Mailbox_Prtcl_Ip.h.

3.82.2.3 reserved1

[ehsm_uint8_t](#) ehsm_fw_get_random_key_cmd::reserved1

Definition at line 178 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.83 ehsm_fw_random_key Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_fw_random_key_type_e](#) key_type
- [ehsm_fw_random_key_slot_e](#) key_slot

3.83.1 Detailed Description

Definition at line 549 of file eHSM_Com_Struct_Ip.h.

3.83.2 Member Data Documentation

3.83.2.1 key_slot

[ehsm_fw_random_key_slot_e](#) ehsm_fw_random_key::key_slot

Definition at line 552 of file eHSM_Com_Struct_Ip.h.

3.83.2.2 key_type

[ehsm_fw_random_key_type_e](#) ehsm_fw_random_key::key_type

Definition at line 551 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.84 ehsm_gen_dh_key_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * p
- [ehsm_uint32_t](#) p_size
- [ehsm_uint8_t](#) * q
- [ehsm_uint32_t](#) q_size
- [ehsm_uint8_t](#) * g
- [ehsm_uint32_t](#) g_size

3.84.1 Detailed Description

Definition at line 732 of file eHSM_Com_Struct_Ip.h.

3.84.2 Member Data Documentation

3.84.2.1 g

[ehsm_uint8_t*](#) ehsm_gen_dh_key_param_st::g

Definition at line 737 of file eHSM_Com_Struct_lp.h.

3.84.2.2 g_size

[ehsm_uint32_t](#) ehsm_gen_dh_key_param_st::g_size

Definition at line 738 of file eHSM_Com_Struct_lp.h.

3.84.2.3 p

[ehsm_uint8_t*](#) ehsm_gen_dh_key_param_st::p

Definition at line 733 of file eHSM_Com_Struct_lp.h.

3.84.2.4 p_size

[ehsm_uint32_t](#) ehsm_gen_dh_key_param_st::p_size

Definition at line 734 of file eHSM_Com_Struct_lp.h.

3.84.2.5 q

[ehsm_uint8_t*](#) ehsm_gen_dh_key_param_st::q

Definition at line 735 of file eHSM_Com_Struct_lp.h.

3.84.2.6 q_size

[ehsm_uint32_t](#) ehsm_gen_dh_key_param_st::q_size

Definition at line 736 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.85 ehsm_gen_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t alg_or_crt](#)
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t reserved1](#) [2]
- [ehsm_uint8_t e_size](#) [2]
- [ehsm_uint8_t n_size](#) [2]
- [ehsm_uint8_t reserved2](#) [4]
- [ehsm_uint8_t key_usage](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_usage_size](#) [4]
- [ehsm_uint8_t reserved3](#) [4]
- [ehsm_uint8_t valid_until](#) [4]
- [ehsm_uint8_t p_size](#) [4]
- [ehsm_uint8_t p](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t q_size](#) [4]
- [ehsm_uint8_t q](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t g_size](#) [4]
- [ehsm_uint8_t g](#) [HOST_ADDRESS_SIZE]

3.85.1 Detailed Description

Definition at line 405 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2 Member Data Documentation

3.85.2.1 alg_or_crt

```
ehsm_uint8_t ehsm_gen_key_cmd::alg_or_crt
```

Definition at line 409 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.2 e_size

```
ehsm_uint8_t ehsm_gen_key_cmd::e_size[2]
```

Definition at line 413 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.3 g

`ehsm_uint8_t ehsm_gen_key_cmd::g[HOST_ADDRESS_SIZE]`

Definition at line 426 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.4 g_size

`ehsm_uint8_t ehsm_gen_key_cmd::g_size[4]`

Definition at line 425 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.5 key_usage

`ehsm_uint8_t ehsm_gen_key_cmd::key_usage[HOST_ADDRESS_SIZE]`

Definition at line 417 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.6 key_usage_size

`ehsm_uint8_t ehsm_gen_key_cmd::key_usage_size[4]`

Definition at line 418 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.7 n_size

`ehsm_uint8_t ehsm_gen_key_cmd::n_size[2]`

Definition at line 415 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.8 p

`ehsm_uint8_t ehsm_gen_key_cmd::p[HOST_ADDRESS_SIZE]`

Definition at line 422 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.9 p_size

```
ehsm_uint8_t ehsm_gen_key_cmd::p_size[4]
```

Definition at line 421 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.10 q

```
ehsm_uint8_t ehsm_gen_key_cmd::q[HOST_ADDRESS_SIZE]
```

Definition at line 424 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.11 q_size

```
ehsm_uint8_t ehsm_gen_key_cmd::q_size[4]
```

Definition at line 423 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.12 reserved1

```
ehsm_uint8_t ehsm_gen_key_cmd::reserved1[2]
```

Definition at line 411 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.13 reserved2

```
ehsm_uint8_t ehsm_gen_key_cmd::reserved2[4]
```

Definition at line 416 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.14 reserved3

```
ehsm_uint8_t ehsm_gen_key_cmd::reserved3[4]
```

Definition at line 419 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.15 type

```
ehsm_uint8_t ehsm_gen_key_cmd::type
```

Definition at line 410 of file eHSM_Mailbox_Prtcl_Ip.h.

3.85.2.16 valid_until

```
ehsm_uint8_t ehsm_gen_key_cmd::valid_until[4]
```

Definition at line 420 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.86 ehsm_gen_key_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) algo_id
- union {
 - [ehsm_uint16_t](#) rsa_e_bit_size
 - [ehsm_gen_dh_key_param_st](#) dh_key_param
- } param

3.86.1 Detailed Description

Definition at line 741 of file eHSM_Com_Struct_Ip.h.

3.86.2 Member Data Documentation

3.86.2.1 algo_id

```
ehsm_uint32_t ehsm_gen_key_param_st::algo_id
```

Definition at line 742 of file eHSM_Com_Struct_Ip.h.

3.86.2.2 dh_key_param

[ehsm_gen_dh_key_param_st](#) ehsm_gen_key_param_st::dh_key_param

Definition at line 747 of file eHSM_Com_Struct_Ip.h.

3.86.2.3 param

union { ... } ehsm_gen_key_param_st::param

3.86.2.4 rsa_e_bit_size

[ehsm_uint16_t](#) ehsm_gen_key_param_st::rsa_e_bit_size

Definition at line 745 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.87 ehsm_gen_sm9_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_gen_sm9_key_type_e](#) sm9_key_type
- [ehsm_uint8_t](#) hid
- [ehsm_uint8_t](#) rev [3]
- union {
 - [ehsm_gen_sm9_master_key_param_st](#) master_key
 - [ehsm_gen_sm9_userpriv_key_param_st](#) priv_key
- [ehsm_uint32_t](#) key_handle

3.87.1 Detailed Description

Definition at line 820 of file eHSM_Com_Struct_Ip.h.

3.87.2 Member Data Documentation

3.87.2.1 hid

`ehsm_uint8_t ehsm_gen_sm9_key_param::hid`

Definition at line 823 of file eHSM_Com_Struct_Ip.h.

3.87.2.2 key_handle

`ehsm_uint32_t ehsm_gen_sm9_key_param::key_handle`

Definition at line 829 of file eHSM_Com_Struct_Ip.h.

3.87.2.3 key_param

`union { ... } ehsm_gen_sm9_key_param::key_param`

3.87.2.4 master_key

`ehsm_gen_sm9_master_key_param_st ehsm_gen_sm9_key_param::master_key`

Definition at line 826 of file eHSM_Com_Struct_Ip.h.

3.87.2.5 priv_key

`ehsm_gen_sm9_userpriv_key_param_st ehsm_gen_sm9_key_param::priv_key`

Definition at line 827 of file eHSM_Com_Struct_Ip.h.

3.87.2.6 rev

`ehsm_uint8_t ehsm_gen_sm9_key_param::rev[3]`

Definition at line 824 of file eHSM_Com_Struct_Ip.h.

3.87.2.7 sm9_key_type

[ehsm_gen_sm9_key_type_e](#) ehsm_gen_sm9_key_param::sm9_key_type

Definition at line 822 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.88 ehsm_gen_sm9_master_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_sm9_master_key_type_e](#) master_key_type

3.88.1 Detailed Description

Definition at line 801 of file eHSM_Com_Struct_Ip.h.

3.88.2 Member Data Documentation

3.88.2.1 master_key_type

[ehsm_sm9_master_key_type_e](#) ehsm_gen_sm9_master_key_param::master_key_type

Definition at line 803 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.89 ehsm_gen_sm9_userpriv_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) rev1 [7]
- [ehsm_uint8_t](#) rev_key_handle [4]
- [ehsm_uint8_t](#) rev_key_auth [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev_key_auth_size [4]
- [ehsm_uint8_t](#) id_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) id_size [4]

3.89.1 Detailed Description

Definition at line 467 of file eHSM_Mailbox_Prtcl_lp.h.

3.89.2 Member Data Documentation

3.89.2.1 id_addr

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::id_addr[HOST_ADDRESS_SIZE]
```

Definition at line 476 of file eHSM_Mailbox_Prtcl_lp.h.

3.89.2.2 id_size

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::id_size[4]
```

Definition at line 478 of file eHSM_Mailbox_Prtcl_lp.h.

3.89.2.3 rev1

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::rev1[7]
```

Definition at line 471 of file eHSM_Mailbox_Prtcl_lp.h.

3.89.2.4 rev_key_auth

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::rev_key_auth[HOST_ADDRESS_SIZE]
```

Definition at line 473 of file eHSM_Mailbox_Prtcl_lp.h.

3.89.2.5 rev_key_auth_size

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::rev_key_auth_size[4]
```

Definition at line 474 of file eHSM_Mailbox_Prtcl_Ip.h.

3.89.2.6 rev_key_handle

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::rev_key_handle[4]
```

Definition at line 472 of file eHSM_Mailbox_Prtcl_Ip.h.

3.89.2.7 type

```
ehsm_uint8_t ehsm_gen_sm9_userpriv_key_cmd::type
```

Definition at line 470 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.90 ehsm_gen_sm9_userpriv_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_sm9_user_privkey_type_e](#) priv_key_type
- [ehsm_uint8_t](#) * user_id_value
- [ehsm_uint32_t](#) user_id_size
- [ehsm_key_mem_type_e](#) type
- [ehsm_uint8_t](#) with_pubkey
- [ehsm_uint8_t](#) kgc_pubkey [128]

3.90.1 Detailed Description

Definition at line 806 of file eHSM_Com_Struct_Ip.h.

3.90.2 Member Data Documentation

3.90.2.1 kgc_pubkey

[ehsm_uint8_t](#) ehsm_gen_sm9_userpriv_key_param::kgc_pubkey[128]

Definition at line 817 of file eHSM_Com_Struct_lp.h.

3.90.2.2 priv_key_type

[ehsm_sm9_user_privkey_type_e](#) ehsm_gen_sm9_userpriv_key_param::priv_key_type

Definition at line 808 of file eHSM_Com_Struct_lp.h.

3.90.2.3 type

[ehsm_key_mem_type_e](#) ehsm_gen_sm9_userpriv_key_param::type

Definition at line 814 of file eHSM_Com_Struct_lp.h.

3.90.2.4 user_id_size

[ehsm_uint32_t](#) ehsm_gen_sm9_userpriv_key_param::user_id_size

Definition at line 812 of file eHSM_Com_Struct_lp.h.

3.90.2.5 user_id_value

[ehsm_uint8_t*](#) ehsm_gen_sm9_userpriv_key_param::user_id_value

Definition at line 810 of file eHSM_Com_Struct_lp.h.

3.90.2.6 with_pubkey

[ehsm_uint8_t](#) ehsm_gen_sm9_userpriv_key_param::with_pubkey

Definition at line 815 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.91 ehsm_get_challenge_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) reserved1 [35]
- [ehsm_uint8_t](#) output_addr [HOST_ADDRESS_SIZE]

3.91.1 Detailed Description

Definition at line 111 of file eHSM_Mailbox_Prtcl_Ip.h.

3.91.2 Member Data Documentation

3.91.2.1 output_addr

```
ehsm\_uint8\_t ehsm_get_challenge_cmd::output_addr[HOST_ADDRESS_SIZE]
```

Definition at line 117 of file eHSM_Mailbox_Prtcl_Ip.h.

3.91.2.2 reserved1

```
ehsm\_uint8\_t ehsm_get_challenge_cmd::reserved1[35]
```

Definition at line 115 of file eHSM_Mailbox_Prtcl_Ip.h.

3.91.2.3 type

```
ehsm\_uint8\_t ehsm_get_challenge_cmd::type
```

Definition at line 113 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.92 ehsm_get_challenge_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_challenge_type_e](#) type
- [ehsm_uint32_t](#) size
- [ehsm_uint8_t](#) * buf

3.92.1 Detailed Description

Definition at line 501 of file eHSM_Com_Struct_Ip.h.

3.92.2 Member Data Documentation

3.92.2.1 buf

```
ehsm_uint8_t* ehsm_get_challenge_st::buf
```

Definition at line 505 of file eHSM_Com_Struct_Ip.h.

3.92.2.2 size

```
ehsm_uint32_t ehsm_get_challenge_st::size
```

Definition at line 504 of file eHSM_Com_Struct_Ip.h.

3.92.2.3 type

```
ehsm_challenge_type_e ehsm_get_challenge_st::type
```

Definition at line 503 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.93 ehsm_get_emu_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) rev1 [8]
- [ehsm_uint8_t](#) rev_key_handle [4]
- [ehsm_uint8_t](#) rev_key_auth_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) rev_key_auth_size
- [ehsm_uint8_t](#) rev2 [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) rev3
- [ehsm_uint8_t](#) emu_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) emu_size

3.93.1 Detailed Description

Definition at line 620 of file eHSM_Mailbox_Prtcl_lp.h.

3.93.2 Member Data Documentation

3.93.2.1 emu_addr

```
ehsm\_uint8\_t ehsm_get_emu_cmd::emu_addr[HOST_ADDRESS_SIZE]
```

Definition at line 627 of file eHSM_Mailbox_Prtcl_lp.h.

3.93.2.2 emu_size

```
ehsm\_uint32\_t ehsm_get_emu_cmd::emu_size
```

Definition at line 628 of file eHSM_Mailbox_Prtcl_lp.h.

3.93.2.3 rev1

```
ehsm\_uint8\_t ehsm_get_emu_cmd::rev1[8]
```

Definition at line 621 of file eHSM_Mailbox_Prtcl_lp.h.

3.93.2.4 rev2

```
ehsm\_uint8\_t ehsm_get_emu_cmd::rev2[HOST_ADDRESS_SIZE]
```

Definition at line 625 of file eHSM_Mailbox_Prtcl_lp.h.

3.93.2.5 rev3

`ehsm_uint32_t ehsm_get_emu_cmd::rev3`

Definition at line 626 of file eHSM_Mailbox_Prtcl_Ip.h.

3.93.2.6 rev_key_auth_addr

`ehsm_uint8_t ehsm_get_emu_cmd::rev_key_auth_addr[HOST_ADDRESS_SIZE]`

Definition at line 623 of file eHSM_Mailbox_Prtcl_Ip.h.

3.93.2.7 rev_key_auth_size

`ehsm_uint32_t ehsm_get_emu_cmd::rev_key_auth_size`

Definition at line 624 of file eHSM_Mailbox_Prtcl_Ip.h.

3.93.2.8 rev_key_handle

`ehsm_uint8_t ehsm_get_emu_cmd::rev_key_handle[4]`

Definition at line 622 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.94 ehsm_get_emu_status_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- `ehsm_uint8_t * emu_addr`
- `ehsm_uint32_t * emu_size`

3.94.1 Detailed Description

Definition at line 972 of file eHSM_Com_Struct_Ip.h.

3.94.2 Member Data Documentation

3.94.2.1 emu_addr

[ehsm_uint8_t*](#) ehsm_get_emu_status_param_st::emu_addr

Definition at line 974 of file eHSM_Com_Struct_Ip.h.

3.94.2.2 emu_size

[ehsm_uint32_t*](#) ehsm_get_emu_status_param_st::emu_size

Definition at line 976 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.95 ehsm_get_pub_from_priv_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) reserved1 [8]
- [ehsm_uint8_t](#) key_handle [4]
- [ehsm_uint8_t](#) key_auth_value [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) key_auth_size [4]
- [ehsm_uint32_t](#) key_alg_id
- [ehsm_uint8_t](#) reserved2 [4]
- [ehsm_uint8_t](#) public_key_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) public_key_buffer_size [4]

3.95.1 Detailed Description

Definition at line 346 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2 Member Data Documentation

3.95.2.1 key_alg_id

`ehsm_uint32_t ehsm_get_pub_from_priv_cmd::key_alg_id`

Definition at line 352 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.2 key_auth_size

`ehsm_uint8_t ehsm_get_pub_from_priv_cmd::key_auth_size[4]`

Definition at line 351 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.3 key_auth_value

`ehsm_uint8_t ehsm_get_pub_from_priv_cmd::key_auth_value[HOST_ADDRESS_SIZE]`

Definition at line 350 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.4 key_handle

`ehsm_uint8_t ehsm_get_pub_from_priv_cmd::key_handle[4]`

Definition at line 349 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.5 public_key_addr

`ehsm_uint8_t ehsm_get_pub_from_priv_cmd::public_key_addr[HOST_ADDRESS_SIZE]`

Definition at line 354 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.6 public_key_buffer_size

`ehsm_uint8_t ehsm_get_pub_from_priv_cmd::public_key_buffer_size[4]`

Definition at line 355 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.7 reserved1

```
ehsm_uint8_t ehsm_get_pub_from_priv_cmd::reserved1[8]
```

Definition at line 348 of file eHSM_Mailbox_Prtcl_Ip.h.

3.95.2.8 reserved2

```
ehsm_uint8_t ehsm_get_pub_from_priv_cmd::reserved2[4]
```

Definition at line 353 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.96 ehsm_get_pub_from_priv_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) key_auth_size
- [ehsm_uint8_t](#) * key_auth_value
- [ehsm_uint32_t](#) key_alg_id
- [ehsm_uint8_t](#) * public_key_addr
- [ehsm_uint32_t](#) public_key_buffer_size
- [ehsm_uint32_t](#) public_key_size

3.96.1 Detailed Description

Definition at line 376 of file eHSM_Com_Struct_Ip.h.

3.96.2 Member Data Documentation

3.96.2.1 key_alg_id

```
ehsm_uint32_t ehsm_get_pub_from_priv_param::key_alg_id
```

Definition at line 385 of file eHSM_Com_Struct_Ip.h.

3.96.2.2 key_auth_size

[ehsm_uint32_t](#) ehsm_get_pub_from_priv_param::key_auth_size

Definition at line 381 of file eHSM_Com_Struct_lp.h.

3.96.2.3 key_auth_value

[ehsm_uint8_t*](#) ehsm_get_pub_from_priv_param::key_auth_value

Definition at line 383 of file eHSM_Com_Struct_lp.h.

3.96.2.4 key_handle

[ehsm_uint32_t](#) ehsm_get_pub_from_priv_param::key_handle

Definition at line 379 of file eHSM_Com_Struct_lp.h.

3.96.2.5 public_key_addr

[ehsm_uint8_t*](#) ehsm_get_pub_from_priv_param::public_key_addr

Definition at line 387 of file eHSM_Com_Struct_lp.h.

3.96.2.6 public_key_buffer_size

[ehsm_uint32_t](#) ehsm_get_pub_from_priv_param::public_key_buffer_size

Definition at line 389 of file eHSM_Com_Struct_lp.h.

3.96.2.7 public_key_size

[ehsm_uint32_t](#) ehsm_get_pub_from_priv_param::public_key_size

Definition at line 391 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.97 ehsm_get_she_id_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) reserved [12]
- [ehsm_uint8_t](#) challenge_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) challenge_size
- [ehsm_uint8_t](#) status_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) status_size
- [ehsm_uint8_t](#) signatrue_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) signatrue_size

3.97.1 Detailed Description

Definition at line 610 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2 Member Data Documentation

3.97.2.1 challenge_addr

```
ehsm\_uint8\_t ehsm_get_she_id_cmd::challenge_addr[HOST_ADDRESS_SIZE]
```

Definition at line 612 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2.2 challenge_size

```
ehsm\_uint32\_t ehsm_get_she_id_cmd::challenge_size
```

Definition at line 613 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2.3 reserved

```
ehsm\_uint8\_t ehsm_get_she_id_cmd::reserved[12]
```

Definition at line 611 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2.4 signatrue_addr

```
ehsm_uint8_t ehsm_get_she_id_cmd::signatrue_addr[HOST_ADDRESS_SIZE]
```

Definition at line 616 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2.5 signatrue_size

```
ehsm_uint32_t ehsm_get_she_id_cmd::signatrue_size
```

Definition at line 617 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2.6 status_addr

```
ehsm_uint8_t ehsm_get_she_id_cmd::status_addr[HOST_ADDRESS_SIZE]
```

Definition at line 614 of file eHSM_Mailbox_Prtcl_Ip.h.

3.97.2.7 status_size

```
ehsm_uint32_t ehsm_get_she_id_cmd::status_size
```

Definition at line 615 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.98 ehsm_image Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_image_process_mode_e](#) process_mode
- [ehsm_uint8_t](#) * image
- [ehsm_uint32_t](#) image_size
- [ehsm_uint8_t](#) * storage
- [ehsm_uint32_t](#) storage_size
- [ehsm_uint8_t](#) * ctx
- [ehsm_uint32_t](#) ctx_size

3.98.1 Detailed Description

Definition at line 570 of file eHSM_Com_Struct_lp.h.

3.98.2 Member Data Documentation

3.98.2.1 ctx

```
ehsm_uint8_t* ehsm_image::ctx
```

Definition at line 577 of file eHSM_Com_Struct_lp.h.

3.98.2.2 ctx_size

```
ehsm_uint32_t ehsm_image::ctx_size
```

Definition at line 578 of file eHSM_Com_Struct_lp.h.

3.98.2.3 image

```
ehsm_uint8_t* ehsm_image::image
```

Definition at line 573 of file eHSM_Com_Struct_lp.h.

3.98.2.4 image_size

```
ehsm_uint32_t ehsm_image::image_size
```

Definition at line 574 of file eHSM_Com_Struct_lp.h.

3.98.2.5 process_mode

```
ehsm_image_process_mode_e ehsm_image::process_mode
```

Definition at line 572 of file eHSM_Com_Struct_lp.h.

3.98.2.6 storage

```
ehsm_uint8_t* ehsm_image::storage
```

Definition at line 575 of file eHSM_Com_Struct_Ip.h.

3.98.2.7 storage_size

```
ehsm_uint32_t ehsm_image::storage_size
```

Definition at line 576 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.99 ehsm_image_upgrade_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t process_mode](#)
- [ehsm_uint8_t rev1](#) [7]
- [ehsm_uint8_t rev_key_handle](#) [4]
- [ehsm_uint8_t rev_key_auth](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev_key_auth_size](#) [4]
- [ehsm_uint8_t image_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t image_size](#) [4]
- [ehsm_uint8_t storage_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev2](#) [4]
- [ehsm_uint8_t rev3](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev4](#) [4]
- [ehsm_uint8_t ctx_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t ctx_size](#) [4]

3.99.1 Detailed Description

Definition at line 195 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2 Member Data Documentation

3.99.2.1 ctx_addr

```
ehsm_uint8_t ehsm_image_upgrade_cmd::ctx_addr[HOST_ADDRESS_SIZE]
```

Definition at line 208 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.2 ctx_size

```
ehsm_uint8_t ehsm_image_upgrade_cmd::ctx_size[4]
```

Definition at line 209 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.3 image_addr

```
ehsm_uint8_t ehsm_image_upgrade_cmd::image_addr[HOST_ADDRESS_SIZE]
```

Definition at line 202 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.4 image_size

```
ehsm_uint8_t ehsm_image_upgrade_cmd::image_size[4]
```

Definition at line 203 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.5 process_mode

```
ehsm_uint8_t ehsm_image_upgrade_cmd::process_mode
```

Definition at line 197 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.6 rev1

```
ehsm_uint8_t ehsm_image_upgrade_cmd::rev1[7]
```

Definition at line 198 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.7 rev2

`ehsm_uint8_t ehsm_image_upgrade_cmd::rev2[4]`

Definition at line 205 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.8 rev3

`ehsm_uint8_t ehsm_image_upgrade_cmd::rev3[HOST_ADDRESS_SIZE]`

Definition at line 206 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.9 rev4

`ehsm_uint8_t ehsm_image_upgrade_cmd::rev4[4]`

Definition at line 207 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.10 rev_key_auth

`ehsm_uint8_t ehsm_image_upgrade_cmd::rev_key_auth[HOST_ADDRESS_SIZE]`

Definition at line 200 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.11 rev_key_auth_size

`ehsm_uint8_t ehsm_image_upgrade_cmd::rev_key_auth_size[4]`

Definition at line 201 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.12 rev_key_handle

`ehsm_uint8_t ehsm_image_upgrade_cmd::rev_key_handle[4]`

Definition at line 199 of file eHSM_Mailbox_Prtcl_Ip.h.

3.99.2.13 storage_addr

```
ehsm_uint8_t ehsm_image_upgrade_cmd::storage_addr[HOST_ADDRESS_SIZE]
```

Definition at line 204 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.100 ehsm_image_verfiy_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) process_mode
- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) rev1 [6]
- [ehsm_uint8_t](#) rev_key_handle [4]
- [ehsm_uint8_t](#) rev_key_auth [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev_key_auth_size [4]
- [ehsm_uint8_t](#) image_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) image_size [4]
- [ehsm_uint8_t](#) rev2 [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev3 [4]
- [ehsm_uint8_t](#) rev4 [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev5 [4]
- [ehsm_uint8_t](#) ctx_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) ctx_size [4]

3.100.1 Detailed Description

Definition at line 212 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2 Member Data Documentation

3.100.2.1 ctx_addr

```
ehsm_uint8_t ehsm_image_verfiy_cmd::ctx_addr[HOST_ADDRESS_SIZE]
```

Definition at line 226 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.2 ctx_size

```
ehsm_uint8_t ehsm_image_verfiy_cmd::ctx_size[4]
```

Definition at line 227 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.3 image_addr

```
ehsm_uint8_t ehsm_image_verfiy_cmd::image_addr[HOST_ADDRESS_SIZE]
```

Definition at line 220 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.4 image_size

```
ehsm_uint8_t ehsm_image_verfiy_cmd::image_size[4]
```

Definition at line 221 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.5 process_mode

```
ehsm_uint8_t ehsm_image_verfiy_cmd::process_mode
```

Definition at line 214 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.6 rev1

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev1[6]
```

Definition at line 216 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.7 rev2

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev2[HOST_ADDRESS_SIZE]
```

Definition at line 222 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.8 rev3

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev3[4]
```

Definition at line 223 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.9 rev4

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev4[HOST_ADDRESS_SIZE]
```

Definition at line 224 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.10 rev5

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev5[4]
```

Definition at line 225 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.11 rev_key_auth

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev_key_auth[HOST_ADDRESS_SIZE]
```

Definition at line 218 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.12 rev_key_auth_size

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev_key_auth_size[4]
```

Definition at line 219 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.13 rev_key_handle

```
ehsm_uint8_t ehsm_image_verfiy_cmd::rev_key_handle[4]
```

Definition at line 217 of file eHSM_Mailbox_Prtcl_Ip.h.

3.100.2.14 type

`ehsm_uint8_t ehsm_image_verfiy_cmd::type`

Definition at line 215 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.101 ehsm_image_verify_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_image_process_mode_e process_mode](#)
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t * image](#)
- [ehsm_uint32_t image_size](#)
- [ehsm_uint8_t * storage](#)
- [ehsm_uint32_t storage_size](#)
- [ehsm_uint8_t * ctx](#)
- [ehsm_uint32_t ctx_size](#)

3.101.1 Detailed Description

Definition at line 581 of file eHSM_Com_Struct_Ip.h.

3.101.2 Member Data Documentation

3.101.2.1 ctx

`ehsm_uint8_t* ehsm_image_verify_st::ctx`

Definition at line 589 of file eHSM_Com_Struct_Ip.h.

3.101.2.2 ctx_size

`ehsm_uint32_t ehsm_image_verify_st::ctx_size`

Definition at line 590 of file eHSM_Com_Struct_Ip.h.

3.101.2.3 image

`ehsm_uint8_t* ehsm_image_verify_st::image`

Definition at line 585 of file eHSM_Com_Struct_lp.h.

3.101.2.4 image_size

`ehsm_uint32_t ehsm_image_verify_st::image_size`

Definition at line 586 of file eHSM_Com_Struct_lp.h.

3.101.2.5 process_mode

`ehsm_image_process_mode_e ehsm_image_verify_st::process_mode`

Definition at line 583 of file eHSM_Com_Struct_lp.h.

3.101.2.6 storage

`ehsm_uint8_t* ehsm_image_verify_st::storage`

Definition at line 587 of file eHSM_Com_Struct_lp.h.

3.101.2.7 storage_size

`ehsm_uint32_t ehsm_image_verify_st::storage_size`

Definition at line 588 of file eHSM_Com_Struct_lp.h.

3.101.2.8 type

`ehsm_uint8_t ehsm_image_verify_st::type`

Definition at line 584 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.102 ehsm_import_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t key_type](#)
- [ehsm_uint8_t rev1](#) [7]
- [ehsm_uint8_t transport_key_handle](#) [4]
- [ehsm_uint8_t transport_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t transport_key_auth_size](#) [4]
- [ehsm_uint8_t authenticity_key_handle](#) [4]
- [ehsm_uint8_t authenticity_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t authenticity_key_auth_size](#) [4]
- [ehsm_uint8_t encrypted_key](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t encrypted_key_size](#) [4]
- [ehsm_uint8_t key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_auth_size](#) [4]

3.102.1 Detailed Description

Definition at line 283 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2 Member Data Documentation

3.102.2.1 authenticity_key_auth_size

```
ehsm_uint8_t ehsm_import_key_cmd::authenticity_key_auth_size[4]
```

Definition at line 291 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.2 authenticity_key_auth_value

```
ehsm_uint8_t ehsm_import_key_cmd::authenticity_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 290 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.3 authenticity_key_handle

```
ehsm_uint8_t ehsm_import_key_cmd::authenticity_key_handle[4]
```

Definition at line 289 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.4 encrypted_key

```
ehsm_uint8_t ehsm_import_key_cmd::encrypted_key[HOST_ADDRESS_SIZE]
```

Definition at line 292 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.5 encrypted_key_size

```
ehsm_uint8_t ehsm_import_key_cmd::encrypted_key_size[4]
```

Definition at line 293 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.6 key_auth_size

```
ehsm_uint8_t ehsm_import_key_cmd::key_auth_size[4]
```

Definition at line 295 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.7 key_auth_value

```
ehsm_uint8_t ehsm_import_key_cmd::key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 294 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.8 key_type

```
ehsm_uint8_t ehsm_import_key_cmd::key_type
```

Definition at line 284 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.9 rev1

```
ehsm_uint8_t ehsm_import_key_cmd::rev1[7]
```

Definition at line 285 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.10 transport_key_auth_size

```
ehsm_uint8_t ehsm_import_key_cmd::transport_key_auth_size[4]
```

Definition at line 288 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.11 transport_key_auth_value

```
ehsm_uint8_t ehsm_import_key_cmd::transport_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 287 of file eHSM_Mailbox_Prtcl_Ip.h.

3.102.2.12 transport_key_handle

```
ehsm_uint8_t ehsm_import_key_cmd::transport_key_handle[4]
```

Definition at line 286 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.103 ehsm_internal_key_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_key_attr_data_st](#) attr
- [ehsm_key_usages_st](#) key_usage
- [ehsm_pubkey_data_st](#) pubkey
- [ehsm_uint32_t](#) prikey_enc_size
- [ehsm_prikey_data_st](#) prikey
- [ehsm_key_signatrue_st](#) key_signatrue

3.103.1 Detailed Description

Definition at line 342 of file eHSM_If_Evita_Types_Ip.h.

3.103.2 Member Data Documentation

3.103.2.1 attr

[ehsm_key_attr_data_st](#) ehsm_internal_key_::attr

Definition at line 344 of file eHSM_If_Evita_Types_Ip.h.

3.103.2.2 key_signatrue

[ehsm_key_signatrue_st](#) ehsm_internal_key_::key_signatrue

Definition at line 349 of file eHSM_If_Evita_Types_Ip.h.

3.103.2.3 key_usage

[ehsm_key_usages_st](#) ehsm_internal_key_::key_usage

Definition at line 345 of file eHSM_If_Evita_Types_Ip.h.

3.103.2.4 prikey

[ehsm_prikey_data_st](#) ehsm_internal_key_::prikey

Definition at line 348 of file eHSM_If_Evita_Types_Ip.h.

3.103.2.5 prikey_enc_size

[ehsm_uint32_t](#) ehsm_internal_key_::prikey_enc_size

Definition at line 347 of file eHSM_If_Evita_Types_Ip.h.

3.103.2.6 pubkey

[ehsm_pubkey_data_st](#) ehsm_internal_key_::pubkey

Definition at line 346 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.104 ehsm_key_attr_data_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_identifier
- [ehsm_uint32_t](#) algo_id
- [ehsm_uint32_t](#) valid_util
- [key_info_st](#) key_info
- [ehsm_uint16_t](#) key_usage_size
- [ehsm_uint16_t](#) key_signatrue_off

3.104.1 Detailed Description

Definition at line 253 of file eHSM_If_Evita_Types_Ip.h.

3.104.2 Member Data Documentation

3.104.2.1 algo_id

```
ehsm\_uint32\_t ehsm_key_attr_data_::algo_id
```

Definition at line 256 of file eHSM_If_Evita_Types_Ip.h.

3.104.2.2 key_identifier

```
ehsm\_uint32\_t ehsm_key_attr_data_::key_identifier
```

Definition at line 255 of file eHSM_If_Evita_Types_Ip.h.

3.104.2.3 key_info

```
key\_info\_st ehsm_key_attr_data_::key_info
```

Definition at line 258 of file eHSM_If_Evita_Types_Ip.h.

3.104.2.4 key_signatrue_off

[ehsm_uint16_t](#) ehsm_key_attr_data_::key_signatrue_off

Definition at line 260 of file eHSM_If_Evita_Types_Ip.h.

3.104.2.5 key_usage_size

[ehsm_uint16_t](#) ehsm_key_attr_data_::key_usage_size

Definition at line 259 of file eHSM_If_Evita_Types_Ip.h.

3.104.2.6 valid_util

[ehsm_uint32_t](#) ehsm_key_attr_data_::valid_util

Definition at line 257 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.105 ehsm_key_copy_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) parent_key_handle
- [ehsm_uint8_t](#) * key_auth_value
- [ehsm_uint32_t](#) key_auth_size
- [ehsm_uint32_t](#) target_key_handle
- [ehsm_uint32_t](#) key_element_size
- [ehsm_key_flags_element_st](#) * key_element_data

3.105.1 Detailed Description

Definition at line 421 of file eHSM_Com_Struct_Ip.h.

3.105.2 Member Data Documentation

3.105.2.1 key_auth_size

[ehsm_uint32_t](#) ehsm_key_copy_param::key_auth_size

Definition at line 425 of file eHSM_Com_Struct_Ip.h.

3.105.2.2 key_auth_value

[ehsm_uint8_t*](#) ehsm_key_copy_param::key_auth_value

Definition at line 424 of file eHSM_Com_Struct_Ip.h.

3.105.2.3 key_element_data

[ehsm_key_flags_element_st*](#) ehsm_key_copy_param::key_element_data

Definition at line 430 of file eHSM_Com_Struct_Ip.h.

3.105.2.4 key_element_size

[ehsm_uint32_t](#) ehsm_key_copy_param::key_element_size

Definition at line 428 of file eHSM_Com_Struct_Ip.h.

3.105.2.5 parent_key_handle

[ehsm_uint32_t](#) ehsm_key_copy_param::parent_key_handle

Definition at line 423 of file eHSM_Com_Struct_Ip.h.

3.105.2.6 target_key_handle

[ehsm_uint32_t](#) ehsm_key_copy_param::target_key_handle

Definition at line 426 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.106 ehsm_key_derived_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) key_deriv_func
- [ehsm_uint32_t](#) key_size
- [ehsm_key_mem_type_e](#) type
- [ehsm_uint32_t](#) key_element_size
- [ehsm_key_flags_element_st](#) * key_element_data
- [crypto_key_derive_type_e](#) derive_type
- [ehsm_uint8_t](#) * passwd
- [ehsm_uint32_t](#) passwd_size
- [ehsm_uint32_t](#) itera_times
- [ehsm_uint32_t](#) parent_key_handle
- [ehsm_uint32_t](#) parent_key_author_size
- [ehsm_uint8_t](#) * parent_key_author_value
- [ehsm_uint32_t](#) salt_size
- [ehsm_uint8_t](#) * salt_data
- [ehsm_uint32_t](#) key_handle

3.106.1 Detailed Description

Definition at line 228 of file eHSM_Com_Struct_Ip.h.

3.106.2 Member Data Documentation

3.106.2.1 derive_type

[crypto_key_derive_type_e](#) ehsm_key_derived_param::derive_type

Definition at line 244 of file eHSM_Com_Struct_Ip.h.

3.106.2.2 itera_times

[ehsm_uint32_t](#) ehsm_key_derived_param::itera_times

Definition at line 248 of file eHSM_Com_Struct_Ip.h.

3.106.2.3 key_deriv_func

`ehsm_uint8_t ehsm_key_derived_param::key_deriv_func`

Definition at line 231 of file eHSM_Com_Struct_lp.h.

3.106.2.4 key_element_data

`ehsm_key_flags_element_st* ehsm_key_derived_param::key_element_data`

Definition at line 243 of file eHSM_Com_Struct_lp.h.

3.106.2.5 key_element_size

`ehsm_uint32_t ehsm_key_derived_param::key_element_size`

Definition at line 241 of file eHSM_Com_Struct_lp.h.

3.106.2.6 key_handle

`ehsm_uint32_t ehsm_key_derived_param::key_handle`

Definition at line 260 of file eHSM_Com_Struct_lp.h.

3.106.2.7 key_size

`ehsm_uint32_t ehsm_key_derived_param::key_size`

Definition at line 233 of file eHSM_Com_Struct_lp.h.

3.106.2.8 parent_key_author_size

`ehsm_uint32_t ehsm_key_derived_param::parent_key_author_size`

Definition at line 252 of file eHSM_Com_Struct_lp.h.

3.106.2.9 parent_key_author_value

`ehsm_uint8_t* ehsm_key_derived_param::parent_key_author_value`

Definition at line 254 of file eHSM_Com_Struct_Ip.h.

3.106.2.10 parent_key_handle

`ehsm_uint32_t ehsm_key_derived_param::parent_key_handle`

Definition at line 250 of file eHSM_Com_Struct_Ip.h.

3.106.2.11 passwd

`ehsm_uint8_t* ehsm_key_derived_param::passwd`

Definition at line 245 of file eHSM_Com_Struct_Ip.h.

3.106.2.12 passwd_size

`ehsm_uint32_t ehsm_key_derived_param::passwd_size`

Definition at line 246 of file eHSM_Com_Struct_Ip.h.

3.106.2.13 salt_data

`ehsm_uint8_t* ehsm_key_derived_param::salt_data`

Definition at line 258 of file eHSM_Com_Struct_Ip.h.

3.106.2.14 salt_size

`ehsm_uint32_t ehsm_key_derived_param::salt_size`

Definition at line 256 of file eHSM_Com_Struct_Ip.h.

3.106.2.15 type

[ehsm_key_mem_type_e](#) ehsm_key_derived_param::type

Definition at line 239 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.107 ehsm_key_flags_element_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint16_t](#) use_flags
- [ehsm_uint8_t](#) trnsp_flags
- [ehsm_uint32_t](#) auth_flag
- [ehsm_uint8_t](#) auth_size
- [ehsm_uint16_t](#) auth_value_exist_flags
- [ehsm_uint8_t](#) auth_value [EHSM_EVITA_AUTH_VALUE_MAX_SIZE]

3.107.1 Detailed Description

Definition at line 123 of file eHSM_Com_Struct_Ip.h.

3.107.2 Member Data Documentation

3.107.2.1 auth_flag

[ehsm_uint32_t](#) ehsm_key_flags_element_st::auth_flag

Definition at line 130 of file eHSM_Com_Struct_Ip.h.

3.107.2.2 auth_size

[ehsm_uint8_t](#) ehsm_key_flags_element_st::auth_size

Definition at line 132 of file eHSM_Com_Struct_Ip.h.

3.107.2.3 auth_value

```
ehsm_uint8_t ehsm_key_flags_element_st::auth_value[EHSM_EVITA_AUTH_VALUE_MAX_SIZE]
```

Definition at line 136 of file eHSM_Com_Struct_Ip.h.

3.107.2.4 auth_value_exist_flags

```
ehsm_uint16_t ehsm_key_flags_element_st::auth_value_exist_flags
```

Definition at line 134 of file eHSM_Com_Struct_Ip.h.

3.107.2.5 trnsp_flags

```
ehsm_uint8_t ehsm_key_flags_element_st::trnsp_flags
```

Definition at line 128 of file eHSM_Com_Struct_Ip.h.

3.107.2.6 use_flags

```
ehsm_uint16_t ehsm_key_flags_element_st::use_flags
```

Definition at line 126 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.108 ehsm_key_remove_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t reserved1](#) [8]
- [ehsm_uint8_t key_handle](#) [4]
- [ehsm_uint8_t key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_auth_size](#) [4]

3.108.1 Detailed Description

Definition at line 358 of file eHSM_Mailbox_Prtcl_Ip.h.

3.108.2 Member Data Documentation

3.108.2.1 key_auth_size

```
ehsm_uint8_t ehsm_key_remove_cmd::key_auth_size[4]
```

Definition at line 363 of file eHSM_Mailbox_Prtcl_Ip.h.

3.108.2.2 key_auth_value

```
ehsm_uint8_t ehsm_key_remove_cmd::key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 362 of file eHSM_Mailbox_Prtcl_Ip.h.

3.108.2.3 key_handle

```
ehsm_uint8_t ehsm_key_remove_cmd::key_handle[4]
```

Definition at line 361 of file eHSM_Mailbox_Prtcl_Ip.h.

3.108.2.4 reserved1

```
ehsm_uint8_t ehsm_key_remove_cmd::reserved1[8]
```

Definition at line 360 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.109 ehsm_key_remove_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint8_t](#) * key_auth_value
- [ehsm_uint32_t](#) key_auth_size

3.109.1 Detailed Description

Definition at line 394 of file eHSM_Com_Struct_Ip.h.

3.109.2 Member Data Documentation

3.109.2.1 key_auth_size

[ehsm_uint32_t](#) ehsm_key_remove_param::key_auth_size

Definition at line 398 of file eHSM_Com_Struct_Ip.h.

3.109.2.2 key_auth_value

[ehsm_uint8_t*](#) ehsm_key_remove_param::key_auth_value

Definition at line 397 of file eHSM_Com_Struct_Ip.h.

3.109.2.3 key_handle

[ehsm_uint32_t](#) ehsm_key_remove_param::key_handle

Definition at line 396 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.110 ehsm_key_signature_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) sign_id
- [ehsm_uint32_t](#) sign_info
- [ehsm_uint32_t](#) sign_key_id
- [ehsm_uint32_t](#) target_key_id
- [ehsm_uint8_t](#) signatrue [EVITA_KEY_SIGNATRUE_SIZE]

3.110.1 Detailed Description

Definition at line 263 of file eHSM_If_Evita_Types_Ip.h.

3.110.2 Member Data Documentation

3.110.2.1 sign_id

`ehsm_uint32_t ehsm_key_signature_::sign_id`

Definition at line 265 of file eHSM_If_Evita_Types_Ip.h.

3.110.2.2 sign_info

`ehsm_uint32_t ehsm_key_signature_::sign_info`

Definition at line 266 of file eHSM_If_Evita_Types_Ip.h.

3.110.2.3 sign_key_id

`ehsm_uint32_t ehsm_key_signature_::sign_key_id`

Definition at line 267 of file eHSM_If_Evita_Types_Ip.h.

3.110.2.4 signatrue

`ehsm_uint8_t ehsm_key_signature_::signatrue[EVITA_KEY_SIGNATRUE_SIZE]`

Definition at line 269 of file eHSM_If_Evita_Types_Ip.h.

3.110.2.5 target_key_id

`ehsm_uint32_t ehsm_key_signature_::target_key_id`

Definition at line 268 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.111 ehsm_key_status_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) keyId [4]
- [ehsm_uint32_t](#) keyIdSize
- [ehsm_uint32_t](#) algo_id
- [ehsm_uint32_t](#) valid_util
- [key_act_use_flags_t](#) activeUseFlag
- [ehsm_uint32_t](#) mem_location
- [ehsm_key_signatrue_st](#) key_sign_data
- [ehsm_pubkey_data_st](#) pubkey
- [ehsm_uint32_t](#) cert_size
- [ehsm_uint8_t](#) cert_data [512]

3.111.1 Detailed Description

Definition at line 328 of file eHSM_If_Evita_Types_Ip.h.

3.111.2 Member Data Documentation

3.111.2.1 activeUseFlag

```
key\_act\_use\_flags\_t ehsm_key_status_::activeUseFlag
```

Definition at line 334 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.2 algo_id

```
ehsm\_uint32\_t ehsm_key_status_::algo_id
```

Definition at line 332 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.3 cert_data

```
ehsm\_uint8\_t ehsm_key_status_::cert_data[512]
```

Definition at line 339 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.4 cert_size

`ehsm_uint32_t ehsm_key_status_::cert_size`

Definition at line 338 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.5 key_sign_data

`ehsm_key_signatrue_st ehsm_key_status_::key_sign_data`

Definition at line 336 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.6 keyId

`ehsm_uint8_t ehsm_key_status_::keyId[4]`

Definition at line 330 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.7 keyIdSize

`ehsm_uint32_t ehsm_key_status_::keyIdSize`

Definition at line 331 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.8 mem_location

`ehsm_uint32_t ehsm_key_status_::mem_location`

Definition at line 335 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.9 pubkey

`ehsm_pubkey_data_st ehsm_key_status_::pubkey`

Definition at line 337 of file eHSM_If_Evita_Types_Ip.h.

3.111.2.10 valid_util

```
ehsm_uint32_t ehsm_key_status_::valid_util
```

Definition at line 333 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.112 ehsm_key_status_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) key_handle [4]
- [ehsm_uint8_t](#) reserved1 [4]
- [ehsm_uint8_t](#) cert_key_handle [4]
- [ehsm_uint8_t](#) cert_key_auth_value [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) cert_key_auth_size [4]
- [ehsm_uint8_t](#) reserved2 [8]
- [ehsm_uint8_t](#) key_status [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) key_status_size [4]

3.112.1 Detailed Description

Definition at line 366 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2 Member Data Documentation

3.112.2.1 cert_key_auth_size

```
ehsm_uint8_t ehsm_key_status_cmd::cert_key_auth_size[4]
```

Definition at line 372 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.2 cert_key_auth_value

```
ehsm_uint8_t ehsm_key_status_cmd::cert_key_auth_value[HOST_ADDRESS_SIZE]
```

Definition at line 371 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.3 cert_key_handle

```
ehsm_uint8_t ehsm_key_status_cmd::cert_key_handle[4]
```

Definition at line 370 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.4 key_handle

```
ehsm_uint8_t ehsm_key_status_cmd::key_handle[4]
```

Definition at line 368 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.5 key_status

```
ehsm_uint8_t ehsm_key_status_cmd::key_status[HOST_ADDRESS_SIZE]
```

Definition at line 374 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.6 key_status_size

```
ehsm_uint8_t ehsm_key_status_cmd::key_status_size[4]
```

Definition at line 375 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.7 reserved1

```
ehsm_uint8_t ehsm_key_status_cmd::reserved1[4]
```

Definition at line 369 of file eHSM_Mailbox_Prtcl_Ip.h.

3.112.2.8 reserved2

```
ehsm_uint8_t ehsm_key_status_cmd::reserved2[8]
```

Definition at line 373 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.113 ehsm_key_status_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) certification_key_handle
- [ehsm_uint32_t](#) certification_key_auth_size
- [ehsm_uint8_t](#) * certification_key_auth_value
- [ehsm_uint32_t](#) key_status_size
- [ehsm_uint8_t](#) * key_status
- [ehsm_uint32_t](#) key_status_buffer_size

3.113.1 Detailed Description

Definition at line 401 of file eHSM_Com_Struct_Ip.h.

3.113.2 Member Data Documentation

3.113.2.1 certification_key_auth_size

```
ehsm\_uint32\_t ehsm_key_status_param::certification_key_auth_size
```

Definition at line 409 of file eHSM_Com_Struct_Ip.h.

3.113.2.2 certification_key_auth_value

```
ehsm\_uint8\_t* ehsm_key_status_param::certification_key_auth_value
```

Definition at line 411 of file eHSM_Com_Struct_Ip.h.

3.113.2.3 certification_key_handle

```
ehsm\_uint32\_t ehsm_key_status_param::certification_key_handle
```

Definition at line 407 of file eHSM_Com_Struct_Ip.h.

3.113.2.4 key_handle

`ehsm_uint32_t ehsm_key_status_param::key_handle`

Definition at line 404 of file eHSM_Com_Struct_Ip.h.

3.113.2.5 key_status

`ehsm_uint8_t* ehsm_key_status_param::key_status`

Definition at line 416 of file eHSM_Com_Struct_Ip.h.

3.113.2.6 key_status_buffer_size

`ehsm_uint32_t ehsm_key_status_param::key_status_buffer_size`

Definition at line 418 of file eHSM_Com_Struct_Ip.h.

3.113.2.7 key_status_size

`ehsm_uint32_t ehsm_key_status_param::key_status_size`

Definition at line 413 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.114 ehsm_key_usages_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_key_flags_element_st sign](#)
- [ehsm_key_flags_element_st verify](#)
- [ehsm_key_flags_element_st encrypt](#)
- [ehsm_key_flags_element_st decrypt](#)
- [ehsm_key_flags_element_st timestamp](#)
- [ehsm_key_flags_element_st secureboot](#)
- [ehsm_key_flags_element_st securestorage](#)
- [ehsm_key_flags_element_st dhkey](#)
- [ehsm_key_flags_element_st utcsync](#)
- [ehsm_key_flags_element_st transport](#)
- [ehsm_key_flags_element_st remove](#)

3.114.1 Detailed Description

Definition at line 139 of file eHSM_Com_Struct_lp.h.

3.114.2 Member Data Documentation

3.114.2.1 decrypt

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::decrypt

Definition at line 144 of file eHSM_Com_Struct_lp.h.

3.114.2.2 dhkey

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::dhkey

Definition at line 148 of file eHSM_Com_Struct_lp.h.

3.114.2.3 encrypt

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::encrypt

Definition at line 143 of file eHSM_Com_Struct_lp.h.

3.114.2.4 remove

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::remove

Definition at line 151 of file eHSM_Com_Struct_lp.h.

3.114.2.5 secureboot

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::secureboot

Definition at line 146 of file eHSM_Com_Struct_lp.h.

3.114.2.6 securestorage

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::securestorage

Definition at line 147 of file eHSM_Com_Struct_lp.h.

3.114.2.7 sign

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::sign

Definition at line 141 of file eHSM_Com_Struct_lp.h.

3.114.2.8 timestamp

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::timestamp

Definition at line 145 of file eHSM_Com_Struct_lp.h.

3.114.2.9 transport

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::transport

Definition at line 150 of file eHSM_Com_Struct_lp.h.

3.114.2.10 utcsync

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::utcsync

Definition at line 149 of file eHSM_Com_Struct_lp.h.

3.114.2.11 verify

[ehsm_key_flags_element_st](#) ehsm_key_usages_st::verify

Definition at line 142 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.115 ehsm_keyexchange_key_info Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) key_auth_size
- [ehsm_uint8_t](#) key_auth_value [32]

3.115.1 Detailed Description

Definition at line 466 of file eHSM_Com_Struct_Ip.h.

3.115.2 Member Data Documentation

3.115.2.1 key_auth_size

[ehsm_uint32_t](#) ehsm_keyexchange_key_info::key_auth_size

Definition at line 471 of file eHSM_Com_Struct_Ip.h.

3.115.2.2 key_auth_value

[ehsm_uint8_t](#) ehsm_keyexchange_key_info::key_auth_value[32]

Definition at line 473 of file eHSM_Com_Struct_Ip.h.

3.115.2.3 key_handle

[ehsm_uint32_t](#) ehsm_keyexchange_key_info::key_handle

Definition at line 469 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.116 ehsm_low_power_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t reserved](#) [4]
- [ehsm_uint8_t power_mode](#)

3.116.1 Detailed Description

Definition at line 585 of file eHSM_Mailbox_Prtcl_Ip.h.

3.116.2 Member Data Documentation

3.116.2.1 power_mode

[ehsm_uint8_t](#) ehsm_low_power_cmd::power_mode

Definition at line 587 of file eHSM_Mailbox_Prtcl_Ip.h.

3.116.2.2 reserved

[ehsm_uint8_t](#) ehsm_low_power_cmd::reserved[4]

Definition at line 586 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.117 ehsm_mailbox_req Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- int `cmd_id`
General service id.
- `ehsm_uint8_t` `api_type`
- `ehsm_uint8_t` `rev` [7]
Reserved.
- union {
 - `ehsm_derive_key_cmd_st` `derive_key`
 - `ehsm_import_key_cmd_st` `import_key`
 - `ehsm_export_key_cmd_st` `export_key`
 - `ehsm_get_challenge_cmd_st` `get_challenge`
 - `ehsm_debug_authentication_cmd_st` `debug_authentication`
 - `ehsm_fw_get_random_key_cmd_st` `fw_random_key`
 - `ehsm_fw_encrypt_key_cmd_st` `fw_encrypt_key`
 - `ehsm_image_upgrade_cmd_st` `image_upgrade`
 - `ehsm_image_verify_cmd_st` `image_verify`
 - `ehsm_soc_image_verify_cmd_st` `soc_image_verify`
 - `ehsm_create_dh_key_cmd_st` `create_dh_key`
 - `ehsm_get_pub_from_priv_cmd_st` `get_pub_from_priv`
 - `ehsm_key_remove_cmd_st` `key_remove`
 - `ehsm_key_status_cmd_st` `key_status`
 - `ehsm_copy_key_cmd_st` `copy_key`
 - `ehsm_she_load_export_key_cmd_st` `she_load_export_key`
 - `ehsm_she_load_plain_key_cmd_st` `she_load_plain_key`
 - `ehsm_gen_key_cmd_st` `ehsm_gen_key`
 - `ehsm_gen_sm9_userpriv_key_cmd_st` `ehsm_gen_sm9_userpriv_key`
 - `ehsm_sm9_exchg_gen_usertmp_cmd_st` `gen_usertmp`
 - `ehsm_sm9_exchg_key_cmd_st` `sm9_exchg_key`
 - `ehsm_sm9_wrap_key_cmd_st` `sm9_wrap_key`
 - `ehsm_sm9_unwrap_key_cmd_st` `sm9_unwrap_key`
 - `ehsm_sm9_export_key_cmd_st` `sm9_export_key`
 - `ehsm_sm9_import_key_cmd_st` `sm9_import_key`
 - `ehsm_sm9_get_mast_pubkey_cmd_st` `get_mast_pubkey`
 - `ehsm_sm9_get_tmp_pubkey_cmd_st` `get_tmp_pubkey`
 - `ehsm_sm9_remove_key_cmd_st` `sm9_remove_key`
 - `ehsm_otp_read_cmd_st` `otp_read`
 - `ehsm_otp_write_cmd_st` `otp_write`
 - `ehsm_uart_cmd_st` `uart_cmd`
 - `ehsm_get_she_id_cmd_st` `get_she_id`
 - `ehsm_get_emu_cmd_st` `get_emu`
 - `ehsm_module_status_cmd_st` `module_status`
 - `ehsm_soc_secure_boot_status_st` `soc_boot`
 - `ehsm_close_debug_cmd_st` `close_debug`
- } `ehsm_cmd`

3.117.1 Detailed Description

Definition at line 650 of file `eHSM_Mailbox_Prtcl_Ip.h`.

3.117.2 Member Data Documentation

3.117.2.1 api_type

`ehsm_uint8_t ehsm_mailbox_req::api_type`

Definition at line 653 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.2 close_debug

`ehsm_close_debug_cmd_st ehsm_mailbox_req::close_debug`

Definition at line 700 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.3 cmd_id

`int ehsm_mailbox_req::cmd_id`

General service id.

Definition at line 652 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.4 copy_key

`ehsm_copy_key_cmd_st ehsm_mailbox_req::copy_key`

Definition at line 678 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.5 create_dh_key

`ehsm_create_dh_key_cmd_st ehsm_mailbox_req::create_dh_key`

Definition at line 674 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.6 debug_authentication

`ehsm_debug_authentication_cmd_st ehsm_mailbox_req::debug_authentication`

Definition at line 662 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.7 derive_key

`ehsm_derive_key_cmd_st` ehsm_mailbox_req::derive_key

Definition at line 658 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.8 ehsm_cmd

`union { ... }` ehsm_mailbox_req::ehsm_cmd

3.117.2.9 ehsm_gen_key

`ehsm_gen_key_cmd_st` ehsm_mailbox_req::ehsm_gen_key

Definition at line 681 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.10 ehsm_gen_sm9_userpriv_key

`ehsm_gen_sm9_userpriv_key_cmd_st` ehsm_mailbox_req::ehsm_gen_sm9_userpriv_key

Definition at line 682 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.11 export_key

`ehsm_export_key_cmd_st` ehsm_mailbox_req::export_key

Definition at line 660 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.12 fw_encrypt_key

`ehsm_fw_encrypt_key_cmd_st` ehsm_mailbox_req::fw_encrypt_key

Definition at line 670 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.13 fw_random_key

[ehsm_fw_get_random_key_cmd_st](#) ehsm_mailbox_req::fw_random_key

Definition at line 669 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.14 gen_usertmp

[ehsm_sm9_exchg_gen_usertmp_cmd_st](#) ehsm_mailbox_req::gen_usertmp

Definition at line 683 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.15 get_challenge

[ehsm_get_challenge_cmd_st](#) ehsm_mailbox_req::get_challenge

Definition at line 661 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.16 get_emu

[ehsm_get_emu_cmd_st](#) ehsm_mailbox_req::get_emu

Definition at line 697 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.17 get_mast_pubkey

[ehsm_sm9_get_mast_pubkey_cmd_st](#) ehsm_mailbox_req::get_mast_pubkey

Definition at line 689 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.18 get_pub_from_priv

[ehsm_get_pub_from_priv_cmd_st](#) ehsm_mailbox_req::get_pub_from_priv

Definition at line 675 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.19 get_she_id

[ehsm_get_she_id_cmd_st](#) ehsm_mailbox_req::get_she_id

Definition at line 696 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.20 get_tmp_pubkey

[ehsm_sm9_get_tmp_pubkey_cmd_st](#) ehsm_mailbox_req::get_tmp_pubkey

Definition at line 690 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.21 image_upgrade

[ehsm_image_upgrade_cmd_st](#) ehsm_mailbox_req::image_upgrade

Definition at line 671 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.22 image_verify

[ehsm_image_verify_cmd_st](#) ehsm_mailbox_req::image_verify

Definition at line 672 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.23 import_key

[ehsm_import_key_cmd_st](#) ehsm_mailbox_req::import_key

Definition at line 659 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.24 key_remove

[ehsm_key_remove_cmd_st](#) ehsm_mailbox_req::key_remove

Definition at line 676 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.25 key_status

`ehsm_key_status_cmd_st` ehsm_mailbox_req::key_status

Definition at line 677 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.26 module_status

`ehsm_module_status_cmd_st` ehsm_mailbox_req::module_status

Definition at line 698 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.27 otp_read

`ehsm_otp_read_cmd_st` ehsm_mailbox_req::otp_read

Definition at line 693 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.28 otp_write

`ehsm_otp_write_cmd_st` ehsm_mailbox_req::otp_write

Definition at line 694 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.29 rev

`ehsm_uint8_t` ehsm_mailbox_req::rev[7]

Reserved.

Definition at line 655 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.30 she_load_export_key

`ehsm_she_load_export_key_cmd_st` ehsm_mailbox_req::she_load_export_key

Definition at line 679 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.31 she_load_plain_key

`ehsm_she_load_plain_key_cmd_st` ehsm_mailbox_req::she_load_plain_key

Definition at line 680 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.32 sm9_exchg_key

`ehsm_sm9_exchg_key_cmd_st` ehsm_mailbox_req::sm9_exchg_key

Definition at line 684 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.33 sm9_export_key

`ehsm_sm9_export_key_cmd_st` ehsm_mailbox_req::sm9_export_key

Definition at line 687 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.34 sm9_import_key

`ehsm_sm9_import_key_cmd_st` ehsm_mailbox_req::sm9_import_key

Definition at line 688 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.35 sm9_remove_key

`ehsm_sm9_remove_key_cmd_st` ehsm_mailbox_req::sm9_remove_key

Definition at line 691 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.36 sm9_unwrap_key

`ehsm_sm9_unwrap_key_cmd_st` ehsm_mailbox_req::sm9_unwrap_key

Definition at line 686 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.37 sm9_wrap_key

[ehsm_sm9_wrap_key_cmd_st](#) ehsm_mailbox_req::sm9_wrap_key

Definition at line 685 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.38 soc_boot

[ehsm_soc_secure_boot_status_st](#) ehsm_mailbox_req::soc_boot

Definition at line 699 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.39 soc_image_verify

[ehsm_soc_image_verify_cmd_st](#) ehsm_mailbox_req::soc_image_verify

Definition at line 673 of file eHSM_Mailbox_Prtcl_Ip.h.

3.117.2.40 uart_cmd

[ehsm_uart_cmd_st](#) ehsm_mailbox_req::uart_cmd

Definition at line 695 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.118 ehsm_mbox_cancel_channel_req Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) cancel_type
- [ehsm_uint8_t](#) api_type
- [ehsm_uint8_t](#) rev [2]
- [ehsm_uint8_t](#) cmd_tag [HOST_ADDRESS_SIZE]

3.118.1 Detailed Description

Definition at line 728 of file eHSM_Mailbox_Prtcl_Ip.h.

3.118.2 Member Data Documentation

3.118.2.1 api_type

[ehsm_uint8_t](#) ehsm_mbox_cancel_channel_req::api_type

Definition at line 733 of file eHSM_Mailbox_Prtcl_Ip.h.

3.118.2.2 cancel_type

[ehsm_uint8_t](#) ehsm_mbox_cancel_channel_req::cancel_type

Definition at line 731 of file eHSM_Mailbox_Prtcl_Ip.h.

3.118.2.3 cmd_tag

[ehsm_uint8_t](#) ehsm_mbox_cancel_channel_req::cmd_tag [[HOST_ADDRESS_SIZE](#)]

Definition at line 736 of file eHSM_Mailbox_Prtcl_Ip.h.

3.118.2.4 rev

[ehsm_uint8_t](#) ehsm_mbox_cancel_channel_req::rev [2]

Definition at line 734 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.119 ehsm_mbox_cancel_channel_rps Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) ret_code [4]
- [ehsm_uint8_t](#) cancel_type
- [ehsm_uint8_t](#) api_type
- [ehsm_uint8_t](#) rev [2]
- [ehsm_uint8_t](#) cmd_tag [[HOST_ADDRESS_SIZE](#)]

3.119.1 Detailed Description

Definition at line 739 of file eHSM_Mailbox_Prtcl_Ip.h.

3.119.2 Member Data Documentation

3.119.2.1 api_type

```
ehsm_uint8_t ehsm_mbox_cancel_channel_rps::api_type
```

Definition at line 745 of file eHSM_Mailbox_Prtcl_Ip.h.

3.119.2.2 cancel_type

```
ehsm_uint8_t ehsm_mbox_cancel_channel_rps::cancel_type
```

Definition at line 743 of file eHSM_Mailbox_Prtcl_Ip.h.

3.119.2.3 cmd_tag

```
ehsm_uint8_t ehsm_mbox_cancel_channel_rps::cmd_tag[HOST_ADDRESS_SIZE]
```

Definition at line 748 of file eHSM_Mailbox_Prtcl_Ip.h.

3.119.2.4 ret_code

```
ehsm_uint8_t ehsm_mbox_cancel_channel_rps::ret_code[4]
```

Definition at line 740 of file eHSM_Mailbox_Prtcl_Ip.h.

3.119.2.5 rev

```
ehsm_uint8_t ehsm_mbox_cancel_channel_rps::rev[2]
```

Definition at line 746 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.120 ehsm_mbox_mgr_channel_req Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- int `cmd_id`
General service id.
- union {
 - `ehsm_low_power_cmd_st` `low_power_cmd`
 - `ehsm_self_test_cmd_st` `self_test_cmd`
 - `ehsm_set_baudrate_cmd_st` `set_baudrate_cmd`
 - `ehsm_change_lifecycle_cmd_st` `change_lifecycle_cmd`
 - `ehsm_change_control_field_cmd_st` `change_control_field_cmd`
 - `ehsm_sensor_resp_init_cmd_st` `sensor_resp_init_cmd`
- } `ehsm_cmd`

3.120.1 Detailed Description

Definition at line 713 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2 Member Data Documentation

3.120.2.1 change_control_field_cmd

```
ehsm_change_control_field_cmd_st ehsm_mbox_mgr_channel_req::change_control_field_cmd
```

Definition at line 721 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2.2 change_lifecycle_cmd

```
ehsm_change_lifecycle_cmd_st ehsm_mbox_mgr_channel_req::change_lifecycle_cmd
```

Definition at line 720 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2.3 cmd_id

```
int ehsm_mbox_mgr_channel_req::cmd_id
```

General service id.

Definition at line 715 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2.4 ehsm_cmd

```
union { ... } ehsm_mbox_mgr_channel_req::ehsm_cmd
```

3.120.2.5 low_power_cmd

```
ehsm_low_power_cmd_st ehsm_mbox_mgr_channel_req::low_power_cmd
```

Definition at line 717 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2.6 self_test_cmd

```
ehsm_self_test_cmd_st ehsm_mbox_mgr_channel_req::self_test_cmd
```

Definition at line 718 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2.7 sensor_resp_init_cmd

```
ehsm_sensor_resp_init_cmd_st ehsm_mbox_mgr_channel_req::sensor_resp_init_cmd
```

Definition at line 722 of file eHSM_Mailbox_Prtcl_Ip.h.

3.120.2.8 set_baudrate_cmd

```
ehsm_set_baudrate_cmd_st ehsm_mbox_mgr_channel_req::set_baudrate_cmd
```

Definition at line 719 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.121 ehsm_module_status_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) algo_id
- [ehsm_uint8_t](#) reserved [6]
- [ehsm_uint8_t](#) key_handle [4]
- [ehsm_uint8_t](#) key_auth_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) key_auth_size
- [ehsm_uint8_t](#) status_addr [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) status_size
- [ehsm_uint8_t](#) signatrue [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t](#) signatrue_size

3.121.1 Detailed Description

Definition at line 631 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2 Member Data Documentation

3.121.2.1 algo_id

[ehsm_uint8_t](#) ehsm_module_status_cmd::algo_id

Definition at line 633 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.2 key_auth_addr

[ehsm_uint8_t](#) ehsm_module_status_cmd::key_auth_addr [HOST_ADDRESS_SIZE]

Definition at line 636 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.3 key_auth_size

[ehsm_uint32_t](#) ehsm_module_status_cmd::key_auth_size

Definition at line 637 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.4 key_handle

[ehsm_uint8_t](#) ehsm_module_status_cmd::key_handle [4]

Definition at line 635 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.5 reserved

```
ehsm_uint8_t ehsm_module_status_cmd::reserved[6]
```

Definition at line 634 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.6 signatrue

```
ehsm_uint8_t ehsm_module_status_cmd::signatrue[HOST_ADDRESS_SIZE]
```

Definition at line 640 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.7 signatrue_size

```
ehsm_uint32_t ehsm_module_status_cmd::signatrue_size
```

Definition at line 641 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.8 status_addr

```
ehsm_uint8_t ehsm_module_status_cmd::status_addr[HOST_ADDRESS_SIZE]
```

Definition at line 638 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.9 status_size

```
ehsm_uint32_t ehsm_module_status_cmd::status_size
```

Definition at line 639 of file eHSM_Mailbox_Prtcl_Ip.h.

3.121.2.10 type

```
ehsm_uint8_t ehsm_module_status_cmd::type
```

Definition at line 632 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.122 ehsm_module_status_st Struct Reference

The structure to store the information of ehsm status. The parameter alg, key_handle, key_auth_size, key_auth_value, sign_size and sign are only valid when type is not EHSM_GET_STATUS_SHE.

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) type
- [ehsm_uint32_t](#) algo_id
- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) key_auth_size
- [const ehsm_uint8_t *](#)key_auth_value
- [ehsm_uint32_t *](#)status_size
- [ehsm_uint8_t *](#)status
- [ehsm_uint32_t *](#)sign_size
- [ehsm_uint8_t *](#)sign

3.122.1 Detailed Description

The structure to store the information of ehsm status. The parameter alg, key_handle, key_auth_size, key_auth_value, sign_size and sign are only valid when type is not EHSM_GET_STATUS_SHE.

Definition at line 1003 of file eHSM_Com_Struct_Ip.h.

3.122.2 Member Data Documentation

3.122.2.1 algo_id

[ehsm_uint32_t](#) ehsm_module_status_st::algo_id

Definition at line 1010 of file eHSM_Com_Struct_Ip.h.

3.122.2.2 key_auth_size

[ehsm_uint32_t](#) ehsm_module_status_st::key_auth_size

Definition at line 1014 of file eHSM_Com_Struct_Ip.h.

3.122.2.3 key_auth_value

```
const ehsm_uint8_t* ehsm_module_status_st::key_auth_value
```

Definition at line 1016 of file eHSM_Com_Struct_Ip.h.

3.122.2.4 key_handle

```
ehsm_uint32_t ehsm_module_status_st::key_handle
```

Definition at line 1012 of file eHSM_Com_Struct_Ip.h.

3.122.2.5 sign

```
ehsm_uint8_t* ehsm_module_status_st::sign
```

Definition at line 1024 of file eHSM_Com_Struct_Ip.h.

3.122.2.6 sign_size

```
ehsm_uint32_t* ehsm_module_status_st::sign_size
```

Definition at line 1022 of file eHSM_Com_Struct_Ip.h.

3.122.2.7 status

```
ehsm_uint8_t* ehsm_module_status_st::status
```

Definition at line 1020 of file eHSM_Com_Struct_Ip.h.

3.122.2.8 status_size

```
ehsm_uint32_t* ehsm_module_status_st::status_size
```

Definition at line 1018 of file eHSM_Com_Struct_Ip.h.

3.122.2.9 type

`ehsm_uint32_t ehsm_module_status_st::type`

Definition at line 1008 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.123 ehsm_otp_read_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- `ehsm_uint8_t ehsm_src_addr` [4]
- `ehsm_uint8_t size` [4]
- `ehsm_uint8_t host_dst_addr` [HOST_ADDRESS_SIZE]

3.123.1 Detailed Description

Definition at line 99 of file eHSM_Mailbox_Prtcl_Ip.h.

3.123.2 Member Data Documentation

3.123.2.1 ehsm_src_addr

`ehsm_uint8_t ehsm_otp_read_cmd::ehsm_src_addr` [4]

Definition at line 100 of file eHSM_Mailbox_Prtcl_Ip.h.

3.123.2.2 host_dst_addr

`ehsm_uint8_t ehsm_otp_read_cmd::host_dst_addr` [HOST_ADDRESS_SIZE]

Definition at line 102 of file eHSM_Mailbox_Prtcl_Ip.h.

3.123.2.3 size

```
ehsm_uint8_t ehsm_otp_read_cmd::size[4]
```

Definition at line 101 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.124 ehsm_otp_read_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) flash_read_addr
- [ehsm_uint32_t](#) read_data_size
- [ehsm_uint8_t](#) * otp_data_addr

3.124.1 Detailed Description

Definition at line 920 of file eHSM_Com_Struct_Ip.h.

3.124.2 Member Data Documentation

3.124.2.1 flash_read_addr

```
ehsm_uint32_t ehsm_otp_read_param_st::flash_read_addr
```

Definition at line 921 of file eHSM_Com_Struct_Ip.h.

3.124.2.2 otp_data_addr

```
ehsm_uint8_t* ehsm_otp_read_param_st::otp_data_addr
```

Definition at line 923 of file eHSM_Com_Struct_Ip.h.

3.124.2.3 read_data_size

`ehsm_uint32_t ehsm_otp_read_param_st::read_data_size`

Definition at line 922 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.125 ehsm_otp_write_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- `ehsm_uint8_t ehsm_dst_addr` [4]
- `ehsm_uint8_t size` [4]
- `ehsm_uint8_t host_src_addr` [HOST_ADDRESS_SIZE]

3.125.1 Detailed Description

Definition at line 105 of file eHSM_Mailbox_Prtcl_Ip.h.

3.125.2 Member Data Documentation

3.125.2.1 ehsm_dst_addr

`ehsm_uint8_t ehsm_otp_write_cmd::ehsm_dst_addr` [4]

Definition at line 106 of file eHSM_Mailbox_Prtcl_Ip.h.

3.125.2.2 host_src_addr

`ehsm_uint8_t ehsm_otp_write_cmd::host_src_addr` [HOST_ADDRESS_SIZE]

Definition at line 108 of file eHSM_Mailbox_Prtcl_Ip.h.

3.125.2.3 size

```
ehsm_uint8_t ehsm_otp_write_cmd::size[4]
```

Definition at line 107 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.126 ehsm_otp_write_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) flash_write_addr
- [ehsm_uint32_t](#) write_data_size
- [ehsm_uint8_t](#) * otp_data_addr

3.126.1 Detailed Description

Definition at line 926 of file eHSM_Com_Struct_Ip.h.

3.126.2 Member Data Documentation

3.126.2.1 flash_write_addr

```
ehsm_uint32_t ehsm_otp_write_param_st::flash_write_addr
```

Definition at line 927 of file eHSM_Com_Struct_Ip.h.

3.126.2.2 otp_data_addr

```
ehsm_uint8_t* ehsm_otp_write_param_st::otp_data_addr
```

Definition at line 929 of file eHSM_Com_Struct_Ip.h.

3.126.2.3 write_data_size

`ehsm_uint32_t ehsm_otp_write_param_st::write_data_size`

Definition at line 928 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.127 ehsm_prikey_data_ Union Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t sym_k](#) [1024]
- [ehsm_uint8_t ecc_k](#) [66]
- [ehsm_uint8_t rsa_d](#) [512]
- [ehsm_uint8_t kdf_k](#) [512]
- [ehsm_uint8_t dh_k](#) [512]
- [ehsm_dh_prikey_st dh](#)
- [ehsm_rsa_ctr_st rsa_crt](#)

3.127.1 Detailed Description

Definition at line 317 of file eHSM_If_Evita_Types_Ip.h.

3.127.2 Member Data Documentation

3.127.2.1 dh

`ehsm_dh_prikey_st ehsm_prikey_data_::dh`

Definition at line 324 of file eHSM_If_Evita_Types_Ip.h.

3.127.2.2 dh_k

`ehsm_uint8_t ehsm_prikey_data_::dh_k` [512]

Definition at line 323 of file eHSM_If_Evita_Types_Ip.h.

3.127.2.3 ecc_k

```
ehsm_uint8_t ehsm_prikey_data_::ecc_k[66]
```

Definition at line 320 of file eHSM_If_Evita_Types_Ip.h.

3.127.2.4 kdf_k

```
ehsm_uint8_t ehsm_prikey_data_::kdf_k[512]
```

Definition at line 322 of file eHSM_If_Evita_Types_Ip.h.

3.127.2.5 rsa_crt

```
ehsm_rsa_ctr_st ehsm_prikey_data_::rsa_crt
```

Definition at line 325 of file eHSM_If_Evita_Types_Ip.h.

3.127.2.6 rsa_d

```
ehsm_uint8_t ehsm_prikey_data_::rsa_d[512]
```

Definition at line 321 of file eHSM_If_Evita_Types_Ip.h.

3.127.2.7 sym_k

```
ehsm_uint8_t ehsm_prikey_data_::sym_k[1024]
```

Definition at line 319 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this union was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.128 ehsm_pub_key_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t algo_id](#)
- [ehsm_uint8_t key_size_info](#) [EHSM_KEY_SIZE_INFO_MAX_LEN]
- [ehsm_uint8_t key_pub_data](#) [0]

3.128.1 Detailed Description

Definition at line 407 of file eHSM_If_Evita_Types_Ip.h.

3.128.2 Member Data Documentation

3.128.2.1 algo_id

```
ehsm_uint32_t ehsm_pub_key_::algo_id
```

Definition at line 409 of file eHSM_If_Evita_Types_Ip.h.

3.128.2.2 key_pub_data

```
ehsm_uint8_t ehsm_pub_key_::key_pub_data[0]
```

Definition at line 411 of file eHSM_If_Evita_Types_Ip.h.

3.128.2.3 key_size_info

```
ehsm_uint8_t ehsm_pub_key_::key_size_info[EHSM_KEY_SIZE_INFO_MAX_LEN]
```

Definition at line 410 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.129 ehsm_pubkey_data_ Union Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_rsa_pubkey_st](#) `rsa`
- [ehsm_ecc_pubkey_st](#) `ecc`
- [ehsm_dh_pubkey_st](#) `dh`

3.129.1 Detailed Description

Definition at line 296 of file eHSM_If_Evita_Types_Ip.h.

3.129.2 Member Data Documentation

3.129.2.1 `dh`

[ehsm_dh_pubkey_st](#) `ehsm_pubkey_data_::dh`

Definition at line 300 of file eHSM_If_Evita_Types_Ip.h.

3.129.2.2 `ecc`

[ehsm_ecc_pubkey_st](#) `ehsm_pubkey_data_::ecc`

Definition at line 299 of file eHSM_If_Evita_Types_Ip.h.

3.129.2.3 `rsa`

[ehsm_rsa_pubkey_st](#) `ehsm_pubkey_data_::rsa`

Definition at line 298 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this union was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.130 ehsm_rng_generate_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t rev1](#) [3]
- [ehsm_uint8_t algorithm](#)
- [ehsm_uint8_t rev2](#) [4]
- [ehsm_uint32_t rev_key_handle](#)
- [ehsm_uint8_t rev_key_auth_value](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t rev_key_auth_size](#)
- [ehsm_uint8_t rev3](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t rev4](#)
- [ehsm_uint8_t random_data_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint32_t request_size](#)

3.130.1 Detailed Description

Definition at line 270 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2 Member Data Documentation

3.130.2.1 algorithm

[ehsm_uint8_t](#) ehsm_rng_generate_cmd::algorithm

Definition at line 272 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.2 random_data_addr

[ehsm_uint8_t](#) ehsm_rng_generate_cmd::random_data_addr [HOST_ADDRESS_SIZE]

Definition at line 279 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.3 request_size

[ehsm_uint32_t](#) ehsm_rng_generate_cmd::request_size

Definition at line 280 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.4 rev1

[ehsm_uint8_t](#) ehsm_rng_generate_cmd::rev1 [3]

Definition at line 271 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.5 rev2

`ehsm_uint8_t ehsm_rng_generate_cmd::rev2[4]`

Definition at line 273 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.6 rev3

`ehsm_uint8_t ehsm_rng_generate_cmd::rev3[HOST_ADDRESS_SIZE]`

Definition at line 277 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.7 rev4

`ehsm_uint32_t ehsm_rng_generate_cmd::rev4`

Definition at line 278 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.8 rev_key_auth_size

`ehsm_uint32_t ehsm_rng_generate_cmd::rev_key_auth_size`

Definition at line 276 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.9 rev_key_auth_value

`ehsm_uint8_t ehsm_rng_generate_cmd::rev_key_auth_value[HOST_ADDRESS_SIZE]`

Definition at line 275 of file eHSM_Mailbox_Prtcl_Ip.h.

3.130.2.10 rev_key_handle

`ehsm_uint32_t ehsm_rng_generate_cmd::rev_key_handle`

Definition at line 274 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.131 ehsm_rsa_cert_param_Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t p](#) [256]
- [ehsm_uint8_t q](#) [256]
- [ehsm_uint8_t dp](#) [256]
- [ehsm_uint8_t dq](#) [256]
- [ehsm_uint8_t u](#) [256]

3.131.1 Detailed Description

Definition at line 308 of file eHSM_If_Evita_Types_Ip.h.

3.131.2 Member Data Documentation

3.131.2.1 dp

```
ehsm_uint8_t ehsm_rsa_cert_param_::dp[256]
```

Definition at line 312 of file eHSM_If_Evita_Types_Ip.h.

3.131.2.2 dq

```
ehsm_uint8_t ehsm_rsa_cert_param_::dq[256]
```

Definition at line 313 of file eHSM_If_Evita_Types_Ip.h.

3.131.2.3 p

```
ehsm_uint8_t ehsm_rsa_cert_param_::p[256]
```

Definition at line 310 of file eHSM_If_Evita_Types_Ip.h.

3.131.2.4 q

```
ehsm_uint8_t ehsm_rsa crt_param::q[256]
```

Definition at line 311 of file eHSM_If_Evita_Types_Ip.h.

3.131.2.5 u

```
ehsm_uint8_t ehsm_rsa crt_param::u[256]
```

Definition at line 314 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.132 ehsm_rsa_dh_key_size_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint16_t rsa_dh_p_size](#)
- [ehsm_uint16_t rsa_dh_q_size](#)
- [ehsm_uint16_t rsa_dh_g_size](#)
- [ehsm_uint8_t reserved](#) [2]

3.132.1 Detailed Description

Definition at line 388 of file eHSM_If_Evita_Types_Ip.h.

3.132.2 Member Data Documentation

3.132.2.1 reserved

```
ehsm_uint8_t ehsm_rsa_dh_key_size::reserved[2]
```

Definition at line 393 of file eHSM_If_Evita_Types_Ip.h.

3.132.2.2 rsa_dh_g_size

[ehsm_uint16_t](#) ehsm_rsa_dh_key_size_::rsa_dh_g_size

Definition at line 392 of file eHSM_If_Evita_Types_Ip.h.

3.132.2.3 rsa_dh_p_size

[ehsm_uint16_t](#) ehsm_rsa_dh_key_size_::rsa_dh_p_size

Definition at line 390 of file eHSM_If_Evita_Types_Ip.h.

3.132.2.4 rsa_dh_q_size

[ehsm_uint16_t](#) ehsm_rsa_dh_key_size_::rsa_dh_q_size

Definition at line 391 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.133 ehsm_rsa_key_size_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint16_t](#) rsa_n_size
- [ehsm_uint16_t](#) rsa_e_size
- [ehsm_uint16_t](#) rsa_d_size
- [ehsm_uint16_t](#) reserved [2]

3.133.1 Detailed Description

Definition at line 380 of file eHSM_If_Evita_Types_Ip.h.

3.133.2 Member Data Documentation

3.133.2.1 reserved

```
ehsm_uint16_t ehsm_rsa_key_size_::reserved[2]
```

Definition at line 385 of file eHSM_If_Evita_Types_Ip.h.

3.133.2.2 rsa_d_size

```
ehsm_uint16_t ehsm_rsa_key_size_::rsa_d_size
```

Definition at line 384 of file eHSM_If_Evita_Types_Ip.h.

3.133.2.3 rsa_e_size

```
ehsm_uint16_t ehsm_rsa_key_size_::rsa_e_size
```

Definition at line 383 of file eHSM_If_Evita_Types_Ip.h.

3.133.2.4 rsa_n_size

```
ehsm_uint16_t ehsm_rsa_key_size_::rsa_n_size
```

Definition at line 382 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.134 ehsm_rsa_pubkey Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t e](#) [512]
- [ehsm_uint8_t n](#) [512]

3.134.1 Detailed Description

Definition at line 277 of file eHSM_If_Evita_Types_Ip.h.

3.134.2 Member Data Documentation

3.134.2.1 e

[ehsm_uint8_t](#) ehsm_rsa_pubkey::e[512]

Definition at line 279 of file eHSM_If_Evita_Types_Ip.h.

3.134.2.2 n

[ehsm_uint8_t](#) ehsm_rsa_pubkey::n[512]

Definition at line 280 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.135 ehsm_se_key_ Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint32_t](#) algo_id
- [ehsm_uint8_t](#) auth_size
- [ehsm_uint8_t](#) reserved [3]
- [ehsm_uint8_t](#) auth_value [EHSM_KEY_AUTH_VALUE_MAX_SIZE]
- [ehsm_uint8_t](#) key_size_info [EHSM_KEY_SIZE_INFO_MAX_LEN]
- [ehsm_uint8_t](#) key_data [0]

3.135.1 Detailed Description

Definition at line 396 of file eHSM_If_Evita_Types_Ip.h.

3.135.2 Member Data Documentation

3.135.2.1 algo_id

```
ehsm_uint32_t ehsm_se_key_::algo_id
```

Definition at line 399 of file eHSM_If_Evita_Types_Ip.h.

3.135.2.2 auth_size

```
ehsm_uint8_t ehsm_se_key_::auth_size
```

Definition at line 400 of file eHSM_If_Evita_Types_Ip.h.

3.135.2.3 auth_value

```
ehsm_uint8_t ehsm_se_key_::auth_value[EHSM_KEY_AUTH_VALUE_MAX_SIZE]
```

Definition at line 402 of file eHSM_If_Evita_Types_Ip.h.

3.135.2.4 key_data

```
ehsm_uint8_t ehsm_se_key_::key_data[0]
```

Definition at line 404 of file eHSM_If_Evita_Types_Ip.h.

3.135.2.5 key_handle

```
ehsm_uint32_t ehsm_se_key_::key_handle
```

Definition at line 398 of file eHSM_If_Evita_Types_Ip.h.

3.135.2.6 key_size_info

```
ehsm_uint8_t ehsm_se_key_::key_size_info[EHSM_KEY_SIZE_INFO_MAX_LEN]
```

Definition at line 403 of file eHSM_If_Evita_Types_Ip.h.

3.135.2.7 reserved

```
ehsm_uint8_t ehsm_se_key_::reserved[3]
```

Definition at line 401 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.136 ehsm_secure_boot_st Struct Reference

```
#include <eHSM_If_Ext_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) rev1 [1]
- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) storage_alg
- [ehsm_uint8_t](#) need_encryption
- const [ehsm_uint8_t](#) * pubkey_addr
- [ehsm_uint32_t](#) pubkey_size
- const [ehsm_uint8_t](#) * image_addr
- [ehsm_uint32_t](#) image_size
- const [ehsm_uint8_t](#) * header_addr
- [ehsm_uint32_t](#) header_size
- const [ehsm_uint8_t](#) * encrypt_iv_addr
- [ehsm_uint32_t](#) encrypt_iv_size
- const [ehsm_uint8_t](#) * sign_addr
- [ehsm_uint32_t](#) sign_size

3.136.1 Detailed Description

Definition at line 25 of file eHSM_If_Ext_Ip.h.

3.136.2 Member Data Documentation

3.136.2.1 encrypt_iv_addr

```
const ehsm\_uint8\_t* ehsm_secure_boot_st::encrypt_iv_addr
```

Definition at line 37 of file eHSM_If_Ext_Ip.h.

3.136.2.2 encrypt_iv_size

```
ehsm_uint32_t ehsm_secure_boot_st::encrypt_iv_size
```

Definition at line 38 of file eHSM_If_Ext_Ip.h.

3.136.2.3 header_addr

```
const ehsm_uint8_t* ehsm_secure_boot_st::header_addr
```

Definition at line 35 of file eHSM_If_Ext_Ip.h.

3.136.2.4 header_size

```
ehsm_uint32_t ehsm_secure_boot_st::header_size
```

Definition at line 36 of file eHSM_If_Ext_Ip.h.

3.136.2.5 image_addr

```
const ehsm_uint8_t* ehsm_secure_boot_st::image_addr
```

Definition at line 33 of file eHSM_If_Ext_Ip.h.

3.136.2.6 image_size

```
ehsm_uint32_t ehsm_secure_boot_st::image_size
```

Definition at line 34 of file eHSM_If_Ext_Ip.h.

3.136.2.7 need_encryption

```
ehsm_uint8_t ehsm_secure_boot_st::need_encryption
```

Definition at line 30 of file eHSM_If_Ext_Ip.h.

3.136.2.8 pubkey_addr

```
const ehsm_uint8_t* ehsm_secure_boot_st::pubkey_addr
```

Definition at line 31 of file eHSM_If_Ext_Ip.h.

3.136.2.9 pubkey_size

```
ehsm_uint32_t ehsm_secure_boot_st::pubkey_size
```

Definition at line 32 of file eHSM_If_Ext_Ip.h.

3.136.2.10 rev1

```
ehsm_uint8_t ehsm_secure_boot_st::rev1[1]
```

Definition at line 27 of file eHSM_If_Ext_Ip.h.

3.136.2.11 sign_addr

```
const ehsm_uint8_t* ehsm_secure_boot_st::sign_addr
```

Definition at line 39 of file eHSM_If_Ext_Ip.h.

3.136.2.12 sign_size

```
ehsm_uint32_t ehsm_secure_boot_st::sign_size
```

Definition at line 40 of file eHSM_If_Ext_Ip.h.

3.136.2.13 storage_alg

```
ehsm_uint8_t ehsm_secure_boot_st::storage_alg
```

Definition at line 29 of file eHSM_If_Ext_Ip.h.

3.136.2.14 type

[ehsm_uint8_t](#) ehsm_secure_boot_st::type

Definition at line 28 of file eHSM_If_Ext_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Ext_Ip.h](#)

3.137 ehsm_self_test_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) test_type

3.137.1 Detailed Description

Definition at line 708 of file eHSM_Mailbox_Prtcl_Ip.h.

3.137.2 Member Data Documentation

3.137.2.1 test_type

[ehsm_uint32_t](#) ehsm_self_test_cmd::test_type

Definition at line 709 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.138 ehsm_sensor_init_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) size
- [ehsm_uint8_t](#) * data

3.138.1 Detailed Description

Definition at line 1043 of file eHSM_Com_Struct_Ip.h.

3.138.2 Member Data Documentation

3.138.2.1 data

[ehsm_uint8_t](#)* ehsm_sensor_init_param_st::data

Definition at line 1045 of file eHSM_Com_Struct_Ip.h.

3.138.2.2 size

[ehsm_uint32_t](#) ehsm_sensor_init_param_st::size

Definition at line 1044 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.139 ehsm_sensor_resp_init_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) data_size
- [ehsm_uint8_t](#) data_addr [[HOST_ADDRESS_SIZE](#)]

3.139.1 Detailed Description

Definition at line 601 of file eHSM_Mailbox_Prtcl_Ip.h.

3.139.2 Member Data Documentation

3.139.2.1 data_addr

```
ehsm_uint8_t ehsm_sensor_resp_init_cmd::data_addr[HOST_ADDRESS_SIZE]
```

Definition at line 603 of file eHSM_Mailbox_Prtcl_Ip.h.

3.139.2.2 data_size

```
ehsm_uint32_t ehsm_sensor_resp_init_cmd::data_size
```

Definition at line 602 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.140 ehsm_service Struct Reference

```
#include <eHSM_Srv_Mgr_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) service_id
- [service_reqhdl](#) reqhdl
- [service_rsphdl](#) rsphdl
- [ehsm_uint32_t](#) timeout

3.140.1 Detailed Description

Definition at line 114 of file eHSM_Srv_Mgr_Ip.h.

3.140.2 Member Data Documentation

3.140.2.1 reqhdl

```
service_reqhdl ehsm_service::reqhdl
```

Definition at line 116 of file eHSM_Srv_Mgr_Ip.h.

3.140.2.2 rsphdl

```
service_rsphdl ehsm_service::rsphdl
```

Definition at line 117 of file eHSM_Srv_Mgr_Ip.h.

3.140.2.3 service_id

```
ehsm_uint32_t ehsm_service::service_id
```

Definition at line 115 of file eHSM_Srv_Mgr_Ip.h.

3.140.2.4 timeout

```
ehsm_uint32_t ehsm_service::timeout
```

Definition at line 119 of file eHSM_Srv_Mgr_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Srv_Mgr_Ip.h](#)

3.141 ehsm_service_info Struct Reference

```
#include <eHSM_Srv_Mgr_Ip.h>
```

Public Attributes

- [ehsm_cmd_req_type_e req_type](#)
- [cmd_req_cb cb](#)
- [ehsm_uint32_t priority](#)
- [ehsm_api_type_e api_type](#)
- void * [service_ctx](#)

3.141.1 Detailed Description

Definition at line 122 of file eHSM_Srv_Mgr_Ip.h.

3.141.2 Member Data Documentation

3.141.2.1 api_type

[ehsm_api_type_e](#) ehsm_service_info::api_type

Definition at line 126 of file eHSM_Srv_Mgr_Ip.h.

3.141.2.2 cb

[cmd_req_cb](#) ehsm_service_info::cb

Definition at line 124 of file eHSM_Srv_Mgr_Ip.h.

3.141.2.3 priority

[ehsm_uint32_t](#) ehsm_service_info::priority

Definition at line 125 of file eHSM_Srv_Mgr_Ip.h.

3.141.2.4 req_type

[ehsm_cmd_req_type_e](#) ehsm_service_info::req_type

Definition at line 123 of file eHSM_Srv_Mgr_Ip.h.

3.141.2.5 service_ctx

void* ehsm_service_info::service_ctx

Definition at line 128 of file eHSM_Srv_Mgr_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Srv_Mgr_Ip.h](#)

3.142 ehsm_set_baudrate_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) baud_div

3.142.1 Detailed Description

Definition at line 606 of file eHSM_Mailbox_Prtcl_Ip.h.

3.142.2 Member Data Documentation

3.142.2.1 baud_div

```
ehsm_uint32_t ehsm_set_baudrate_cmd::baud_div
```

Definition at line 607 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.143 ehsm_she_get_id_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * challenge
- [ehsm_uint32_t](#) challenge_size
- [ehsm_uint8_t](#) * status
- [ehsm_uint32_t](#) status_size
- [ehsm_uint8_t](#) * signatrue
- [ehsm_uint32_t](#) signatrue_size

3.143.1 Detailed Description

Definition at line 963 of file eHSM_Com_Struct_Ip.h.

3.143.2 Member Data Documentation

3.143.2.1 challenge

```
ehsm_uint8_t* ehsm_she_get_id_param_st::challenge
```

Definition at line 964 of file eHSM_Com_Struct_Ip.h.

3.143.2.2 challenge_size

`ehsm_uint32_t ehsm_she_get_id_param_st::challenge_size`

Definition at line 965 of file eHSM_Com_Struct_Ip.h.

3.143.2.3 signatrue

`ehsm_uint8_t* ehsm_she_get_id_param_st::signatrue`

Definition at line 968 of file eHSM_Com_Struct_Ip.h.

3.143.2.4 signatrue_size

`ehsm_uint32_t ehsm_she_get_id_param_st::signatrue_size`

Definition at line 969 of file eHSM_Com_Struct_Ip.h.

3.143.2.5 status

`ehsm_uint8_t* ehsm_she_get_id_param_st::status`

Definition at line 966 of file eHSM_Com_Struct_Ip.h.

3.143.2.6 status_size

`ehsm_uint32_t ehsm_she_get_id_param_st::status_size`

Definition at line 967 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.144 ehsm_she_key_host_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```


Public Attributes

- [ehsm_uint8_t](#) * m1
- [ehsm_uint8_t](#) * m2
- [ehsm_uint8_t](#) * m3
- [ehsm_uint8_t](#) * m4
- [ehsm_uint8_t](#) * m5
- [ehsm_bool_t](#) she_ext_flag

3.144.1 Detailed Description

Definition at line 445 of file eHSM_Com_Struct_lp.h.

3.144.2 Member Data Documentation

3.144.2.1 m1

[ehsm_uint8_t](#)* ehsm_she_key_host_param::m1

Definition at line 447 of file eHSM_Com_Struct_lp.h.

3.144.2.2 m2

[ehsm_uint8_t](#)* ehsm_she_key_host_param::m2

Definition at line 448 of file eHSM_Com_Struct_lp.h.

3.144.2.3 m3

[ehsm_uint8_t](#)* ehsm_she_key_host_param::m3

Definition at line 449 of file eHSM_Com_Struct_lp.h.

3.144.2.4 m4

[ehsm_uint8_t](#)* ehsm_she_key_host_param::m4

Definition at line 450 of file eHSM_Com_Struct_lp.h.

3.144.2.5 m5

```
ehsm_uint8_t* ehsm_she_key_host_param::m5
```

Definition at line 451 of file eHSM_Com_Struct_Ip.h.

3.144.2.6 she_ext_flag

```
ehsm_bool_t ehsm_she_key_host_param::she_ext_flag
```

Definition at line 453 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.145 ehsm_she_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) m1 [[EHSM_SHE_M1_MAX_SIZE](#)]
- [ehsm_uint8_t](#) m2 [[EHSM_SHE_M2_MAX_SIZE](#)]
- [ehsm_uint8_t](#) m3 [[EHSM_SHE_M3_MAX_SIZE](#)]
- [ehsm_uint8_t](#) m4 [[EHSM_SHE_M4_MAX_SIZE](#)]
- [ehsm_uint8_t](#) m5 [[EHSM_SHE_M5_MAX_SIZE](#)]
- [ehsm_bool_t](#) she_ext_flag

3.145.1 Detailed Description

Definition at line 433 of file eHSM_Com_Struct_Ip.h.

3.145.2 Member Data Documentation

3.145.2.1 m1

```
ehsm_uint8_t ehsm_she_key_param::m1 [EHSM\_SHE\_M1\_MAX\_SIZE]
```

Definition at line 435 of file eHSM_Com_Struct_Ip.h.

3.145.2.2 m2

```
ehsm_uint8_t ehsm_she_key_param::m2[EHSM_SHE_M2_MAX_SIZE]
```

Definition at line 436 of file eHSM_Com_Struct_Ip.h.

3.145.2.3 m3

```
ehsm_uint8_t ehsm_she_key_param::m3[EHSM_SHE_M3_MAX_SIZE]
```

Definition at line 437 of file eHSM_Com_Struct_Ip.h.

3.145.2.4 m4

```
ehsm_uint8_t ehsm_she_key_param::m4[EHSM_SHE_M4_MAX_SIZE]
```

Definition at line 438 of file eHSM_Com_Struct_Ip.h.

3.145.2.5 m5

```
ehsm_uint8_t ehsm_she_key_param::m5[EHSM_SHE_M5_MAX_SIZE]
```

Definition at line 439 of file eHSM_Com_Struct_Ip.h.

3.145.2.6 she_ext_flag

```
ehsm_bool_t ehsm_she_key_param::she_ext_flag
```

Definition at line 441 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.146 ehsm_she_key_st Struct Reference

```
#include <eHSM_If_She_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) counter
- [ehsm_uint8_t](#) secure_flag
- [ehsm_uint8_t](#) reserved [3]
- [ehsm_uint8_t](#) raw_key [EHSM_SHE_KEY_MAX_SIZE]

3.146.1 Detailed Description

Definition at line 109 of file eHSM_If_She_Types_Ip.h.

3.146.2 Member Data Documentation

3.146.2.1 counter

```
ehsm_uint32_t ehsm_she_key_st::counter
```

Definition at line 111 of file eHSM_If_She_Types_Ip.h.

3.146.2.2 raw_key

```
ehsm_uint8_t ehsm_she_key_st::raw_key [EHSM_SHE_KEY_MAX_SIZE]
```

Definition at line 114 of file eHSM_If_She_Types_Ip.h.

3.146.2.3 reserved

```
ehsm_uint8_t ehsm_she_key_st::reserved [3]
```

Definition at line 113 of file eHSM_If_She_Types_Ip.h.

3.146.2.4 secure_flag

```
ehsm_uint8_t ehsm_she_key_st::secure_flag
```

Definition at line 112 of file eHSM_If_She_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_She_Types_Ip.h](#)

3.147 ehsm_she_load_export_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t m1](#) [[HOST_ADDRESS_SIZE](#)]
- [ehsm_uint8_t m2](#) [[HOST_ADDRESS_SIZE](#)]
- [ehsm_uint8_t m3](#) [[HOST_ADDRESS_SIZE](#)]
- [ehsm_uint8_t m4](#) [[HOST_ADDRESS_SIZE](#)]
- [ehsm_uint8_t m5](#) [[HOST_ADDRESS_SIZE](#)]
- [ehsm_bool_t she_ext_flag](#)

3.147.1 Detailed Description

Definition at line 388 of file eHSM_Mailbox_Prtcl_Ip.h.

3.147.2 Member Data Documentation

3.147.2.1 m1

```
ehsm\_uint8\_t ehsm_she_load_export_key_cmd::m1 [HOST\_ADDRESS\_SIZE]
```

Definition at line 390 of file eHSM_Mailbox_Prtcl_Ip.h.

3.147.2.2 m2

```
ehsm\_uint8\_t ehsm_she_load_export_key_cmd::m2 [HOST\_ADDRESS\_SIZE]
```

Definition at line 391 of file eHSM_Mailbox_Prtcl_Ip.h.

3.147.2.3 m3

```
ehsm\_uint8\_t ehsm_she_load_export_key_cmd::m3 [HOST\_ADDRESS\_SIZE]
```

Definition at line 392 of file eHSM_Mailbox_Prtcl_Ip.h.

3.147.2.4 m4

```
ehsm_uint8_t ehsm_she_load_export_key_cmd::m4[HOST_ADDRESS_SIZE]
```

Definition at line 393 of file eHSM_Mailbox_Prtcl_Ip.h.

3.147.2.5 m5

```
ehsm_uint8_t ehsm_she_load_export_key_cmd::m5[HOST_ADDRESS_SIZE]
```

Definition at line 394 of file eHSM_Mailbox_Prtcl_Ip.h.

3.147.2.6 she_ext_flag

```
ehsm_bool_t ehsm_she_load_export_key_cmd::she_ext_flag
```

Definition at line 395 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.148 ehsm_she_load_plain_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t key_data](#) [HOST_ADDRESS_SIZE]

3.148.1 Detailed Description

Definition at line 398 of file eHSM_Mailbox_Prtcl_Ip.h.

3.148.2 Member Data Documentation

3.148.2.1 key_data

```
ehsm_uint8_t ehsm_she_load_plain_key_cmd::key_data[HOST_ADDRESS_SIZE]
```

Definition at line 400 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.149 ehsm_she_plain_key_host_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * [key_data](#)

3.149.1 Detailed Description

Definition at line 461 of file eHSM_Com_Struct_Ip.h.

3.149.2 Member Data Documentation

3.149.2.1 key_data

```
ehsm_uint8_t* ehsm_she_plain_key_host_param::key_data
```

Definition at line 463 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.150 ehsm_she_plain_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) [key_data](#) [16]

3.150.1 Detailed Description

Definition at line 456 of file eHSM_Com_Struct_lp.h.

3.150.2 Member Data Documentation

3.150.2.1 key_data

```
ehsm_uint8_t ehsm_she_plain_key_param::key_data[16]
```

Definition at line 458 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.151 ehsm_sm9_exchg_gen_usertmp_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) rev1 [7]
- [ehsm_uint8_t](#) rev_key_handle [4]
- [ehsm_uint8_t](#) kgc_public_key [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev_key_auth_size [4]
- [ehsm_uint8_t](#) peer_id [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) peer_id_size [4]

3.151.1 Detailed Description

Definition at line 557 of file eHSM_Mailbox_Prtcl_lp.h.

3.151.2 Member Data Documentation

3.151.2.1 kgc_public_key

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::kgc_public_key[HOST_ADDRESS_SIZE]
```

Definition at line 564 of file eHSM_Mailbox_Prtcl_lp.h.

3.151.2.2 peer_id

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::peer_id[HOST_ADDRESS_SIZE]
```

Definition at line 567 of file eHSM_Mailbox_Prtcl_Ip.h.

3.151.2.3 peer_id_size

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::peer_id_size[4]
```

Definition at line 569 of file eHSM_Mailbox_Prtcl_Ip.h.

3.151.2.4 rev1

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::rev1[7]
```

Definition at line 561 of file eHSM_Mailbox_Prtcl_Ip.h.

3.151.2.5 rev_key_auth_size

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::rev_key_auth_size[4]
```

Definition at line 565 of file eHSM_Mailbox_Prtcl_Ip.h.

3.151.2.6 rev_key_handle

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::rev_key_handle[4]
```

Definition at line 562 of file eHSM_Mailbox_Prtcl_Ip.h.

3.151.2.7 type

```
ehsm_uint8_t ehsm_sm9_exchg_gen_usertmp_cmd::type
```

Definition at line 560 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.152 ehsm_sm9_exchg_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) rev1 [6]
- [ehsm_uint8_t](#) role
- [ehsm_uint8_t](#) user_priv_key_handle [4]
- [ehsm_uint8_t](#) rev2 [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev3 [4]
- [ehsm_uint8_t](#) info_stuct [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) user_tmp_key_handle [4]
- [ehsm_uint8_t](#) rev4 [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) key_size [4]

3.152.1 Detailed Description

Definition at line 443 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2 Member Data Documentation

3.152.2.1 info_stuct

```
ehsm\_uint8\_t ehsm_sm9_exchg_key_cmd::info_stuct [HOST_ADDRESS_SIZE]
```

Definition at line 451 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.2 key_size

```
ehsm\_uint8\_t ehsm_sm9_exchg_key_cmd::key_size [4]
```

Definition at line 454 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.3 rev1

```
ehsm\_uint8\_t ehsm_sm9_exchg_key_cmd::rev1 [6]
```

Definition at line 446 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.4 rev2

`ehsm_uint8_t ehsm_sm9_exchg_key_cmd::rev2[HOST_ADDRESS_SIZE]`

Definition at line 449 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.5 rev3

`ehsm_uint8_t ehsm_sm9_exchg_key_cmd::rev3[4]`

Definition at line 450 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.6 rev4

`ehsm_uint8_t ehsm_sm9_exchg_key_cmd::rev4[HOST_ADDRESS_SIZE]`

Definition at line 453 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.7 role

`ehsm_uint8_t ehsm_sm9_exchg_key_cmd::role`

Definition at line 447 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.8 type

`ehsm_uint8_t ehsm_sm9_exchg_key_cmd::type`

Definition at line 445 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.9 user_priv_key_handle

`ehsm_uint8_t ehsm_sm9_exchg_key_cmd::user_priv_key_handle[4]`

Definition at line 448 of file eHSM_Mailbox_Prtcl_Ip.h.

3.152.2.10 user_tmp_key_handle

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd::user_tmp_key_handle[4]
```

Definition at line 452 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.153 ehsm_sm9_exchg_key_cmd_child Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t peer_tmp_pub](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t kgc_pub_key](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t fp12g](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t self_id](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t peer_id](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t self_id_size](#) [4]
- [ehsm_uint8_t peer_id_size](#) [4]
- [ehsm_uint8_t s1_s2](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t sa_sb](#) [HOST_ADDRESS_SIZE]

3.153.1 Detailed Description

Definition at line 429 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2 Member Data Documentation

3.153.2.1 fp12g

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::fp12g[HOST_ADDRESS_SIZE]
```

Definition at line 433 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.2 kgc_pub_key

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::kgc_pub_key[HOST_ADDRESS_SIZE]
```

Definition at line 432 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.3 peer_id

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::peer_id[HOST_ADDRESS_SIZE]
```

Definition at line 435 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.4 peer_id_size

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::peer_id_size[4]
```

Definition at line 437 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.5 peer_tmp_pub

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::peer_tmp_pub[HOST_ADDRESS_SIZE]
```

Definition at line 431 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.6 s1_s2

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::s1_s2[HOST_ADDRESS_SIZE]
```

Definition at line 438 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.7 sa_sb

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::sa_sb[HOST_ADDRESS_SIZE]
```

Definition at line 439 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.8 self_id

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::self_id[HOST_ADDRESS_SIZE]
```

Definition at line 434 of file eHSM_Mailbox_Prtcl_Ip.h.

3.153.2.9 self_id_size

```
ehsm_uint8_t ehsm_sm9_exchg_key_cmd_child::self_id_size[4]
```

Definition at line 436 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.154 ehsm_sm9_exckey_gen_tmpkey_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_sm9_user_privkey_type_e](#) priv_key_type
- const [ehsm_uint8_t](#) * peer_id
- [ehsm_uint32_t](#) peer_id_size
- [ehsm_key_mem_type_e](#) type
- const [ehsm_uint8_t](#) * kgc_pub_key
- [ehsm_uint32_t](#) key_handle

3.154.1 Detailed Description

Definition at line 884 of file eHSM_Com_Struct_Ip.h.

3.154.2 Member Data Documentation

3.154.2.1 key_handle

```
ehsm_uint32_t ehsm_sm9_exckey_gen_tmpkey_param::key_handle
```

Definition at line 891 of file eHSM_Com_Struct_Ip.h.

3.154.2.2 kgc_pub_key

```
const ehsm_uint8_t* ehsm_sm9_exckey_gen_tmpkey_param::kgc_pub_key
```

Definition at line 890 of file eHSM_Com_Struct_Ip.h.

3.154.2.3 peer_id

```
const ehsm_uint8_t* ehsm_sm9_exckey_gen_tmpkey_param::peer_id
```

Definition at line 887 of file eHSM_Com_Struct_Ip.h.

3.154.2.4 peer_id_size

```
ehsm_uint32_t ehsm_sm9_exckey_gen_tmpkey_param::peer_id_size
```

Definition at line 888 of file eHSM_Com_Struct_Ip.h.

3.154.2.5 priv_key_type

```
ehsm_sm9_user_privkey_type_e ehsm_sm9_exckey_gen_tmpkey_param::priv_key_type
```

Definition at line 886 of file eHSM_Com_Struct_Ip.h.

3.154.2.6 type

```
ehsm_key_mem_type_e ehsm_sm9_exckey_gen_tmpkey_param::type
```

Definition at line 889 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.155 ehsm_sm9_export_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t rev1](#) [8]
- [ehsm_uint8_t key_handle](#) [4]
- [ehsm_uint8_t rev_key_auth](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev_key_auth_size](#) [4]
- [ehsm_uint8_t encrypted_key](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t encrypted_key_size](#) [4]
- [ehsm_uint8_t authenticated_key](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t authenticated_key_size](#) [4]

3.155.1 Detailed Description

Definition at line 514 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2 Member Data Documentation

3.155.2.1 authenticated_key

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::authenticated_key[HOST_ADDRESS_SIZE]
```

Definition at line 522 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.2 authenticated_key_size

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::authenticated_key_size[4]
```

Definition at line 523 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.3 encrypted_key

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::encrypted_key[HOST_ADDRESS_SIZE]
```

Definition at line 520 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.4 encrypted_key_size

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::encrypted_key_size[4]
```

Definition at line 521 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.5 key_handle

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::key_handle[4]
```

Definition at line 517 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.6 rev1

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::rev1[8]
```

Definition at line 516 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.7 rev_key_auth

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::rev_key_auth[HOST_ADDRESS_SIZE]
```

Definition at line 518 of file eHSM_Mailbox_Prtcl_Ip.h.

3.155.2.8 rev_key_auth_size

```
ehsm_uint8_t ehsm_sm9_export_key_cmd::rev_key_auth_size[4]
```

Definition at line 519 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.156 ehsm_sm9_gen_mast_pubkey Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_sm9_master_key_type_e](#) key_type
- [ehsm_uint8_t](#) * pub_key

3.156.1 Detailed Description

Definition at line 906 of file eHSM_Com_Struct_Ip.h.

3.156.2 Member Data Documentation

3.156.2.1 key_type

[ehsm_sm9_master_key_type_e](#) ehsm_sm9_gen_mast_pubkey::key_type

Definition at line 908 of file eHSM_Com_Struct_Ip.h.

3.156.2.2 pub_key

[ehsm_uint8_t*](#) ehsm_sm9_gen_mast_pubkey::pub_key

Definition at line 909 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.157 ehsm_sm9_gen_tmp_pubkey_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- const [ehsm_uint8_t](#) * user_id
- [ehsm_uint32_t](#) id_size
- [ehsm_uint8_t](#) * pub_key

3.157.1 Detailed Description

Definition at line 912 of file eHSM_Com_Struct_Ip.h.

3.157.2 Member Data Documentation

3.157.2.1 id_size

[ehsm_uint32_t](#) ehsm_sm9_gen_tmp_pubkey_param::id_size

Definition at line 916 of file eHSM_Com_Struct_Ip.h.

3.157.2.2 key_handle

[ehsm_uint32_t](#) ehsm_sm9_gen_tmp_pubkey_param::key_handle

Definition at line 914 of file eHSM_Com_Struct_Ip.h.

3.157.2.3 pub_key

[ehsm_uint8_t*](#) ehsm_sm9_gen_tmp_pubkey_param::pub_key

Definition at line 917 of file eHSM_Com_Struct_Ip.h.

3.157.2.4 user_id

const [ehsm_uint8_t*](#) ehsm_sm9_gen_tmp_pubkey_param::user_id

Definition at line 915 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.158 ehsm_sm9_get_mast_pubkey_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) master_key_type
- [ehsm_uint8_t](#) reserved2 [35]
- [ehsm_uint8_t](#) public_key_addr [HOST_ADDRESS_SIZE]

3.158.1 Detailed Description

Definition at line 540 of file eHSM_Mailbox_Prtcl_Ip.h.

3.158.2 Member Data Documentation

3.158.2.1 master_key_type

`ehsm_uint8_t ehsm_sm9_get_mast_pubkey_cmd::master_key_type`

Definition at line 542 of file eHSM_Mailbox_Prtcl_Ip.h.

3.158.2.2 public_key_addr

`ehsm_uint8_t ehsm_sm9_get_mast_pubkey_cmd::public_key_addr[HOST_ADDRESS_SIZE]`

Definition at line 544 of file eHSM_Mailbox_Prtcl_Ip.h.

3.158.2.3 reserved2

`ehsm_uint8_t ehsm_sm9_get_mast_pubkey_cmd::reserved2[35]`

Definition at line 543 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.159 ehsm_sm9_get_tmp_pubkey_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t reserved1](#) [8]
- [ehsm_uint8_t key_handle](#) [4]
- [ehsm_uint8_t reserved2](#) [12]
- [ehsm_uint8_t user_id](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t user_id_size](#) [4]
- [ehsm_uint8_t public_key_addr](#) [HOST_ADDRESS_SIZE]

3.159.1 Detailed Description

Definition at line 547 of file eHSM_Mailbox_Prtcl_Ip.h.

3.159.2 Member Data Documentation

3.159.2.1 key_handle

`ehsm_uint8_t ehsm_sm9_get_tmp_pubkey_cmd::key_handle[4]`

Definition at line 550 of file eHSM_Mailbox_Prtcl_Ip.h.

3.159.2.2 public_key_addr

`ehsm_uint8_t ehsm_sm9_get_tmp_pubkey_cmd::public_key_addr[HOST_ADDRESS_SIZE]`

Definition at line 554 of file eHSM_Mailbox_Prtcl_Ip.h.

3.159.2.3 reserved1

`ehsm_uint8_t ehsm_sm9_get_tmp_pubkey_cmd::reserved1[8]`

Definition at line 549 of file eHSM_Mailbox_Prtcl_Ip.h.

3.159.2.4 reserved2

`ehsm_uint8_t ehsm_sm9_get_tmp_pubkey_cmd::reserved2[12]`

Definition at line 551 of file eHSM_Mailbox_Prtcl_Ip.h.

3.159.2.5 user_id

`ehsm_uint8_t ehsm_sm9_get_tmp_pubkey_cmd::user_id[HOST_ADDRESS_SIZE]`

Definition at line 552 of file eHSM_Mailbox_Prtcl_Ip.h.

3.159.2.6 user_id_size

`ehsm_uint8_t ehsm_sm9_get_tmp_pubkey_cmd::user_id_size[4]`

Definition at line 553 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.160 ehsm_sm9_import_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) key_is_plain
- [ehsm_uint8_t](#) rev1 [6]
- [ehsm_uint8_t](#) rev_key_handle [4]
- [ehsm_uint8_t](#) rev_key_auth [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) rev_key_auth_size [4]
- [ehsm_uint8_t](#) encrypted_key [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) encrypted_key_size [4]
- [ehsm_uint8_t](#) authenticated_key [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t](#) authenticated_key_size [4]

3.160.1 Detailed Description

Definition at line 526 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2 Member Data Documentation

3.160.2.1 authenticated_key

```
ehsm\_uint8\_t ehsm_sm9_import_key_cmd::authenticated_key[HOST_ADDRESS_SIZE]
```

Definition at line 536 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.2 authenticated_key_size

```
ehsm\_uint8\_t ehsm_sm9_import_key_cmd::authenticated_key_size[4]
```

Definition at line 537 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.3 encrypted_key

```
ehsm\_uint8\_t ehsm_sm9_import_key_cmd::encrypted_key[HOST_ADDRESS_SIZE]
```

Definition at line 534 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.4 encrypted_key_size

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::encrypted_key_size[4]
```

Definition at line 535 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.5 key_is_plain

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::key_is_plain
```

Definition at line 529 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.6 rev1

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::rev1[6]
```

Definition at line 530 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.7 rev_key_auth

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::rev_key_auth[HOST_ADDRESS_SIZE]
```

Definition at line 532 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.8 rev_key_auth_size

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::rev_key_auth_size[4]
```

Definition at line 533 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.9 rev_key_handle

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::rev_key_handle[4]
```

Definition at line 531 of file eHSM_Mailbox_Prtcl_Ip.h.

3.160.2.10 type

```
ehsm_uint8_t ehsm_sm9_import_key_cmd::type
```

Definition at line 528 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.161 ehsm_sm9_inexport_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) key_handle
- [ehsm_uint8_t](#) * key_blob
- [ehsm_uint32_t](#) key_blob_size
- [ehsm_uint8_t](#) * key_auth_value
- [ehsm_uint32_t](#) key_auth_size
- [ehsm_uint8_t](#) key_is_plain
- [ehsm_key_mem_type_e](#) type

3.161.1 Detailed Description

Definition at line 895 of file eHSM_Com_Struct_Ip.h.

3.161.2 Member Data Documentation

3.161.2.1 key_auth_size

```
ehsm_uint32_t ehsm_sm9_inexport_key_param::key_auth_size
```

Definition at line 901 of file eHSM_Com_Struct_Ip.h.

3.161.2.2 key_auth_value

```
ehsm_uint8_t* ehsm_sm9_inexport_key_param::key_auth_value
```

Definition at line 900 of file eHSM_Com_Struct_Ip.h.

3.161.2.3 key_blob

`ehsm_uint8_t*` ehsm_sm9_inexport_key_param::key_blob

Definition at line 898 of file eHSM_Com_Struct_Ip.h.

3.161.2.4 key_blob_size

`ehsm_uint32_t` ehsm_sm9_inexport_key_param::key_blob_size

Definition at line 899 of file eHSM_Com_Struct_Ip.h.

3.161.2.5 key_handle

`ehsm_uint32_t` ehsm_sm9_inexport_key_param::key_handle

Definition at line 897 of file eHSM_Com_Struct_Ip.h.

3.161.2.6 key_is_plain

`ehsm_uint8_t` ehsm_sm9_inexport_key_param::key_is_plain

Definition at line 902 of file eHSM_Com_Struct_Ip.h.

3.161.2.7 type

`ehsm_key_mem_type_e` ehsm_sm9_inexport_key_param::type

Definition at line 903 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.162 ehsm_sm9_remove_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t reserved1](#) [8]
- [ehsm_uint8_t key_handle](#) [4]

3.162.1 Detailed Description

Definition at line 572 of file eHSM_Mailbox_Prtcl_Ip.h.

3.162.2 Member Data Documentation

3.162.2.1 key_handle

[ehsm_uint8_t](#) ehsm_sm9_remove_key_cmd::key_handle [4]

Definition at line 575 of file eHSM_Mailbox_Prtcl_Ip.h.

3.162.2.2 reserved1

[ehsm_uint8_t](#) ehsm_sm9_remove_key_cmd::reserved1 [8]

Definition at line 574 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.163 ehsm_sm9_unwrap_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t rev1](#) [8]
- [ehsm_uint8_t user_priv_key_handle](#) [4]
- [ehsm_uint8_t rev_key_auth](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev_key_auth_size](#) [4]
- [ehsm_uint8_t cipher_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t cipher_size](#) [4]
- [ehsm_uint8_t key_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_size](#) [4]
- [ehsm_uint8_t id_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t id_size](#) [4]

3.163.1 Detailed Description

Definition at line 500 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2 Member Data Documentation

3.163.2.1 cipher_addr

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::cipher_addr[HOST_ADDRESS_SIZE]
```

Definition at line 506 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.2 cipher_size

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::cipher_size[4]
```

Definition at line 507 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.3 id_addr

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::id_addr[HOST_ADDRESS_SIZE]
```

Definition at line 510 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.4 id_size

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::id_size[4]
```

Definition at line 511 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.5 key_addr

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::key_addr[HOST_ADDRESS_SIZE]
```

Definition at line 508 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.6 key_size

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::key_size[4]
```

Definition at line 509 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.7 rev1

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::rev1[8]
```

Definition at line 502 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.8 rev_key_auth

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::rev_key_auth[HOST_ADDRESS_SIZE]
```

Definition at line 504 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.9 rev_key_auth_size

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::rev_key_auth_size[4]
```

Definition at line 505 of file eHSM_Mailbox_Prtcl_Ip.h.

3.163.2.10 user_priv_key_handle

```
ehsm_uint8_t ehsm_sm9_unwrap_key_cmd::user_priv_key_handle[4]
```

Definition at line 503 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.164 ehsm_sm9_unwrap_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) user_priv_key_handle
- const [ehsm_uint8_t](#) * cipher_addr
- [ehsm_uint32_t](#) cipher_size
- const [ehsm_uint8_t](#) * id_addr
- [ehsm_uint32_t](#) id_size
- [ehsm_uint8_t](#) * key_addr
- [ehsm_uint32_t](#) key_size

3.164.1 Detailed Description

Definition at line 866 of file eHSM_Com_Struct_lp.h.

3.164.2 Member Data Documentation

3.164.2.1 cipher_addr

```
const ehsm\_uint8\_t* ehsm_sm9_unwrap_key_param::cipher_addr
```

Definition at line 871 of file eHSM_Com_Struct_lp.h.

3.164.2.2 cipher_size

```
ehsm\_uint32\_t ehsm_sm9_unwrap_key_param::cipher_size
```

Definition at line 873 of file eHSM_Com_Struct_lp.h.

3.164.2.3 id_addr

```
const ehsm\_uint8\_t* ehsm_sm9_unwrap_key_param::id_addr
```

Definition at line 875 of file eHSM_Com_Struct_lp.h.

3.164.2.4 id_size

```
ehsm\_uint32\_t ehsm_sm9_unwrap_key_param::id_size
```

Definition at line 877 of file eHSM_Com_Struct_lp.h.

3.164.2.5 key_addr

```
ehsm_uint8_t* ehsm_sm9_unwrap_key_param::key_addr
```

Definition at line 879 of file eHSM_Com_Struct_Ip.h.

3.164.2.6 key_size

```
ehsm_uint32_t ehsm_sm9_unwrap_key_param::key_size
```

Definition at line 881 of file eHSM_Com_Struct_Ip.h.

3.164.2.7 user_priv_key_handle

```
ehsm_uint32_t ehsm_sm9_unwrap_key_param::user_priv_key_handle
```

Definition at line 869 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.165 ehsm_sm9_wrap_key_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t rev1](#) [6]
- [ehsm_uint8_t hid](#)
- [ehsm_uint8_t rev2](#) [1]
- [ehsm_uint8_t rev_key_handle](#) [4]
- [ehsm_uint8_t pub_key](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev_key_auth_size](#) [4]
- [ehsm_uint8_t id_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t id_size](#) [4]
- [ehsm_uint8_t key_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t key_size](#) [4]
- [ehsm_uint8_t rev3](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t rev4](#) [4]
- [ehsm_uint8_t fp12g](#) [HOST_ADDRESS_SIZE]

3.165.1 Detailed Description

Definition at line 482 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2 Member Data Documentation

3.165.2.1 fp12g

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::fp12g[HOST_ADDRESS_SIZE]`

Definition at line 496 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.2 hid

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::hid`

Definition at line 485 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.3 id_addr

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::id_addr[HOST_ADDRESS_SIZE]`

Definition at line 490 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.4 id_size

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::id_size[4]`

Definition at line 491 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.5 key_addr

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::key_addr[HOST_ADDRESS_SIZE]`

Definition at line 492 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.6 key_size

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::key_size[4]`

Definition at line 493 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.7 pub_key

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::pub_key[HOST_ADDRESS_SIZE]`

Definition at line 488 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.8 rev1

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::rev1[6]`

Definition at line 484 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.9 rev2

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::rev2[1]`

Definition at line 486 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.10 rev3

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::rev3[HOST_ADDRESS_SIZE]`

Definition at line 494 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.11 rev4

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::rev4[4]`

Definition at line 495 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.12 rev_key_auth_size

`ehsm_uint8_t ehsm_sm9_wrap_key_cmd::rev_key_auth_size[4]`

Definition at line 489 of file eHSM_Mailbox_Prtcl_Ip.h.

3.165.2.13 rev_key_handle

```
ehsm_uint8_t ehsm_sm9_wrap_key_cmd::rev_key_handle[4]
```

Definition at line 487 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.166 ehsm_sm9_wrap_key_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- const [ehsm_uint8_t](#) * user_id
- [ehsm_uint32_t](#) id_size
- const [ehsm_uint8_t](#) * fp12g
- const [ehsm_uint8_t](#) * pub_key
- [ehsm_uint8_t](#) * key_addr
- [ehsm_uint32_t](#) key_size
- [ehsm_uint8_t](#) hid

3.166.1 Detailed Description

Definition at line 852 of file eHSM_Com_Struct_Ip.h.

3.166.2 Member Data Documentation

3.166.2.1 fp12g

```
const ehsm\_uint8\_t* ehsm_sm9_wrap_key_param::fp12g
```

Definition at line 856 of file eHSM_Com_Struct_Ip.h.

3.166.2.2 hid

```
ehsm\_uint8\_t ehsm_sm9_wrap_key_param::hid
```

Definition at line 863 of file eHSM_Com_Struct_Ip.h.

3.166.2.3 id_size

```
ehsm_uint32_t ehsm_sm9_wrap_key_param::id_size
```

Definition at line 855 of file eHSM_Com_Struct_Ip.h.

3.166.2.4 key_addr

```
ehsm_uint8_t* ehsm_sm9_wrap_key_param::key_addr
```

Definition at line 860 of file eHSM_Com_Struct_Ip.h.

3.166.2.5 key_size

```
ehsm_uint32_t ehsm_sm9_wrap_key_param::key_size
```

Definition at line 862 of file eHSM_Com_Struct_Ip.h.

3.166.2.6 pub_key

```
const ehsm_uint8_t* ehsm_sm9_wrap_key_param::pub_key
```

Definition at line 857 of file eHSM_Com_Struct_Ip.h.

3.166.2.7 user_id

```
const ehsm_uint8_t* ehsm_sm9_wrap_key_param::user_id
```

Definition at line 854 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.167 ehsm_soc_image_verify_cmd Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t resered1](#) [1]
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t storage_alg](#)
- [ehsm_uint8_t storage_encryption_flag](#)
- [ehsm_uint8_t resered2](#) [4]
- [ehsm_uint8_t resered3](#) [4]
- [ehsm_uint8_t pubkey_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t pubkey_size](#) [4]
- [ehsm_uint8_t storage_image_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t storage_image_size](#) [4]
- [ehsm_uint8_t header_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t header_size](#) [4]
- [ehsm_uint8_t storage_iv_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t storage_iv_size](#) [4]
- [ehsm_uint8_t storage_sign_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t storage_sign_size](#) [4]

3.167.1 Detailed Description

Definition at line 230 of file eHSM_Mailbox_Prtcl_lp.h.

3.167.2 Member Data Documentation

3.167.2.1 header_addr

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::header_addr [HOST_ADDRESS_SIZE]
```

Definition at line 242 of file eHSM_Mailbox_Prtcl_lp.h.

3.167.2.2 header_size

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::header_size [4]
```

Definition at line 243 of file eHSM_Mailbox_Prtcl_lp.h.

3.167.2.3 pubkey_addr

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::pubkey_addr [HOST_ADDRESS_SIZE]
```

Definition at line 238 of file eHSM_Mailbox_Prtcl_lp.h.

3.167.2.4 pubkey_size

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::pubkey_size[4]
```

Definition at line 239 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.5 resered1

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::resered1[1]
```

Definition at line 232 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.6 resered2

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::resered2[4]
```

Definition at line 236 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.7 resered3

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::resered3[4]
```

Definition at line 237 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.8 storage_alg

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_alg
```

Definition at line 234 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.9 storage_encryption_flag

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_encryption_flag
```

Definition at line 235 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.10 storage_image_addr

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_image_addr[HOST_ADDRESS_SIZE]
```

Definition at line 240 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.11 storage_image_size

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_image_size[4]
```

Definition at line 241 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.12 storage_iv_addr

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_iv_addr[HOST_ADDRESS_SIZE]
```

Definition at line 244 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.13 storage_iv_size

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_iv_size[4]
```

Definition at line 245 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.14 storage_sign_addr

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_sign_addr[HOST_ADDRESS_SIZE]
```

Definition at line 246 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.15 storage_sign_size

```
ehsm_uint8_t ehsm_soc_image_verify_cmd::storage_sign_size[4]
```

Definition at line 247 of file eHSM_Mailbox_Prtcl_Ip.h.

3.167.2.16 type

`ehsm_uint8_t ehsm_soc_image_verify_cmd::type`

Definition at line 233 of file `eHSM_Mailbox_Prtcl_Ip.h`.

The documentation for this struct was generated from the following file:

- `eHSM_Mailbox_Prtcl_Ip.h`

3.168 ehsm_soc_image_verify_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- `ehsm_uint8_t rev1 [1]`
- `ehsm_uint8_t type`
- `ehsm_uint8_t storage_alg`
- `ehsm_uint8_t need_encryption`
- `ehsm_uint8_t * pubkey_addr`
- `ehsm_uint32_t pubkey_size`
- `ehsm_uint8_t * image_addr`
- `ehsm_uint32_t image_size`
- `ehsm_uint8_t * header_addr`
- `ehsm_uint32_t header_size`
- `ehsm_uint8_t * encrypt_iv_addr`
- `ehsm_uint32_t encrypt_iv_size`
- `ehsm_uint8_t * sign_addr`
- `ehsm_uint32_t sign_size`

3.168.1 Detailed Description

Definition at line 653 of file `eHSM_Com_Struct_Ip.h`.

3.168.2 Member Data Documentation

3.168.2.1 encrypt_iv_addr

`ehsm_uint8_t* ehsm_soc_image_verify_st::encrypt_iv_addr`

Definition at line 665 of file `eHSM_Com_Struct_Ip.h`.

3.168.2.2 encrypt_iv_size

`ehsm_uint32_t ehsm_soc_image_verify_st::encrypt_iv_size`

Definition at line 666 of file eHSM_Com_Struct_lp.h.

3.168.2.3 header_addr

`ehsm_uint8_t* ehsm_soc_image_verify_st::header_addr`

Definition at line 663 of file eHSM_Com_Struct_lp.h.

3.168.2.4 header_size

`ehsm_uint32_t ehsm_soc_image_verify_st::header_size`

Definition at line 664 of file eHSM_Com_Struct_lp.h.

3.168.2.5 image_addr

`ehsm_uint8_t* ehsm_soc_image_verify_st::image_addr`

Definition at line 661 of file eHSM_Com_Struct_lp.h.

3.168.2.6 image_size

`ehsm_uint32_t ehsm_soc_image_verify_st::image_size`

Definition at line 662 of file eHSM_Com_Struct_lp.h.

3.168.2.7 need_encryption

`ehsm_uint8_t ehsm_soc_image_verify_st::need_encryption`

Definition at line 658 of file eHSM_Com_Struct_lp.h.

3.168.2.8 pubkey_addr

```
ehsm_uint8_t* ehsm_soc_image_verify_st::pubkey_addr
```

Definition at line 659 of file eHSM_Com_Struct_lp.h.

3.168.2.9 pubkey_size

```
ehsm_uint32_t ehsm_soc_image_verify_st::pubkey_size
```

Definition at line 660 of file eHSM_Com_Struct_lp.h.

3.168.2.10 rev1

```
ehsm_uint8_t ehsm_soc_image_verify_st::rev1[1]
```

Definition at line 655 of file eHSM_Com_Struct_lp.h.

3.168.2.11 sign_addr

```
ehsm_uint8_t* ehsm_soc_image_verify_st::sign_addr
```

Definition at line 667 of file eHSM_Com_Struct_lp.h.

3.168.2.12 sign_size

```
ehsm_uint32_t ehsm_soc_image_verify_st::sign_size
```

Definition at line 668 of file eHSM_Com_Struct_lp.h.

3.168.2.13 storage_alg

```
ehsm_uint8_t ehsm_soc_image_verify_st::storage_alg
```

Definition at line 657 of file eHSM_Com_Struct_lp.h.

3.168.2.14 type

`ehsm_uint8_t ehsm_soc_image_verify_st::type`

Definition at line 656 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.169 ehsm_soc_secure_boot_status_st Struct Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_uint8_t status](#)

3.169.1 Detailed Description

Definition at line 645 of file eHSM_Mailbox_Prtcl_Ip.h.

3.169.2 Member Data Documentation

3.169.2.1 status

`ehsm_uint8_t ehsm_soc_secure_boot_status_st::status`

Definition at line 647 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.170 ehsm_storage_area_param_st Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint32_t addr](#)
- [ehsm_uint32_t size](#)

3.170.1 Detailed Description

Definition at line 933 of file eHSM_Com_Struct_lp.h.

3.170.2 Member Data Documentation

3.170.2.1 addr

```
ehsm_uint32_t ehsm_storage_area_param_st::addr
```

Definition at line 934 of file eHSM_Com_Struct_lp.h.

3.170.2.2 size

```
ehsm_uint32_t ehsm_storage_area_param_st::size
```

Definition at line 935 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.171 ehsm_sym_key_size_ Struct Reference

```
#include <eHSM_If_Evita_Types_lp.h>
```

Public Attributes

- [ehsm_uint16_t key_size](#)
- [ehsm_uint16_t reserved](#) [2]

3.171.1 Detailed Description

Definition at line 368 of file eHSM_If_Evita_Types_lp.h.

3.171.2 Member Data Documentation

3.171.2.1 key_size

```
ehsm_uint16_t ehsm_sym_key_size_::key_size
```

Definition at line 370 of file eHSM_If_Evita_Types_Ip.h.

3.171.2.2 reserved

```
ehsm_uint16_t ehsm_sym_key_size_::reserved[2]
```

Definition at line 371 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.172 ehsm_tick_value Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) current_ticks
- [ehsm_uint32_t](#) tick_length
- [ehsm_uint32_t](#) tick_accuracy

3.172.1 Detailed Description

Definition at line 449 of file eHSM_If_Evita_Types_Ip.h.

3.172.2 Member Data Documentation

3.172.2.1 current_ticks

```
ehsm_uint32_t ehsm_tick_value::current_ticks
```

Definition at line 450 of file eHSM_If_Evita_Types_Ip.h.

3.172.2.2 tick_accuracy

```
ehsm_uint32_t ehsm_tick_value::tick_accuracy
```

Definition at line 452 of file eHSM_If_Evita_Types_Ip.h.

3.172.2.3 tick_length

```
ehsm_uint32_t ehsm_tick_value::tick_length
```

Definition at line 451 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.173 ehsm_uart_cmd Struct Reference

Struct for command of getting challenge, the command is from uart or JTAG channel of mailbox.

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- void * [uart_cmd_buffer](#)

3.173.1 Detailed Description

Struct for command of getting challenge, the command is from uart or JTAG channel of mailbox.

Definition at line 461 of file eHSM_Mailbox_Prtcl_Ip.h.

3.173.2 Member Data Documentation

3.173.2.1 uart_cmd_buffer

```
void* ehsm_uart_cmd::uart_cmd_buffer
```

Definition at line 462 of file eHSM_Mailbox_Prtcl_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.174 hash_hmac_t Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t hash_hmac](#) [EVITA_HASH_BUF_SIZE]
- [ehsm_uint32_t hash_hmac_size](#)
- [ehsm_utc_time_t utc_time](#)

3.174.1 Detailed Description

Definition at line 425 of file eHSM_If_Evita_Types_Ip.h.

3.174.2 Member Data Documentation

3.174.2.1 hash_hmac

```
ehsm_uint8_t hash_hmac_t::hash_hmac [EVITA_HASH_BUF_SIZE]
```

Definition at line 426 of file eHSM_If_Evita_Types_Ip.h.

3.174.2.2 hash_hmac_size

```
ehsm_uint32_t hash_hmac_t::hash_hmac_size
```

Definition at line 427 of file eHSM_If_Evita_Types_Ip.h.

3.174.2.3 utc_time

```
ehsm_utc_time_t hash_hmac_t::utc_time
```

Definition at line 428 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.175 hash_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- const char * [key](#)
- const char * [plaintext](#)
- const char * [digest](#)
- unsigned int [psize](#)
- unsigned short [ksize](#)
- int [setkey_error](#)
- int [digest_error](#)
- unsigned short [tap](#) [[MAX_TAP](#)]
- unsigned short [np](#)
- const char * [iv](#)
- unsigned int [iv_len](#)

3.175.1 Detailed Description

Definition at line 26 of file `utest_vecs_st.h`.

3.175.2 Member Data Documentation

3.175.2.1 `digest`

```
const char* hash_testvec::digest
```

Definition at line 29 of file `utest_vecs_st.h`.

3.175.2.2 `digest_error`

```
int hash_testvec::digest_error
```

Definition at line 33 of file `utest_vecs_st.h`.

3.175.2.3 `iv`

```
const char* hash_testvec::iv
```

Definition at line 36 of file `utest_vecs_st.h`.

3.175.2.4 iv_len

```
unsigned int hash_testvec::iv_len
```

Definition at line 37 of file utest_vecs_st.h.

3.175.2.5 key

```
const char* hash_testvec::key
```

Definition at line 27 of file utest_vecs_st.h.

3.175.2.6 ksize

```
unsigned short hash_testvec::ksize
```

Definition at line 31 of file utest_vecs_st.h.

3.175.2.7 np

```
unsigned short hash_testvec::np
```

Definition at line 35 of file utest_vecs_st.h.

3.175.2.8 plaintext

```
const char* hash_testvec::plaintext
```

Definition at line 28 of file utest_vecs_st.h.

3.175.2.9 psize

```
unsigned int hash_testvec::psize
```

Definition at line 30 of file utest_vecs_st.h.

3.175.2.10 setkey_error

```
int hash_testvec::setkey_error
```

Definition at line 32 of file `utest_vecs_st.h`.

3.175.2.11 tap

```
unsigned short hash_testvec::tap[MAX\_TAP]
```

Definition at line 34 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.176 HSM_AsymCfgType Struct Reference

Specifies the asymmetric algo config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_KeyId](#) KeyId
- [uint8](#) Sync
- [HSM_PaddingType](#) Padding
- [HSM_SignDirection](#) SignDir
- [HSM_CipherDirection](#) CipherDir
- [HSM_HashAlgoType](#) HAlgo
- [HSM_AsymAlgoType](#) AsymAlgo

3.176.1 Detailed Description

Specifies the asymmetric algo config struct.

Definition at line 540 of file `Hsm_Hal.h`.

3.176.2 Member Data Documentation

3.176.2.1 AsymAlgo

`HSM_AsymAlgoType` `HSM_AsymCfgType::AsymAlgo`

symmetric algo type, eg.ecc/rsa/ecc

Definition at line 548 of file Hsm_Hal.h.

3.176.2.2 CipherDir

`HSM_CipherDirection` `HSM_AsymCfgType::CipherDir`

request opeartion, eg.decrypt/encrypt

Definition at line 546 of file Hsm_Hal.h.

3.176.2.3 HAlgo

`HSM_HashAlgoType` `HSM_AsymCfgType::HAlgo`

when asym algo mode, eg.SHA1.. for rsa/ecc

Definition at line 547 of file Hsm_Hal.h.

3.176.2.4 KeyId

`HSM_KeyId` `HSM_AsymCfgType::KeyId`

Key Index

Definition at line 542 of file Hsm_Hal.h.

3.176.2.5 Padding

`HSM_PaddingType` `HSM_AsymCfgType::Padding`

data padding

Definition at line 544 of file Hsm_Hal.h.

3.176.2.6 SignDir

[HSM_SignDirection](#) HSM_AsymCfgType::SignDir

request operation, eg.decrypt/encrypt

Definition at line 545 of file Hsm_Hal.h.

3.176.2.7 Sync

uint8 HSM_AsymCfgType::Sync

0:sync 1:async

Definition at line 543 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.177 HSM_BootCfgType Struct Reference

Specifies the secureboot information.

```
#include <Hsm_Hal.h>
```

Public Attributes

- __IOM uint8 [InitLSram](#)
- __IOM uint8 [InitUSram](#)
- __IOM uint8 [SwitchClock](#)
- __IOM uint8 [LogEnable](#)
- __IOM uint8 [StbWaitHSM](#)
- __IOM uint8 [StbWaitVerify](#)
- __IOM uint16 [WdgTimeout](#)
- __IOM uint32 [BootAddr](#)
- __IOM uint32 [VerifySize](#)
- __IOM uint32 [SignAddr](#)
- __IOM uint32 [PubKeyAddr](#)
- __IOM uint32 [HeaderAddr](#)
- __IOM uint32 [HeaderSize](#)
- __IOM uint8 [VersionUpdateEn](#)
- __IOM uint8 [ResetDisable](#)
- __IOM uint8 [PllFreq](#)
- __IOM uint8 [Reserve1](#)
- __IOM uint32 [VersionAddr](#)

3.177.1 Detailed Description

Specifies the secureboot information.

Definition at line 247 of file Hsm_Hal.h.

3.177.2 Member Data Documentation

3.177.2.1 BootAddr

```
__IOM uint32 HSM_BootCfgType::BootAddr
```

Definition at line 256 of file Hsm_Hal.h.

3.177.2.2 HeaderAddr

```
__IOM uint32 HSM_BootCfgType::HeaderAddr
```

Definition at line 260 of file Hsm_Hal.h.

3.177.2.3 HeaderSize

```
__IOM uint32 HSM_BootCfgType::HeaderSize
```

Definition at line 261 of file Hsm_Hal.h.

3.177.2.4 InitLSram

```
__IOM uint8 HSM_BootCfgType::InitLSram
```

init sram enable

Definition at line 249 of file Hsm_Hal.h.

3.177.2.5 InitUSram

```
__IOM uint8 HSM_BootCfgType::InitUSram
```

Definition at line 250 of file Hsm_Hal.h.

3.177.2.6 LogEnable

```
__IOM uint8 HSM_BootCfgType::LogEnable
```

Definition at line 252 of file Hsm_Hal.h.

3.177.2.7 PllFreq

```
__IOM uint8 HSM_BootCfgType::PllFreq
```

Definition at line 264 of file Hsm_Hal.h.

3.177.2.8 PubKeyAddr

```
__IOM uint32 HSM_BootCfgType::PubKeyAddr
```

Definition at line 259 of file Hsm_Hal.h.

3.177.2.9 Reserve1

```
__IOM uint8 HSM_BootCfgType::Reserve1
```

Definition at line 265 of file Hsm_Hal.h.

3.177.2.10 ResetDisable

```
__IOM uint8 HSM_BootCfgType::ResetDisable
```

Definition at line 263 of file Hsm_Hal.h.

3.177.2.11 SignAddr

```
__IOM uint32 HSM_BootCfgType::SignAddr
```

Definition at line 258 of file Hsm_Hal.h.

3.177.2.12 StbWaitHSM

```
__IOM uint8 HSM_BootCfgType::StbWaitHSM
```

Definition at line 253 of file Hsm_Hal.h.

3.177.2.13 StbWaitVerify

```
__IOM uint8 HSM_BootCfgType::StbWaitVerify
```

Definition at line 254 of file Hsm_Hal.h.

3.177.2.14 SwitchClock

```
__IOM uint8 HSM_BootCfgType::SwitchClock
```

Definition at line 251 of file Hsm_Hal.h.

3.177.2.15 VerifySize

```
__IOM uint32 HSM_BootCfgType::VerifySize
```

Definition at line 257 of file Hsm_Hal.h.

3.177.2.16 VersionAddr

```
__IOM uint32 HSM_BootCfgType::VersionAddr
```

Definition at line 266 of file Hsm_Hal.h.

3.177.2.17 VersionUpdateEn

```
__IOM uint8 HSM_BootCfgType::VersionUpdateEn
```

Definition at line 262 of file Hsm_Hal.h.

3.177.2.18 WdgTimeout

```
__IOM uint16 HSM_BootCfgType::WdgTimeout
```

Definition at line 255 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.178 HSM_CMacCfgType Struct Reference

Specifies the Cipher Mac algo config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint32 [KeyId](#)
- uint8 [Sync](#)
- [HSM_MacDirection](#) MacDir
- [HSM_SymAlgoType](#) SymAlgo

3.178.1 Detailed Description

Specifies the Cipher Mac algo config struct.

Definition at line 580 of file Hsm_Hal.h.

3.178.2 Member Data Documentation

3.178.2.1 KeyId

```
uint32 HSM_CMacCfgType::KeyId
```

Key Index

Definition at line 582 of file Hsm_Hal.h.

3.178.2.2 MacDir

```
HSM\_MacDirection HSM_CMacCfgType::MacDir
```

request operation, eg.gen/ver

Definition at line 584 of file Hsm_Hal.h.

3.178.2.3 SymAlgo

[HSM_SymAlgoType](#) HSM_CMacCfgType::SymAlgo

symmetric algo type, eg.AES_128/AES_256

Definition at line 585 of file Hsm_Hal.h.

3.178.2.4 Sync

uint8 HSM_CMacCfgType::Sync

0:sync 1:async

Definition at line 583 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.179 HSM_DebugAuthConfigType Struct Reference

the deug auth config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_ChallengeType](#) Type
- uint32 [SignatureSize](#)
- uint8 * [Signature](#)
- [HSM_DebugAuthAlgoType](#) Alg
- uint32 [PubKeySize](#)
- uint8 * [PubKey](#)

3.179.1 Detailed Description

the deug auth config struct.

Definition at line 887 of file Hsm_Hal.h.

3.179.2 Member Data Documentation

3.179.2.1 Alg

[HSM_DebugAuthAlgoType](#) HSM_DebugAuthConfigType::Alg

Definition at line 892 of file Hsm_Hal.h.

3.179.2.2 PubKey

uint8* HSM_DebugAuthConfigType::PubKey

Definition at line 894 of file Hsm_Hal.h.

3.179.2.3 PubKeySize

uint32 HSM_DebugAuthConfigType::PubKeySize

Definition at line 893 of file Hsm_Hal.h.

3.179.2.4 Signature

uint8* HSM_DebugAuthConfigType::Signature

Definition at line 891 of file Hsm_Hal.h.

3.179.2.5 SignatureSize

uint32 HSM_DebugAuthConfigType::SignatureSize

Definition at line 890 of file Hsm_Hal.h.

3.179.2.6 Type

[HSM_ChallengeType](#) HSM_DebugAuthConfigType::Type

Definition at line 889 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.180 HSM_DeriveKeyCfgType Struct Reference

the derive Key config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_KdfType](#) Kdf
- uint32 [KeySize](#)
- uint32 [ValidUntil](#)
- [HSM_KeyStorageType](#) KeyType
- uint8 [KeyUsageSize](#)
- [HSM_KeyFlagsElementType](#) * [KeyUsage](#)
- uint32 [SaltDataSize](#)
- uint8 * [SaltData](#)
- [HSM_KeyId](#) KeyId

3.180.1 Detailed Description

the derive Key config struct.

Definition at line 786 of file Hsm_Hal.h.

3.180.2 Member Data Documentation

3.180.2.1 Kdf

[HSM_KdfType](#) HSM_DeriveKeyCfgType::Kdf

Definition at line 788 of file Hsm_Hal.h.

3.180.2.2 KeyId

[HSM_KeyId](#) HSM_DeriveKeyCfgType::KeyId

key id

Definition at line 800 of file Hsm_Hal.h.

3.180.2.3 KeySize

```
uint32 HSM_DeriveKeyCfgType::KeySize
```

Definition at line 789 of file Hsm_Hal.h.

3.180.2.4 KeyType

```
HSM_KeyStorageType HSM_DeriveKeyCfgType::KeyType
```

Definition at line 791 of file Hsm_Hal.h.

3.180.2.5 KeyUsage

```
HSM_KeyFlagsElementType* HSM_DeriveKeyCfgType::KeyUsage
```

Definition at line 794 of file Hsm_Hal.h.

3.180.2.6 KeyUsageSize

```
uint8 HSM_DeriveKeyCfgType::KeyUsageSize
```

Definition at line 792 of file Hsm_Hal.h.

3.180.2.7 SaltData

```
uint8* HSM_DeriveKeyCfgType::SaltData
```

Definition at line 798 of file Hsm_Hal.h.

3.180.2.8 SaltDataSize

```
uint32 HSM_DeriveKeyCfgType::SaltDataSize
```

Definition at line 796 of file Hsm_Hal.h.

3.180.2.9 ValidUntil

```
uint32 HSM_DeriveKeyCfgType::ValidUntil
```

key valid time when hsm time support open be used

Definition at line 790 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.181 HSM_DhParamType Struct Reference

the Dh algo param.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [p](#) [512]
- uint8 [q](#) [512]
- uint8 [g](#) [512]

3.181.1 Detailed Description

the Dh algo param.

Definition at line 708 of file Hsm_Hal.h.

3.181.2 Member Data Documentation

3.181.2.1 g

```
uint8 HSM_DhParamType::g[512]
```

Definition at line 712 of file Hsm_Hal.h.

3.181.2.2 p

```
uint8 HSM_DhParamType::p[512]
```

Definition at line 710 of file Hsm_Hal.h.

3.181.2.3 q

```
uint8 HSM_DhParamType::q[512]
```

Definition at line 711 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.182 HSM_DhPriKeyType Struct Reference

the Dh private key struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [Priv](#) [512]

3.182.1 Detailed Description

the Dh private key struct.

Definition at line 737 of file Hsm_Hal.h.

3.182.2 Member Data Documentation

3.182.2.1 Priv

```
uint8 HSM_DhPriKeyType::Priv[512]
```

Definition at line 739 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.183 Hsm_DhPubKeyType_ Struct Reference

the Dh pubkey struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [pub](#) [512]
- [HSM_DhParamType](#) DhParam

3.183.1 Detailed Description

the Dh pubkey struct.

Definition at line 718 of file Hsm_Hal.h.

3.183.2 Member Data Documentation

3.183.2.1 DhParam

[HSM_DhParamType](#) Hsm_DhPubKeyType_::DhParam

Definition at line 721 of file Hsm_Hal.h.

3.183.2.2 pub

uint8 Hsm_DhPubKeyType_::pub[512]

Definition at line 720 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.184 HSM_EccPubKeyType Struct Reference

the ecc pubkey struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [p](#) [132]

3.184.1 Detailed Description

the ecc pubkey struct.

Definition at line 691 of file Hsm_Hal.h.

3.184.2 Member Data Documentation

3.184.2.1 p

uint8 HSM_EccPubKeyType::p[132]

ecc pubkey, eg.SECP256/384

Definition at line 693 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.185 HSM_FlashKeyPageType Struct Reference

Specifies the flash Key content,size is DFLASH_PAGE_SIZE.

Public Attributes

- [HSM_FlashKeyType KeyInfo](#) [HSM_FLASH_KEY_MAX_SLOT_NUM]
- uint8 [PageReverse](#) [HSM_FLASH_PAGE_REVERSE_LEN]
- uint8 [PageValid](#) [HSM_FLASH_PAGE_VALID_LEN]

3.185.1 Detailed Description

Specifies the flash Key content,size is DFLASH_PAGE_SIZE.

Definition at line 87 of file Hsm_Hal.c.

3.185.2 Member Data Documentation

3.185.2.1 KeyInfo

[HSM_FlashKeyType](#) HSM_FlashKeyPageType::KeyInfo [HSM_FLASH_KEY_MAX_SLOT_NUM]

Definition at line 89 of file Hsm_Hal.c.

3.185.2.2 PageReverse

```
uint8 HSM_FlashKeyPageType::PageReverse[HSM_FLASH_PAGE_REVERSE_LEN]
```

Definition at line 90 of file Hsm_Hal.c.

3.185.2.3 PageValid

```
uint8 HSM_FlashKeyPageType::PageValid[HSM_FLASH_PAGE_VALID_LEN]
```

page valid flag: 0xa5 means valid, other means invalid

Definition at line 91 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.186 HSM_FlashKeyType Struct Reference

Specifies the Key info content,size is 56Bytes.

Public Attributes

- [HSM_KeySlotInfoType SlotInfo](#)
- [HSM_KeyIndexInfoType IndexInfo](#)
- [HSM_KeyHandleInfoType HandleInfo](#)

3.186.1 Detailed Description

Specifies the Key info content,size is 56Bytes.

Definition at line 77 of file Hsm_Hal.c.

3.186.2 Member Data Documentation

3.186.2.1 HandleInfo

```
HSM\_KeyHandleInfoType HSM_FlashKeyType::HandleInfo
```

Definition at line 81 of file Hsm_Hal.c.

3.186.2.2 IndexInfo

[HSM_KeyIndexInfoType](#) [HSM_FlashKeyType::IndexInfo](#)

Definition at line 80 of file Hsm_Hal.c.

3.186.2.3 SlotInfo

[HSM_KeySlotInfoType](#) [HSM_FlashKeyType::SlotInfo](#)

Definition at line 79 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.187 HSM_GenKeyCfgType Struct Reference

the generate random Key config struct by random generator in hsm.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_GenKeyAlgo](#) KeyAlgo
- uint32 [KeySize](#)
- uint32 [ValidUntil](#)
- [HSM_KeyStorageType](#) KeyType
- uint8 [KeyUsageSize](#)
- [HSM_KeyFlagsElementType](#) * [KeyUsage](#)
- [HSM_KeyId](#) KeyId

3.187.1 Detailed Description

the generate random Key config struct by random generator in hsm.

Definition at line 806 of file Hsm_Hal.h.

3.187.2 Member Data Documentation

3.187.2.1 KeyAlgo

[HSM_GenKeyAlgo](#) [HSM_GenKeyCfgType::KeyAlgo](#)

Definition at line 808 of file Hsm_Hal.h.

3.187.2.2 KeyId

[HSM_KeyId](#) HSM_GenKeyCfgType::KeyId

Definition at line 814 of file Hsm_Hal.h.

3.187.2.3 KeySize

uint32 HSM_GenKeyCfgType::KeySize

Definition at line 809 of file Hsm_Hal.h.

3.187.2.4 KeyType

[HSM_KeyStorageType](#) HSM_GenKeyCfgType::KeyType

Definition at line 811 of file Hsm_Hal.h.

3.187.2.5 KeyUsage

[HSM_KeyFlagsElementType*](#) HSM_GenKeyCfgType::KeyUsage

Definition at line 813 of file Hsm_Hal.h.

3.187.2.6 KeyUsageSize

uint8 HSM_GenKeyCfgType::KeyUsageSize

Definition at line 812 of file Hsm_Hal.h.

3.187.2.7 ValidUntil

uint32 HSM_GenKeyCfgType::ValidUntil

Definition at line 810 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.188 HSM_HMacCfgType Struct Reference

Specifies the Hash Mac algo config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint32 [KeyId](#)
- uint8 [Sync](#)
- [HSM_MacDirection](#) MacDir
- [HSM_HashAlgoType](#) HAlgo

3.188.1 Detailed Description

Specifies the Hash Mac algo config struct.

Definition at line 569 of file Hsm_Hal.h.

3.188.2 Member Data Documentation

3.188.2.1 HAlgo

[HSM_HashAlgoType](#) HSM_HMacCfgType::HAlgo

when asym algo mode, eg.SHA1.. for rsa/ecc

Definition at line 574 of file Hsm_Hal.h.

3.188.2.2 KeyId

uint32 HSM_HMacCfgType::KeyId

Key Index

Definition at line 571 of file Hsm_Hal.h.

3.188.2.3 MacDir

[HSM_MacDirection](#) HSM_HMacCfgType::MacDir

request opeartion, eg.gen/ver

Definition at line 573 of file Hsm_Hal.h.

3.188.2.4 Sync

```
uint8 HSM_HMacCfgType::Sync
```

0:sync 1:async

Definition at line 572 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.189 HSM_ImageVerifyType__ Struct Reference

Host image verify information.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [UpdateVersionFlag](#)
- uint8 [Type](#)
- uint8 [StorageAlg](#)
- uint8 [StorageEncryptionFlag](#)
- uint8 [VersionAddr](#) [4]
- uint8 [VersionSize](#) [4]
- uint8 [PubkeyAddr](#) [4]
- uint8 [PubkeySize](#) [4]
- uint8 [StorageImageAddr](#) [4]
- uint8 [StorageImageSize](#) [4]
- uint8 [HeaderAddr](#) [4]
- uint8 [HeaderSize](#) [4]
- uint8 [StorageIvAddr](#) [4]
- uint8 [StorageIvSize](#) [4]
- uint8 [StorageSignAddr](#) [4]
- uint8 [StorageSignSize](#) [4]

3.189.1 Detailed Description

Host image verify information.

Definition at line 306 of file Hsm_Hal.h.

3.189.2 Member Data Documentation

3.189.2.1 HeaderAddr

```
uint8 HSM_ImageVerifyType__::HeaderAddr[4]
```

Definition at line 318 of file Hsm_Hal.h.

3.189.2.2 HeaderSize

```
uint8 HSM_ImageVerifyType__::HeaderSize[4]
```

Definition at line 319 of file Hsm_Hal.h.

3.189.2.3 PubkeyAddr

```
uint8 HSM_ImageVerifyType__::PubkeyAddr[4]
```

Definition at line 314 of file Hsm_Hal.h.

3.189.2.4 PubkeySize

```
uint8 HSM_ImageVerifyType__::PubkeySize[4]
```

Definition at line 315 of file Hsm_Hal.h.

3.189.2.5 StorageAlg

```
uint8 HSM_ImageVerifyType__::StorageAlg
```

Definition at line 310 of file Hsm_Hal.h.

3.189.2.6 StorageEncryptionFlag

```
uint8 HSM_ImageVerifyType__::StorageEncryptionFlag
```

Definition at line 311 of file Hsm_Hal.h.

3.189.2.7 StorageImageAddr

```
uint8 HSM_ImageVerifyType__::StorageImageAddr[4]
```

Definition at line 316 of file Hsm_Hal.h.

3.189.2.8 StorageImageSize

```
uint8 HSM_ImageVerifyType__::StorageImageSize[4]
```

Definition at line 317 of file Hsm_Hal.h.

3.189.2.9 StorageIvAddr

```
uint8 HSM_ImageVerifyType__::StorageIvAddr[4]
```

Definition at line 320 of file Hsm_Hal.h.

3.189.2.10 StorageIvSize

```
uint8 HSM_ImageVerifyType__::StorageIvSize[4]
```

Definition at line 321 of file Hsm_Hal.h.

3.189.2.11 StorageSignAddr

```
uint8 HSM_ImageVerifyType__::StorageSignAddr[4]
```

Definition at line 322 of file Hsm_Hal.h.

3.189.2.12 StorageSignSize

```
uint8 HSM_ImageVerifyType__::StorageSignSize[4]
```

Definition at line 323 of file Hsm_Hal.h.

3.189.2.13 Type

```
uint8 HSM_ImageVerifyType__::Type
```

Definition at line 309 of file Hsm_Hal.h.

3.189.2.14 UpdateVersionFlag

```
uint8 HSM_ImageVerifyType__::UpdateVersionFlag
```

Definition at line 308 of file Hsm_Hal.h.

3.189.2.15 VersionAddr

```
uint8 HSM_ImageVerifyType__::VersionAddr[4]
```

Definition at line 312 of file Hsm_Hal.h.

3.189.2.16 VersionSize

```
uint8 HSM_ImageVerifyType__::VersionSize[4]
```

Definition at line 313 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.190 HSM_InOutMacType Struct Reference

the Input and output struct about gen/ver Mac interface.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_InOutType BasicInOut](#)
- uint8 * [MacInBuf](#)
- uint32 [MacInBufLen](#)
- uint8 * [Vry](#)

3.190.1 Detailed Description

the Input and output struct about gen/ver Mac interface.

Definition at line 602 of file Hsm_Hal.h.

3.190.2 Member Data Documentation

3.190.2.1 BasicInOut

[HSM_InOutType](#) HSM_InOutMacType::BasicInOut

Definition at line 604 of file Hsm_Hal.h.

3.190.2.2 MacInBuf

uint8* HSM_InOutMacType::MacInBuf

algo type, eg.AES_128 AES_256

Definition at line 605 of file Hsm_Hal.h.

3.190.2.3 MacInBufLen

uint32 HSM_InOutMacType::MacInBufLen

when sym algo mode, eg.ECB/CBC

Definition at line 606 of file Hsm_Hal.h.

3.190.2.4 Vry

uint8* HSM_InOutMacType::Vry

Definition at line 608 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.191 HSM_InOutSignType Struct Reference

the Input and output struct about gen/ver signature interface.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_InOutType BasicInOut](#)
- uint8 * [SignInBuf](#)
- uint32 [SignInBufLen](#)
- uint8 * [Vry](#)

3.191.1 Detailed Description

the Input and output struct about gen/ver signature interface.

Definition at line 614 of file Hsm_Hal.h.

3.191.2 Member Data Documentation

3.191.2.1 BasicInOut

```
HSM\_InOutType HSM_InOutSignType::BasicInOut
```

Definition at line 616 of file Hsm_Hal.h.

3.191.2.2 SignInBuf

```
uint8* HSM_InOutSignType::SignInBuf
```

algo type, eg.AES_128 AES_256

Definition at line 618 of file Hsm_Hal.h.

3.191.2.3 SignInBufLen

```
uint32 HSM_InOutSignType::SignInBufLen
```

when sym algo mode, eg.ECB/CBC

Definition at line 619 of file Hsm_Hal.h.

3.191.2.4 Vry

```
uint8* HSM_InOutSignType::Vry
```

Definition at line 620 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.192 HSM_InOutType Struct Reference

the basic Input and output struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 * [InBuf](#)
- uint32 [InBufLen](#)
- uint8 * [OutBuf](#)
- uint32 * [OutBufLen](#)

3.192.1 Detailed Description

the basic Input and output struct.

Definition at line 591 of file Hsm_Hal.h.

3.192.2 Member Data Documentation

3.192.2.1 InBuf

```
uint8* HSM_InOutType::InBuf
```

process mode, eg.init/update/finsh/onepass

Definition at line 593 of file Hsm_Hal.h.

3.192.2.2 InBufLen

```
uint32 HSM_InOutType::InBufLen
```

data padding

Definition at line 594 of file Hsm_Hal.h.

3.192.2.3 OutBuf

```
uint8* HSM_InOutType::OutBuf
```

algo type, eg.AES_128 AES_256

Definition at line 595 of file Hsm_Hal.h.

3.192.2.4 OutBufLen

```
uint32* HSM_InOutType::OutBufLen
```

when sym algo mode, eg.ECB/CBC

Definition at line 596 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.193 HSM_KeyActUseFlagsType Struct Reference

the key active attribute config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- boolean [sign](#)
- boolean [verify](#)
- boolean [encrypt](#)
- boolean [decrypt](#)
- boolean [timestamp](#)
- boolean [secureboot](#)
- boolean [securestorage](#)
- boolean [createkey](#)
- boolean [utcsync](#)
- boolean [transport](#)
- boolean [remove](#)

3.193.1 Detailed Description

the key active attribute config struct.

Definition at line 674 of file Hsm_Hal.h.

3.193.2 Member Data Documentation

3.193.2.1 createkey

`boolean HSM_KeyActUseFlagsType::createkey`

Definition at line 682 of file Hsm_Hal.h.

3.193.2.2 decrypt

`boolean HSM_KeyActUseFlagsType::decrypt`

Definition at line 678 of file Hsm_Hal.h.

3.193.2.3 encrypt

`boolean HSM_KeyActUseFlagsType::encrypt`

Definition at line 677 of file Hsm_Hal.h.

3.193.2.4 remove

`boolean HSM_KeyActUseFlagsType::remove`

Definition at line 685 of file Hsm_Hal.h.

3.193.2.5 secureboot

`boolean HSM_KeyActUseFlagsType::secureboot`

Definition at line 680 of file Hsm_Hal.h.

3.193.2.6 securestorage

`boolean HSM_KeyActUseFlagsType::securestorage`

Definition at line 681 of file Hsm_Hal.h.

3.193.2.7 sign

```
boolean HSM_KeyActUseFlagsType::sign
```

Definition at line 675 of file Hsm_Hal.h.

3.193.2.8 timestamp

```
boolean HSM_KeyActUseFlagsType::timestamp
```

Definition at line 679 of file Hsm_Hal.h.

3.193.2.9 transport

```
boolean HSM_KeyActUseFlagsType::transport
```

Definition at line 684 of file Hsm_Hal.h.

3.193.2.10 utcsync

```
boolean HSM_KeyActUseFlagsType::utcsync
```

Definition at line 683 of file Hsm_Hal.h.

3.193.2.11 verify

```
boolean HSM_KeyActUseFlagsType::verify
```

Definition at line 676 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.194 HSM_KeyFlagsElementType Struct Reference

the key flags.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint16 [use_flags](#)
- uint8 [trnsp_flags](#)
- uint32 [auth_flag](#)
- uint8 [auth_size](#)
- uint16 [auth_value_exist_flags](#)
- uint8 [auth_value](#) [32]

3.194.1 Detailed Description

the key flags.

Definition at line 637 of file Hsm_Hal.h.

3.194.2 Member Data Documentation

3.194.2.1 auth_flag

```
uint32 HSM_KeyFlagsElementType::auth_flag
```

Definition at line 644 of file Hsm_Hal.h.

3.194.2.2 auth_size

```
uint8 HSM_KeyFlagsElementType::auth_size
```

Definition at line 646 of file Hsm_Hal.h.

3.194.2.3 auth_value

```
uint8 HSM_KeyFlagsElementType::auth_value[32]
```

Definition at line 650 of file Hsm_Hal.h.

3.194.2.4 auth_value_exist_flags

```
uint16 HSM_KeyFlagsElementType::auth_value_exist_flags
```

Definition at line 648 of file Hsm_Hal.h.

3.194.2.5 trnsp_flags

```
uint8 HSM_KeyFlagsElementType::trnsp_flags
```

Definition at line 642 of file Hsm_Hal.h.

3.194.2.6 use_flags

```
uint16 HSM_KeyFlagsElementType::use_flags
```

Definition at line 640 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.195 HSM_KeyHandleInfoType Struct Reference

Specifies the Key handle info content,size is 40Bytes.

Public Attributes

- uint32 [KeyHandle](#)
- uint8 [AuthValue](#) [32]
- uint32 [AuthSize](#)

3.195.1 Detailed Description

Specifies the Key handle info content,size is 40Bytes.

Definition at line 49 of file Hsm_Hal.c.

3.195.2 Member Data Documentation

3.195.2.1 AuthSize

```
uint32 HSM_KeyHandleInfoType::AuthSize
```

authentication size

Definition at line 53 of file Hsm_Hal.c.

3.195.2.2 AuthValue

```
uint8 HSM_KeyHandleInfoType::AuthValue[32]
```

authentication

Definition at line 52 of file Hsm_Hal.c.

3.195.2.3 KeyHandle

```
uint32 HSM_KeyHandleInfoType::KeyHandle
```

key handle id by hsm

Definition at line 51 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.196 HSM_KeyIndexInfoType Struct Reference

Specifies the Key index info content,size is 8Bytes.

Public Attributes

- uint32 [KeyValid](#)
- uint32 [KeyIndex](#)

3.196.1 Detailed Description

Specifies the Key index info content,size is 8Bytes.

Definition at line 68 of file Hsm_Hal.c.

3.196.2 Member Data Documentation

3.196.2.1 KeyIndex

```
uint32 HSM_KeyIndexInfoType::KeyIndex
```

key index by host

Definition at line 71 of file Hsm_Hal.c.

3.196.2.2 KeyValid

```
uint32 HSM_KeyIndexInfoType::KeyValid
```

key valid flag: 0x52525252 means valid, other means invalid

Definition at line 70 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.197 HSM_KeySlotInfoType Struct Reference

Specifies the Key slot info content,size is 8Bytes.

Public Attributes

- uint32 [SlotValid](#)
- uint32 [SlotIndex](#)

3.197.1 Detailed Description

Specifies the Key slot info content,size is 8Bytes.

Definition at line 59 of file Hsm_Hal.c.

3.197.2 Member Data Documentation

3.197.2.1 SlotIndex

```
uint32 HSM_KeySlotInfoType::SlotIndex
```

slot index

Definition at line 62 of file Hsm_Hal.c.

3.197.2.2 SlotValid

```
uint32 HSM_KeySlotInfoType::SlotValid
```

slot valid flag: 0xA5 * 4 means invalid, others depends on 'dataValidFlag'

Definition at line 61 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.198 HSM_KeyStatusType Struct Reference

the key status in hsm.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_KeyId](#) TargetKeyId
- [HSM_KeyId](#) CertificationKeyId
- uint32 [CertificationAuthSize](#)
- uint8 * [CertificationAuth](#)
- uint32 * [KeyStatusSize](#)
- ehsm_key_status_st * [KeyStatus](#)

3.198.1 Detailed Description

the key status in hsm.

Definition at line 900 of file Hsm_Hal.h.

3.198.2 Member Data Documentation

3.198.2.1 CertificationAuth

```
uint8* HSM_KeyStatusType::CertificationAuth
```

Definition at line 905 of file Hsm_Hal.h.

3.198.2.2 CertificationAuthSize

```
uint32 HSM_KeyStatusType::CertificationAuthSize
```

Definition at line 904 of file Hsm_Hal.h.

3.198.2.3 CertificationKeyId

```
HSM\_KeyId HSM_KeyStatusType::CertificationKeyId
```

Definition at line 903 of file Hsm_Hal.h.

3.198.2.4 KeyStatus

```
ehsm_key_status_st* HSM_KeyStatusType::KeyStatus
```

Definition at line 907 of file Hsm_Hal.h.

3.198.2.5 KeyStatusSize

```
uint32* HSM_KeyStatusType::KeyStatusSize
```

Definition at line 906 of file Hsm_Hal.h.

3.198.2.6 TargetKeyId

```
HSM_KeyId HSM_KeyStatusType::TargetKeyId
```

Definition at line 902 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.199 HSM_KeyUsagesType Struct Reference

the key attribute config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_KeyFlagsElementType sign](#)
- [HSM_KeyFlagsElementType verify](#)
- [HSM_KeyFlagsElementType encrypt](#)
- [HSM_KeyFlagsElementType decrypt](#)
- [HSM_KeyFlagsElementType timestamp](#)
- [HSM_KeyFlagsElementType secureboot](#)
- [HSM_KeyFlagsElementType securestorage](#)
- [HSM_KeyFlagsElementType dhkey](#)
- [HSM_KeyFlagsElementType utcsync](#)
- [HSM_KeyFlagsElementType transport](#)
- [HSM_KeyFlagsElementType remove](#)

3.199.1 Detailed Description

the key attribute config struct.

Definition at line 656 of file Hsm_Hal.h.

3.199.2 Member Data Documentation

3.199.2.1 decrypt

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::decrypt

Definition at line 661 of file Hsm_Hal.h.

3.199.2.2 dhkey

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::dhkey

Definition at line 665 of file Hsm_Hal.h.

3.199.2.3 encrypt

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::encrypt

Definition at line 660 of file Hsm_Hal.h.

3.199.2.4 remove

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::remove

Definition at line 668 of file Hsm_Hal.h.

3.199.2.5 secureboot

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::secureboot

Definition at line 663 of file Hsm_Hal.h.

3.199.2.6 securestorage

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::securestorage

Definition at line 664 of file Hsm_Hal.h.

3.199.2.7 sign

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::sign

Definition at line 658 of file Hsm_Hal.h.

3.199.2.8 timestamp

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::timestamp

Definition at line 662 of file Hsm_Hal.h.

3.199.2.9 transport

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::transport

Definition at line 667 of file Hsm_Hal.h.

3.199.2.10 utcsync

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::utcsync

Definition at line 666 of file Hsm_Hal.h.

3.199.2.11 verify

[HSM_KeyFlagsElementType](#) HSM_KeyUsagesType::verify

Definition at line 659 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.200 HSM_PlainKeyCfgType Struct Reference

the set plain key config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_GenKeyAlgo](#) KeyAlgo
- uint16 [RandomKeySize](#)
- [HSM_KeyStorageType](#) KeyType
- uint8 * [PrivKey](#)
- uint32 [PrivKeyLen](#)
- uint8 * [PubKey](#)
- uint32 [PubKeyLen](#)
- uint8 * [KeyUsages](#)
- uint8 [KeyUsagesCnt](#)
- uint8 * [AuthValue](#)
- uint32 [AuthValueSize](#)
- uint32 [ValidUtil](#)
- uint16 [ExtParam](#)
- [HSM_KeyId](#) KeyId

3.200.1 Detailed Description

the set plain key config struct.

Definition at line 820 of file Hsm_Hal.h.

3.200.2 Member Data Documentation

3.200.2.1 AuthValue

```
uint8* HSM_PlainKeyCfgType::AuthValue
```

- key authentication value

Definition at line 832 of file Hsm_Hal.h.

3.200.2.2 AuthValueSize

```
uint32 HSM_PlainKeyCfgType::AuthValueSize
```

- key authentication value size

Definition at line 833 of file Hsm_Hal.h.

3.200.2.3 ExtParam

uint16 HSM_PlainKeyCfgType::ExtParam

- e param size only rsa key, default set 0

Definition at line 836 of file Hsm_Hal.h.

3.200.2.4 KeyAlgo

[HSM_GenKeyAlgo](#) HSM_PlainKeyCfgType::KeyAlgo

key generate algo

Definition at line 822 of file Hsm_Hal.h.

3.200.2.5 KeyId

[HSM_KeyId](#) HSM_PlainKeyCfgType::KeyId

- key id according to key algo type

Definition at line 838 of file Hsm_Hal.h.

3.200.2.6 KeyType

[HSM_KeyStorageType](#) HSM_PlainKeyCfgType::KeyType

Definition at line 824 of file Hsm_Hal.h.

3.200.2.7 KeyUsages

uint8* HSM_PlainKeyCfgType::KeyUsages

- key usages for attribute, eg. enc/dec..

Definition at line 829 of file Hsm_Hal.h.

3.200.2.8 KeyUsagesCnt

```
uint8 HSM_PlainKeyCfgType::KeyUsagesCnt
```

- key usages size

Definition at line 830 of file Hsm_Hal.h.

3.200.2.9 PrivKey

```
uint8* HSM_PlainKeyCfgType::PrivKey
```

*pointer to the buffer,the private key store in buffer

Definition at line 825 of file Hsm_Hal.h.

3.200.2.10 PrivKeyLen

```
uint32 HSM_PlainKeyCfgType::PrivKeyLen
```

- private key len

Definition at line 826 of file Hsm_Hal.h.

3.200.2.11 PubKey

```
uint8* HSM_PlainKeyCfgType::PubKey
```

*pointer to the buffer,the pubkey key store in buffer

Definition at line 827 of file Hsm_Hal.h.

3.200.2.12 PubKeyLen

```
uint32 HSM_PlainKeyCfgType::PubKeyLen
```

- pubkey key len

Definition at line 828 of file Hsm_Hal.h.

3.200.2.13 RandomKeySize

```
uint16 HSM_PlainKeyCfgType::RandomKeySize
```

key size only when keyalgo = RANDOM be used

Definition at line 823 of file Hsm_Hal.h.

3.200.2.14 ValidUtil

```
uint32 HSM_PlainKeyCfgType::ValidUtil
```

Definition at line 834 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.201 Hsm_PriKeyDataType_ Union Reference

the private key struct for all algo.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [SymKey](#) [1024]
- uint8 [EccKey](#) [66]
- uint8 [RsaD](#) [512]
- uint8 [KdfKey](#) [512]
- uint8 [DhKey](#) [512]
- [HSM_DhPriKeyType](#) Dh
- [HSM_RsaPrvCrtType](#) RsaCtr

3.201.1 Detailed Description

the private key struct for all algo.

Definition at line 757 of file Hsm_Hal.h.

3.201.2 Member Data Documentation

3.201.2.1 Dh

[HSM_DhPriKeyType](#) Hsm_PriKeyDataType_::Dh

Definition at line 764 of file Hsm_Hal.h.

3.201.2.2 DhKey

```
uint8 Hsm_PriKeyDataType_::DhKey[512]
```

Definition at line 763 of file Hsm_Hal.h.

3.201.2.3 EccKey

```
uint8 Hsm_PriKeyDataType_::EccKey[66]
```

Definition at line 760 of file Hsm_Hal.h.

3.201.2.4 KdfKey

```
uint8 Hsm_PriKeyDataType_::KdfKey[512]
```

Definition at line 762 of file Hsm_Hal.h.

3.201.2.5 RsaCtr

[HSM_RsaPrvCrtType](#) Hsm_PriKeyDataType_::RsaCtr

Definition at line 765 of file Hsm_Hal.h.

3.201.2.6 RsaD

```
uint8 Hsm_PriKeyDataType_::RsaD[512]
```

Definition at line 761 of file Hsm_Hal.h.

3.201.2.7 SymKey

```
uint8 Hsm_PriKeyDataType_::SymKey[1024]
```

Definition at line 759 of file Hsm_Hal.h.

The documentation for this union was generated from the following file:

- [Hsm_Hal.h](#)

3.202 Hsm_PubKeyDataType_ Union Reference

the public key struct for all algo.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_RsaPubKeyType](#) Rsa
- [HSM_EccPubKeyType](#) Ecc
- [HSM_DhPubKeyType](#) Dh

3.202.1 Detailed Description

the public key struct for all algo.

Definition at line 727 of file Hsm_Hal.h.

3.202.2 Member Data Documentation

3.202.2.1 Dh

```
HSM\_DhPubKeyType Hsm_PubKeyDataType_::Dh
```

Definition at line 731 of file Hsm_Hal.h.

3.202.2.2 Ecc

```
HSM\_EccPubKeyType Hsm_PubKeyDataType_::Ecc
```

Definition at line 730 of file Hsm_Hal.h.

3.202.2.3 Rsa

[HSM_RsaPubKeyType](#) `Hsm_PubKeyDataType_::Rsa`

Definition at line 729 of file `Hsm_Hal.h`.

The documentation for this union was generated from the following file:

- [Hsm_Hal.h](#)

3.203 HSM_RamKeyHeadType Struct Reference

Specifies the Ram Key space header.

Public Attributes

- `uint8 *` [KeyMemStart](#)
- `uint32` [KeyMemUsedSize](#)
- `uint32` [KeyNum](#)
- [HSM_RamKeyInfoType](#) `KeyInfo` [[HSM_RAM_KEY_MAX_NUM](#)]

3.203.1 Detailed Description

Specifies the Ram Key space header.

Definition at line 107 of file `Hsm_Hal.c`.

3.203.2 Member Data Documentation

3.203.2.1 KeyInfo

[HSM_RamKeyInfoType](#) `HSM_RamKeyHeadType::KeyInfo` [[HSM_RAM_KEY_MAX_NUM](#)]

ram key handle buffer slot

Definition at line 112 of file `Hsm_Hal.c`.

3.203.2.2 KeyMemStart

`uint8*` `HSM_RamKeyHeadType::KeyMemStart`

ram key handle buffer address

Definition at line 109 of file `Hsm_Hal.c`.

3.203.2.3 KeyMemUsedSize

```
uint32 HSM_RamKeyHeadType::KeyMemUsedSize
```

ram key space used size

Definition at line 110 of file Hsm_Hal.c.

3.203.2.4 KeyNum

```
uint32 HSM_RamKeyHeadType::KeyNum
```

alread existed key number

Definition at line 111 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.204 HSM_RamKeyInfoType Struct Reference

Specifies the Ram Key info.

Public Attributes

- uint8 * [KeyHandleAddr](#)
- uint32 [Used](#)
- uint32 [KeyIndex](#)

3.204.1 Detailed Description

Specifies the Ram Key info.

Definition at line 97 of file Hsm_Hal.c.

3.204.2 Member Data Documentation

3.204.2.1 KeyHandleAddr

```
uint8* HSM_RamKeyInfoType::KeyHandleAddr
```

pointer key handle

Definition at line 99 of file Hsm_Hal.c.

3.204.2.2 KeyIndex

```
uint32 HSM_RamKeyInfoType::KeyIndex
```

key index

Definition at line 101 of file Hsm_Hal.c.

3.204.2.3 Used

```
uint32 HSM_RamKeyInfoType::Used
```

key slot be used

Definition at line 100 of file Hsm_Hal.c.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.c](#)

3.205 Hsm_RsaCrtType_ Struct Reference

the rsa private key struct when CRT.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [p](#) [256]
- uint8 [q](#) [256]
- uint8 [dp](#) [256]
- uint8 [dq](#) [256]
- uint8 [u](#) [256]

3.205.1 Detailed Description

the rsa private key struct when CRT.

Definition at line 745 of file Hsm_Hal.h.

3.205.2 Member Data Documentation

3.205.2.1 dp

```
uint8 Hsm_RsaCrtType_::dp[256]
```

Definition at line 749 of file Hsm_Hal.h.

3.205.2.2 dq

```
uint8 Hsm_RsaCrtType_::dq[256]
```

Definition at line 750 of file Hsm_Hal.h.

3.205.2.3 p

```
uint8 Hsm_RsaCrtType_::p[256]
```

Definition at line 747 of file Hsm_Hal.h.

3.205.2.4 q

```
uint8 Hsm_RsaCrtType_::q[256]
```

Definition at line 748 of file Hsm_Hal.h.

3.205.2.5 u

```
uint8 Hsm_RsaCrtType_::u[256]
```

Definition at line 751 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.206 HSM_RsaPubKeyType Struct Reference

the rsa pubkey struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [e](#) [512]
- uint8 [n](#) [512]

3.206.1 Detailed Description

the rsa pubkey struct.

Definition at line 699 of file Hsm_Hal.h.

3.206.2 Member Data Documentation

3.206.2.1 [e](#)

```
uint8 HSM_RsaPubKeyType::e[512]
```

Definition at line 701 of file Hsm_Hal.h.

3.206.2.2 [n](#)

```
uint8 HSM_RsaPubKeyType::n[512]
```

Definition at line 702 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.207 HSM_SecretKeyCfgType Struct Reference

the Secret Key config struct for key import or export by secret.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_GenKeyAlgo](#) KeyAlgo
- [HSM_KeyStorageType](#) KeyType
- [HSM_KeyActUseFlagsType](#) * SetKeyUseFlag
- uint8 * [AuthValue](#)
- uint32 * [AuthVAlueSize](#)
- uint8 * [SecretKeyBlob](#)
- uint32 * [SetKeyBlobSize](#)

3.207.1 Detailed Description

the Secret Key config struct for key import or export by secret.

Definition at line 771 of file Hsm_Hal.h.

3.207.2 Member Data Documentation

3.207.2.1 AuthValue

```
uint8* HSM_SecretKeyCfgType::AuthValue
```

Definition at line 776 of file Hsm_Hal.h.

3.207.2.2 AuthValueSize

```
uint32* HSM_SecretKeyCfgType::AuthValueSize
```

Definition at line 777 of file Hsm_Hal.h.

3.207.2.3 KeyAlgo

```
HSM_GenKeyAlgo HSM_SecretKeyCfgType::KeyAlgo
```

Definition at line 773 of file Hsm_Hal.h.

3.207.2.4 KeyType

```
HSM_KeyStorageType HSM_SecretKeyCfgType::KeyType
```

Definition at line 774 of file Hsm_Hal.h.

3.207.2.5 SecretKeyBlob

```
uint8* HSM_SecretKeyCfgType::SecretKeyBlob
```

only for export/import secret buffer , buffer size must \geq 3924 bytes

Definition at line 779 of file Hsm_Hal.h.

3.207.2.6 SetKeyBlobSize

```
uint32* HSM_SecretKeyCfgType::SetKeyBlobSize
```

only for export/import buffer size, buffer size must ≥ 3924 bytes

Definition at line 780 of file Hsm_Hal.h.

3.207.2.7 SetKeyUseFlag

```
HSM_KeyActUseFlagsType* HSM_SecretKeyCfgType::SetKeyUseFlag
```

Definition at line 775 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.208 HSM_SecureUpgradeType__ Struct Reference

Host image secure upgrade information.

```
#include <Hsm_Hal.h>
```

Public Attributes

- uint8 [StorageAlg](#)
- uint8 [UpgradeAlg](#)
- uint8 [StorageEncryptionFlag](#)
- uint8 [UpgradeDecryptionFlag](#)
- uint8 [ProcessMode](#)
- uint8 [CheckVersionFlag](#)
- uint8 [Rev](#) [2]
- uint8 [UpgradeSignAddr](#) [4]
- uint8 [UpgradeSignSize](#) [4]
- uint8 [UpgradeVersionAddr](#) [4]
- uint8 [UpgradeVersionSize](#) [4]
- uint8 [UpgradelImageAddr](#) [4]
- uint8 [UpgradelImageSize](#) [4]
- uint8 [StorageImageAddr](#) [4]
- uint8 [StorageImageSize](#) [4]
- uint8 [MacSignAddr](#) [4]
- uint8 [MacSignSize](#) [4]
- uint8 [UpgradePubkeyAddr](#) [4]
- uint8 [UpgradePubkeySize](#) [4]
- uint8 [UpgradelvAddr](#) [4]
- uint8 [UpgradelvSize](#) [4]
- uint8 [StorageelvAddr](#) [4]
- uint8 [StorageelvSize](#) [4]
- uint8 [HeaderAddr](#) [4]
- uint8 [HeaderSize](#) [4]
- uint8 [CtxAddr](#) [4]
- uint8 [CtxSize](#) [4]

3.208.1 Detailed Description

Host image secure upgrade information.

Definition at line 272 of file Hsm_Hal.h.

3.208.2 Member Data Documentation

3.208.2.1 CheckVersionFlag

```
uint8 HSM_SecureUpgradeType__::CheckVersionFlag
```

Definition at line 279 of file Hsm_Hal.h.

3.208.2.2 CtxAddr

```
uint8 HSM_SecureUpgradeType__::CtxAddr[4]
```

Definition at line 299 of file Hsm_Hal.h.

3.208.2.3 CtxSize

```
uint8 HSM_SecureUpgradeType__::CtxSize[4]
```

Definition at line 300 of file Hsm_Hal.h.

3.208.2.4 HeaderAddr

```
uint8 HSM_SecureUpgradeType__::HeaderAddr[4]
```

Definition at line 297 of file Hsm_Hal.h.

3.208.2.5 HeaderSize

```
uint8 HSM_SecureUpgradeType__::HeaderSize[4]
```

Definition at line 298 of file Hsm_Hal.h.

3.208.2.6 MacSignAddr

```
uint8 HSM_SecureUpgradeType__::MacSignAddr[4]
```

Definition at line 289 of file Hsm_Hal.h.

3.208.2.7 MacSignSize

```
uint8 HSM_SecureUpgradeType__::MacSignSize[4]
```

Definition at line 290 of file Hsm_Hal.h.

3.208.2.8 ProcessMode

```
uint8 HSM_SecureUpgradeType__::ProcessMode
```

Definition at line 278 of file Hsm_Hal.h.

3.208.2.9 Rev

```
uint8 HSM_SecureUpgradeType__::Rev[2]
```

Definition at line 280 of file Hsm_Hal.h.

3.208.2.10 StorageAlg

```
uint8 HSM_SecureUpgradeType__::StorageAlg
```

Definition at line 274 of file Hsm_Hal.h.

3.208.2.11 StorageEncryptionFlag

```
uint8 HSM_SecureUpgradeType__::StorageEncryptionFlag
```

Definition at line 276 of file Hsm_Hal.h.

3.208.2.12 StorageImageAddr

```
uint8 HSM_SecureUpgradeType__::StorageImageAddr[4]
```

Definition at line 287 of file Hsm_Hal.h.

3.208.2.13 StorageImageSize

```
uint8 HSM_SecureUpgradeType__::StorageImageSize[4]
```

Definition at line 288 of file Hsm_Hal.h.

3.208.2.14 StorageIvAddr

```
uint8 HSM_SecureUpgradeType__::StorageIvAddr[4]
```

Definition at line 295 of file Hsm_Hal.h.

3.208.2.15 StorageIvSize

```
uint8 HSM_SecureUpgradeType__::StorageIvSize[4]
```

Definition at line 296 of file Hsm_Hal.h.

3.208.2.16 UpgradeAlg

```
uint8 HSM_SecureUpgradeType__::UpgradeAlg
```

Definition at line 275 of file Hsm_Hal.h.

3.208.2.17 UpgradeDecryptionFlag

```
uint8 HSM_SecureUpgradeType__::UpgradeDecryptionFlag
```

Definition at line 277 of file Hsm_Hal.h.

3.208.2.18 UpgradelImageAddr

```
uint8 HSM_SecureUpgradeType__::UpgradeImageAddr[4]
```

Definition at line 285 of file Hsm_Hal.h.

3.208.2.19 UpgradelImageSize

```
uint8 HSM_SecureUpgradeType__::UpgradeImageSize[4]
```

Definition at line 286 of file Hsm_Hal.h.

3.208.2.20 UpgradelvAddr

```
uint8 HSM_SecureUpgradeType__::UpgradeIvAddr[4]
```

Definition at line 293 of file Hsm_Hal.h.

3.208.2.21 UpgradelvSize

```
uint8 HSM_SecureUpgradeType__::UpgradeIvSize[4]
```

Definition at line 294 of file Hsm_Hal.h.

3.208.2.22 UpgradePubkeyAddr

```
uint8 HSM_SecureUpgradeType__::UpgradePubkeyAddr[4]
```

Definition at line 291 of file Hsm_Hal.h.

3.208.2.23 UpgradePubkeySize

```
uint8 HSM_SecureUpgradeType__::UpgradePubkeySize[4]
```

Definition at line 292 of file Hsm_Hal.h.

3.208.2.24 UpgradeSignAddr

```
uint8 HSM_SecureUpgradeType__::UpgradeSignAddr[4]
```

Definition at line 281 of file Hsm_Hal.h.

3.208.2.25 UpgradeSignSize

```
uint8 HSM_SecureUpgradeType__::UpgradeSignSize[4]
```

Definition at line 282 of file Hsm_Hal.h.

3.208.2.26 UpgradeVersionAddr

```
uint8 HSM_SecureUpgradeType__::UpgradeVersionAddr[4]
```

Definition at line 283 of file Hsm_Hal.h.

3.208.2.27 UpgradeVersionSize

```
uint8 HSM_SecureUpgradeType__::UpgradeVersionSize[4]
```

Definition at line 284 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.209 HSM_SymCfgType Struct Reference

Specifies the symmetric algo config struct.

```
#include <Hsm_Hal.h>
```

Public Attributes

- [HSM_KeyId KeyId](#)
- [uint8 * Iv](#)
- [uint8 IvLen](#)
- [uint8 Sync](#)
- [HSM_PaddingType Padding](#)
- [HSM_CipherDirection CipherDir](#)
- [HSM_CipherMode CMode](#)
- [HSM_SymAlgoType SymAlgo](#)

3.209.1 Detailed Description

Specifies the symmetric algo config struct.

Definition at line 554 of file Hsm_Hal.h.

3.209.2 Member Data Documentation

3.209.2.1 CipherDir

`HSM_CipherDirection` `HSM_SymCfgType::CipherDir`

request operation, eg.decrypt/encrypt

Definition at line 561 of file Hsm_Hal.h.

3.209.2.2 CMode

`HSM_CipherMode` `HSM_SymCfgType::CMode`

for sym algo and CMAC algo, eg.ECB/CBC..

Definition at line 562 of file Hsm_Hal.h.

3.209.2.3 Iv

`uint8*` `HSM_SymCfgType::Iv`

Definition at line 557 of file Hsm_Hal.h.

3.209.2.4 IvLen

`uint8` `HSM_SymCfgType::IvLen`

Definition at line 558 of file Hsm_Hal.h.

3.209.2.5 KeyId

[HSM_KeyId](#) HSM_SymCfgType::KeyId

Key Index

Definition at line 556 of file Hsm_Hal.h.

3.209.2.6 Padding

[HSM_PaddingType](#) HSM_SymCfgType::Padding

data padding

Definition at line 560 of file Hsm_Hal.h.

3.209.2.7 SymAlgo

[HSM_SymAlgoType](#) HSM_SymCfgType::SymAlgo

symmetric algo type, eg.AES_128/AES_256

Definition at line 563 of file Hsm_Hal.h.

3.209.2.8 Sync

uint8 HSM_SymCfgType::Sync

0:sync 1:async

Definition at line 559 of file Hsm_Hal.h.

The documentation for this struct was generated from the following file:

- [Hsm_Hal.h](#)

3.210 kdf_testvec Struct Reference

```
#include <utest_vecs_st.h>
```


Public Attributes

- const char * [salt](#)
- unsigned int [salt_size](#)
- const char * [plaintext](#)
- unsigned int [psize](#)
- const char * [shared_info](#)
- unsigned int [shared_info_size](#)
- const char * [key](#)
- unsigned int [ksize](#)

3.210.1 Detailed Description

Definition at line 40 of file `utest_vecs_st.h`.

3.210.2 Member Data Documentation

3.210.2.1 [key](#)

```
const char* kdf_testvec::key
```

Definition at line 47 of file `utest_vecs_st.h`.

3.210.2.2 [ksize](#)

```
unsigned int kdf_testvec::ksize
```

Definition at line 48 of file `utest_vecs_st.h`.

3.210.2.3 [plaintext](#)

```
const char* kdf_testvec::plaintext
```

Definition at line 43 of file `utest_vecs_st.h`.

3.210.2.4 [psize](#)

```
unsigned int kdf_testvec::psize
```

Definition at line 44 of file `utest_vecs_st.h`.

3.210.2.5 salt

```
const char* kdf_testvec::salt
```

Definition at line 41 of file utest_vecs_st.h.

3.210.2.6 salt_size

```
unsigned int kdf_testvec::salt_size
```

Definition at line 42 of file utest_vecs_st.h.

3.210.2.7 shared_info

```
const char* kdf_testvec::shared_info
```

Definition at line 45 of file utest_vecs_st.h.

3.210.2.8 shared_info_size

```
unsigned int kdf_testvec::shared_info_size
```

Definition at line 46 of file utest_vecs_st.h.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.211 key_act_use_flags_t Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_bool_t](#) sign
- [ehsm_bool_t](#) verify
- [ehsm_bool_t](#) encrypt
- [ehsm_bool_t](#) decrypt
- [ehsm_bool_t](#) timestamp
- [ehsm_bool_t](#) secureboot
- [ehsm_bool_t](#) securestorage
- [ehsm_bool_t](#) createkey
- [ehsm_bool_t](#) utcsync
- [ehsm_bool_t](#) transport
- [ehsm_bool_t](#) remove

3.211.1 Detailed Description

Definition at line 206 of file eHSM_If_Evita_Types_Ip.h.

3.211.2 Member Data Documentation

3.211.2.1 createkey

`ehsm_bool_t key_act_use_flags_t::createkey`

Definition at line 214 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.2 decrypt

`ehsm_bool_t key_act_use_flags_t::decrypt`

Definition at line 210 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.3 encrypt

`ehsm_bool_t key_act_use_flags_t::encrypt`

Definition at line 209 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.4 remove

`ehsm_bool_t key_act_use_flags_t::remove`

Definition at line 217 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.5 secureboot

`ehsm_bool_t key_act_use_flags_t::secureboot`

Definition at line 212 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.6 securestorage

`ehsm_bool_t key_act_use_flags_t::securestorage`

Definition at line 213 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.7 sign

`ehsm_bool_t key_act_use_flags_t::sign`

Definition at line 207 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.8 timestamp

`ehsm_bool_t key_act_use_flags_t::timestamp`

Definition at line 211 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.9 transport

`ehsm_bool_t key_act_use_flags_t::transport`

Definition at line 216 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.10 utcsync

`ehsm_bool_t key_act_use_flags_t::utcsync`

Definition at line 215 of file eHSM_If_Evita_Types_Ip.h.

3.211.2.11 verify

`ehsm_bool_t key_act_use_flags_t::verify`

Definition at line 208 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.212 key_info_st Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint32_t](#) storage_key_type
- union {
 - [ehsm_uint16_t](#) kdf_key_size
 - [ehsm_uint16_t](#) dh_key_size
 - [ehsm_uint16_t](#) rsa_e_bytes_size
 - [ehsm_uint16_t](#) random_key_size
 - [ehsm_uint32_t](#) key_handle
 - [ehsm_dh_param_size_info_st](#) dh_param
- [storage_info](#)

3.212.1 Detailed Description

Definition at line 240 of file eHSM_If_Evita_Types_Ip.h.

3.212.2 Member Data Documentation

3.212.2.1 dh_key_size

```
ehsm\_uint16\_t key_info_st::dh_key_size
```

Definition at line 245 of file eHSM_If_Evita_Types_Ip.h.

3.212.2.2 dh_param

```
ehsm\_dh\_param\_size\_info\_st key_info_st::dh_param
```

Definition at line 249 of file eHSM_If_Evita_Types_Ip.h.

3.212.2.3 kdf_key_size

```
ehsm\_uint16\_t key_info_st::kdf_key_size
```

Definition at line 244 of file eHSM_If_Evita_Types_Ip.h.

3.212.2.4 key_handle

```
ehsm_uint32_t key_info_st::key_handle
```

Definition at line 248 of file eHSM_If_Evita_Types_Ip.h.

3.212.2.5 random_key_size

```
ehsm_uint16_t key_info_st::random_key_size
```

Definition at line 247 of file eHSM_If_Evita_Types_Ip.h.

3.212.2.6 rsa_e_bytes_size

```
ehsm_uint16_t key_info_st::rsa_e_bytes_size
```

Definition at line 246 of file eHSM_If_Evita_Types_Ip.h.

3.212.2.7 storage_info

```
union { ... } key_info_st::storage_info
```

3.212.2.8 storage_key_type

```
ehsm_uint32_t key_info_st::storage_key_type
```

Definition at line 242 of file eHSM_If_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.213 kpp_testvec Struct Reference

Test struct for shared secret.

```
#include <utest_vecs_st.h>
```

Public Attributes

- const unsigned char * [secret](#)
- const unsigned char * [b_secret](#)
- const unsigned char * [b_public](#)
- const unsigned char * [expected_a_public](#)
- const unsigned char * [expected_ss](#)
- unsigned short [secret_size](#)
- unsigned short [b_secret_size](#)
- unsigned short [b_public_size](#)
- unsigned short [expected_a_public_size](#)
- unsigned short [expected_ss_size](#)
- int [genkey](#)
- const unsigned char * [p](#)
- const unsigned char * [q](#)
- const unsigned char * [g](#)
- unsigned short [p_size](#)
- unsigned short [q_size](#)
- unsigned short [g_size](#)

3.213.1 Detailed Description

Test struct for shared secret.

Parameters

<i>secret</i>	The private key of local part.
<i>b_secret</i>	Private key of remote part.
<i>b_public</i>	Public key of remote part.
<i>expected_a_public</i>	The public key of local part.
<i>expected_ss</i>	The expected shared secret.
<i>secret_size</i>	Size of local private key.
<i>b_secret_size</i>	Size of remote private key.
<i>b_public_size</i>	Size of remote public key.
<i>expected_a_public_size</i>	Size of local public key.
<i>expected_ss_size</i>	Size of shared secret.
<i>genkey</i>	This case is only used for <code>get_pubkey_from_privkey</code> if true, otherwise, it's can be used both for <code>get_pubkey_from_privkey</code> and <code>shared_secret</code> .

Definition at line 160 of file `utest_vecs_st.h`.

3.213.2 Member Data Documentation

3.213.2.1 b_public

```
const unsigned char* kpp_testvec::b_public
```

Definition at line 163 of file `utest_vecs_st.h`.

3.213.2.2 b_public_size

```
unsigned short kpp_testvec::b_public_size
```

Definition at line 168 of file `utest_vecs_st.h`.

3.213.2.3 b_secret

```
const unsigned char* kpp_testvec::b_secret
```

Definition at line 162 of file `utest_vecs_st.h`.

3.213.2.4 b_secret_size

```
unsigned short kpp_testvec::b_secret_size
```

Definition at line 167 of file `utest_vecs_st.h`.

3.213.2.5 expected_a_public

```
const unsigned char* kpp_testvec::expected_a_public
```

Definition at line 164 of file `utest_vecs_st.h`.

3.213.2.6 expected_a_public_size

```
unsigned short kpp_testvec::expected_a_public_size
```

Definition at line 169 of file `utest_vecs_st.h`.

3.213.2.7 expected_ss

```
const unsigned char* kpp_testvec::expected_ss
```

Definition at line 165 of file `utest_vecs_st.h`.

3.213.2.8 expected_ss_size

```
unsigned short kpp_testvec::expected_ss_size
```

Definition at line 170 of file `utest_vecs_st.h`.

3.213.2.9 g

```
const unsigned char* kpp_testvec::g
```

Definition at line 175 of file `utest_vecs_st.h`.

3.213.2.10 g_size

```
unsigned short kpp_testvec::g_size
```

Definition at line 178 of file `utest_vecs_st.h`.

3.213.2.11 genkey

```
int kpp_testvec::genkey
```

Definition at line 171 of file `utest_vecs_st.h`.

3.213.2.12 p

```
const unsigned char* kpp_testvec::p
```

Definition at line 173 of file `utest_vecs_st.h`.

3.213.2.13 p_size

```
unsigned short kpp_testvec::p_size
```

Definition at line 176 of file `utest_vecs_st.h`.

3.213.2.14 q

```
const unsigned char* kpp_testvec::q
```

Definition at line 174 of file `utest_vecs_st.h`.

3.213.2.15 q_size

```
unsigned short kpp_testvec::q_size
```

Definition at line 177 of file `utest_vecs_st.h`.

3.213.2.16 secret

```
const unsigned char* kpp_testvec::secret
```

Definition at line 161 of file `utest_vecs_st.h`.

3.213.2.17 secret_size

```
unsigned short kpp_testvec::secret_size
```

Definition at line 166 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.214 mac_t Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t mac_value \[EVITA_MAC_BUF_SIZE\]](#)
- [ehsm_uint32_t mac_size](#)
- [ehsm_utc_time_t utc_time](#)

3.214.1 Detailed Description

Definition at line 442 of file `eHSM_If_Evita_Types_Ip.h`.

3.214.2 Member Data Documentation

3.214.2.1 mac_size

```
ehsm_uint32_t mac_t::mac_size
```

Definition at line 444 of file eHSM_Ip_Evita_Types_Ip.h.

3.214.2.2 mac_value

```
ehsm_uint8_t mac_t::mac_value[EVITA_MAC_BUF_SIZE]
```

Definition at line 443 of file eHSM_Ip_Evita_Types_Ip.h.

3.214.2.3 utc_time

```
ehsm_utc_time_t mac_t::utc_time
```

Definition at line 445 of file eHSM_Ip_Evita_Types_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Ip_Evita_Types_Ip.h](#)

3.215 mailbox_channel Struct Reference

```
#include <eHSM_Mailbox_Ip.h>
```

Public Attributes

- [mailbox_channel_e](#) type
- [ehsm_bool_t](#) surpport_asyn
- [ehsm_uint32_t](#) s2h_start_addr
- [ehsm_uint32_t](#) s2h_word_size
- [ehsm_uint32_t](#) s2h_note_bit
- [ehsm_uint32_t](#) h2s_start_addr
- [ehsm_uint32_t](#) h2s_word_size
- [ehsm_uint32_t](#) h2s_note_bit
- [ehsm_cmd_req_st](#) * cmd_inprogres

3.215.1 Detailed Description

Definition at line 23 of file eHSM_Mailbox_Ip.h.

3.215.2 Member Data Documentation

3.215.2.1 cmd_inprogres

`ehsm_cmd_req_st* mailbox_channel::cmd_inprogres`

Definition at line 33 of file eHSM_Mailbox_lp.h.

3.215.2.2 h2s_note_bit

`ehsm_uint32_t mailbox_channel::h2s_note_bit`

Definition at line 32 of file eHSM_Mailbox_lp.h.

3.215.2.3 h2s_start_addr

`ehsm_uint32_t mailbox_channel::h2s_start_addr`

Definition at line 30 of file eHSM_Mailbox_lp.h.

3.215.2.4 h2s_word_size

`ehsm_uint32_t mailbox_channel::h2s_word_size`

Definition at line 31 of file eHSM_Mailbox_lp.h.

3.215.2.5 s2h_note_bit

`ehsm_uint32_t mailbox_channel::s2h_note_bit`

Definition at line 29 of file eHSM_Mailbox_lp.h.

3.215.2.6 s2h_start_addr

`ehsm_uint32_t mailbox_channel::s2h_start_addr`

Definition at line 27 of file eHSM_Mailbox_lp.h.

3.215.2.7 s2h_word_size

`ehsm_uint32_t mailbox_channel::s2h_word_size`

Definition at line 28 of file eHSM_Mailbox_Ip.h.

3.215.2.8 surpport_asyn

`ehsm_bool_t mailbox_channel::surpport_asyn`

Definition at line 26 of file eHSM_Mailbox_Ip.h.

3.215.2.9 type

`mailbox_channel_e mailbox_channel::type`

Definition at line 25 of file eHSM_Mailbox_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Mailbox_Ip.h](#)

3.216 ehsm_cmd_cipher_st::osr_cmd_hdr_u Union Reference

```
#include <eHSM_Mailbox_Prtcl_Ip.h>
```

Public Attributes

- [ehsm_cmd_hdr_ske_st](#) `hdr_ske`
- [ehsm_cmd_hdr_pke_st](#) `hdr_pke`
- [ehsm_cmd_hdr_ecise_st](#) `hdr_ecise`
- [ehsm_cmd_hdr_sm9_st](#) `hdr_sm9`
- [ehsm_cmd_hdr_eccp_keygen_st](#) `hdr_eccp_keygen`
- [ehsm_cmd_hdr_rsa_keygen_st](#) `hdr_rsa_keygen`
- [ehsm_cmd_hdr_rng_st](#) `hdr_rng`

3.216.1 Detailed Description

Definition at line 886 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2 Member Data Documentation

3.216.2.1 hdr_eccp_keygen

[ehsm_cmd_hdr_eccp_keygen_st](#) ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_eccp_keygen

Definition at line 892 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2.2 hdr_ecise

[ehsm_cmd_hdr_ecise_st](#) ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_ecise

Definition at line 890 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2.3 hdr_pke

[ehsm_cmd_hdr_pke_st](#) ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_pke

Definition at line 889 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2.4 hdr_rng

[ehsm_cmd_hdr_rng_st](#) ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_rng

Definition at line 894 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2.5 hdr_rsa_keygen

[ehsm_cmd_hdr_rsa_keygen_st](#) ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_rsa_keygen

Definition at line 893 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2.6 hdr_ske

[ehsm_cmd_hdr_ske_st](#) ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_ske

Definition at line 888 of file eHSM_Mailbox_Prtcl_Ip.h.

3.216.2.7 `hdr_sm9`

`ehsm_cmd_hdr_sm9_st` `ehsm_cmd_cipher_st::osr_cmd_hdr_u::hdr_sm9`

Definition at line 891 of file `eHSM_Mailbox_Prtcl_Ip.h`.

The documentation for this union was generated from the following file:

- [eHSM_Mailbox_Prtcl_Ip.h](#)

3.217 rsacipher_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- `const unsigned char * n`
- `const unsigned char * e`
- `const unsigned char * d`
- `const unsigned char * p`
- `const unsigned char * q`
- `const unsigned char * dp`
- `const unsigned char * dq`
- `const unsigned char * u`
- `unsigned int n_byte_size`
- `unsigned int e_byte_size`
- `unsigned int d_byte_size`
- `unsigned int p_byte_size`
- `unsigned int q_byte_size`
- `unsigned int dp_byte_size`
- `unsigned int dq_byte_size`
- `unsigned int u_byte_size`
- `const unsigned char * m`
- `unsigned int m_size`
- `const unsigned char * c`
- `unsigned int c_size`
- `unsigned int is crt_mode`
- `unsigned char public_key_vec`
- `unsigned char siggen_sigver_test`
- `unsigned char hash_alg`
- `const unsigned char * salt`
- `unsigned int salt_byte_size`

3.217.1 Detailed Description

Definition at line 181 of file `utest_vecs_st.h`.

3.217.2 Member Data Documentation

3.217.2.1 c

```
const unsigned char* rsacipher_testvec::c
```

Definition at line 200 of file `utest_vecs_st.h`.

3.217.2.2 c_size

```
unsigned int rsacipher_testvec::c_size
```

Definition at line 201 of file `utest_vecs_st.h`.

3.217.2.3 d

```
const unsigned char* rsacipher_testvec::d
```

Definition at line 184 of file `utest_vecs_st.h`.

3.217.2.4 d_byte_size

```
unsigned int rsacipher_testvec::d_byte_size
```

Definition at line 192 of file `utest_vecs_st.h`.

3.217.2.5 dp

```
const unsigned char* rsacipher_testvec::dp
```

Definition at line 187 of file `utest_vecs_st.h`.

3.217.2.6 dp_byte_size

```
unsigned int rsacipher_testvec::dp_byte_size
```

Definition at line 195 of file `utest_vecs_st.h`.

3.217.2.7 dq

```
const unsigned char* rsacipher_testvec::dq
```

Definition at line 188 of file `utest_vecs_st.h`.

3.217.2.8 dq_byte_size

```
unsigned int rsacipher_testvec::dq_byte_size
```

Definition at line 196 of file `utest_vecs_st.h`.

3.217.2.9 e

```
const unsigned char* rsacipher_testvec::e
```

Definition at line 183 of file `utest_vecs_st.h`.

3.217.2.10 e_byte_size

```
unsigned int rsacipher_testvec::e_byte_size
```

Definition at line 191 of file `utest_vecs_st.h`.

3.217.2.11 hash_alg

```
unsigned char rsacipher_testvec::hash_alg
```

Definition at line 205 of file `utest_vecs_st.h`.

3.217.2.12 is crt_mode

```
unsigned int rsacipher_testvec::is crt_mode
```

Definition at line 202 of file `utest_vecs_st.h`.

3.217.2.13 m

```
const unsigned char* rsacipher_testvec::m
```

Definition at line 198 of file `utest_vecs_st.h`.

3.217.2.14 m_size

```
unsigned int rsacipher_testvec::m_size
```

Definition at line 199 of file `utest_vecs_st.h`.

3.217.2.15 n

```
const unsigned char* rsacipher_testvec::n
```

Definition at line 182 of file `utest_vecs_st.h`.

3.217.2.16 n_byte_size

```
unsigned int rsacipher_testvec::n_byte_size
```

Definition at line 190 of file `utest_vecs_st.h`.

3.217.2.17 p

```
const unsigned char* rsacipher_testvec::p
```

Definition at line 185 of file `utest_vecs_st.h`.

3.217.2.18 p_byte_size

```
unsigned int rsacipher_testvec::p_byte_size
```

Definition at line 193 of file `utest_vecs_st.h`.

3.217.2.19 public_key_vec

```
unsigned char rsacipher_testvec::public_key_vec
```

Definition at line 203 of file `utest_vecs_st.h`.

3.217.2.20 q

```
const unsigned char* rsacipher_testvec::q
```

Definition at line 186 of file `utest_vecs_st.h`.

3.217.2.21 q_byte_size

```
unsigned int rsacipher_testvec::q_byte_size
```

Definition at line 194 of file `utest_vecs_st.h`.

3.217.2.22 salt

```
const unsigned char* rsacipher_testvec::salt
```

Definition at line 208 of file `utest_vecs_st.h`.

3.217.2.23 salt_byte_size

```
unsigned int rsacipher_testvec::salt_byte_size
```

Definition at line 209 of file `utest_vecs_st.h`.

3.217.2.24 siggen_sigver_test

```
unsigned char rsacipher_testvec::siggen_sigver_test
```

Definition at line 204 of file `utest_vecs_st.h`.

3.217.2.25 u

```
const unsigned char* rsacipher_testvec::u
```

Definition at line 189 of file `utest_vecs_st.h`.

3.217.2.26 u_byte_size

```
unsigned int rsacipher_testvec::u_byte_size
```

Definition at line 197 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.218 signature_t Struct Reference

```
#include <eHSM_If_Evita_Types_Ip.h>
```

Public Attributes

- [ehsm_uint8_t signature](#) [EVITA_SIGNATURE_BUF_SIZE]
- [ehsm_uint32_t signature_size](#)
- [ehsm_utc_time_t utc_time](#)

3.218.1 Detailed Description

Definition at line 432 of file `eHSM_If_Evita_Types_Ip.h`.

3.218.2 Member Data Documentation

3.218.2.1 signature

```
ehsm\_uint8\_t signature_t::signature [EVITA\_SIGNATURE\_BUF\_SIZE]
```

Definition at line 433 of file `eHSM_If_Evita_Types_Ip.h`.

3.218.2.2 signature_size

`ehsm_uint32_t signature_t::signature_size`

Definition at line 434 of file `eHSM_If_Evita_Types_Ip.h`.

3.218.2.3 utc_time

`ehsm_utc_time_t signature_t::utc_time`

Definition at line 435 of file `eHSM_If_Evita_Types_Ip.h`.

The documentation for this struct was generated from the following file:

- [eHSM_If_Evita_Types_Ip.h](#)

3.219 sm2_ext_param Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t sm2_role](#)
- [ehsm_uint8_t * peer_pubkey](#)
- [ehsm_uint8_t * peer_temp_pubkey](#)
- [ehsm_uint8_t * s1_s2_value](#)
- [ehsm_uint8_t * sa_sb_value](#)
- [ehsm_uint32_t local_tmp_key_handle](#)
- [ehsm_uint32_t local_tmp_key_auth_size](#)
- [ehsm_uint8_t * local_tmp_key_auth_value](#)

3.219.1 Detailed Description

Definition at line 324 of file `eHSM_Com_Struct_Ip.h`.

3.219.2 Member Data Documentation

3.219.2.1 local_tmp_key_auth_size

`ehsm_uint32_t sm2_ext_param::local_tmp_key_auth_size`

Definition at line 332 of file `eHSM_Com_Struct_Ip.h`.

3.219.2.2 local_tmp_key_auth_value

`ehsm_uint8_t*` sm2_ext_param::local_tmp_key_auth_value

Definition at line 333 of file eHSM_Com_Struct_lp.h.

3.219.2.3 local_tmp_key_handle

`ehsm_uint32_t` sm2_ext_param::local_tmp_key_handle

Definition at line 331 of file eHSM_Com_Struct_lp.h.

3.219.2.4 peer_pubkey

`ehsm_uint8_t*` sm2_ext_param::peer_pubkey

Definition at line 327 of file eHSM_Com_Struct_lp.h.

3.219.2.5 peer_temp_pubkey

`ehsm_uint8_t*` sm2_ext_param::peer_temp_pubkey

Definition at line 328 of file eHSM_Com_Struct_lp.h.

3.219.2.6 s1_s2_value

`ehsm_uint8_t*` sm2_ext_param::s1_s2_value

Definition at line 329 of file eHSM_Com_Struct_lp.h.

3.219.2.7 sa_sb_value

`ehsm_uint8_t*` sm2_ext_param::sa_sb_value

Definition at line 330 of file eHSM_Com_Struct_lp.h.

3.219.2.8 sm2_role

```
ehsm_uint8_t sm2_ext_param::sm2_role
```

Definition at line 326 of file eHSM_Com_Struct_lp.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_lp.h](#)

3.220 sm9cipher_testvec Struct Reference

```
#include <utest_vecs_st.h>
```

Public Attributes

- unsigned char [hid](#)
- const unsigned char * [Ppub](#)
- const unsigned char * [priv](#)
- const unsigned char * [id](#)
- unsigned int [id_sz](#)
- const unsigned char * [m](#)
- unsigned int [m_sz](#)
- const unsigned char * [r](#)
- unsigned int [r_sz](#)
- const unsigned char * [c](#)
- unsigned int [c_sz](#)
- const unsigned char * [h](#)
- const unsigned char * [sig](#)
- unsigned char [enc_type](#)
- unsigned int [K2len](#)
- unsigned char [padding](#)
- unsigned char [sigen_sigver_test](#)

3.220.1 Detailed Description

Definition at line 212 of file utest_vecs_st.h.

3.220.2 Member Data Documentation

3.220.2.1 c

```
const unsigned char* sm9cipher_testvec::c
```

Definition at line 222 of file utest_vecs_st.h.

3.220.2.2 c_sz

```
unsigned int sm9cipher_testvec::c_sz
```

Definition at line 223 of file utest_vecs_st.h.

3.220.2.3 enc_typde

```
unsigned char sm9cipher_testvec::enc_typde
```

Definition at line 226 of file utest_vecs_st.h.

3.220.2.4 h

```
const unsigned char* sm9cipher_testvec::h
```

Definition at line 224 of file utest_vecs_st.h.

3.220.2.5 hid

```
unsigned char sm9cipher_testvec::hid
```

Definition at line 213 of file utest_vecs_st.h.

3.220.2.6 id

```
const unsigned char* sm9cipher_testvec::id
```

Definition at line 216 of file utest_vecs_st.h.

3.220.2.7 id_sz

```
unsigned int sm9cipher_testvec::id_sz
```

Definition at line 217 of file utest_vecs_st.h.

3.220.2.8 K2len

```
unsigned int sm9cipher_testvec::K2len
```

Definition at line 227 of file `utest_vecs_st.h`.

3.220.2.9 m

```
const unsigned char* sm9cipher_testvec::m
```

Definition at line 218 of file `utest_vecs_st.h`.

3.220.2.10 m_sz

```
unsigned int sm9cipher_testvec::m_sz
```

Definition at line 219 of file `utest_vecs_st.h`.

3.220.2.11 padding

```
unsigned char sm9cipher_testvec::padding
```

Definition at line 228 of file `utest_vecs_st.h`.

3.220.2.12 Ppub

```
const unsigned char* sm9cipher_testvec::Ppub
```

Definition at line 214 of file `utest_vecs_st.h`.

3.220.2.13 priv

```
const unsigned char* sm9cipher_testvec::priv
```

Definition at line 215 of file `utest_vecs_st.h`.

3.220.2.14 r

```
const unsigned char* sm9cipher_testvec::r
```

Definition at line 220 of file `utest_vecs_st.h`.

3.220.2.15 r_sz

```
unsigned int sm9cipher_testvec::r_sz
```

Definition at line 221 of file `utest_vecs_st.h`.

3.220.2.16 sig

```
const unsigned char* sm9cipher_testvec::sig
```

Definition at line 225 of file `utest_vecs_st.h`.

3.220.2.17 siggen_sigver_test

```
unsigned char sm9cipher_testvec::siggen_sigver_test
```

Definition at line 229 of file `utest_vecs_st.h`.

The documentation for this struct was generated from the following file:

- [utest_vecs_st.h](#)

3.221 soc_image_upgrade_info Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- ehsm_uint8_t * cmd_input
- ehsm_uint8_t storage_alg
- ehsm_uint8_t upgrade_alg
- ehsm_uint8_t storage_encryption_flag
- ehsm_uint8_t upgrade_decryption_flag
- ehsm_uint8_t process_mode
- ehsm_uint8_t check_version_flag
- ehsm_uint8_t * upgrade_sign
- ehsm_uint32_t upgrade_sign_size
- ehsm_uint8_t * upgrade_version
- ehsm_uint32_t upgrade_version_size
- ehsm_uint8_t * upgrade_image
- ehsm_uint32_t upgrade_image_size
- ehsm_uint8_t * storage_image
- ehsm_uint32_t storage_image_size
- ehsm_uint8_t * mac_sign
- ehsm_uint32_t mac_sign_size
- ehsm_uint8_t * upgrade_pubkey
- ehsm_uint32_t upgrade_pubkey_size
- ehsm_uint8_t * upgrade_iv
- ehsm_uint32_t upgrade_iv_size
- ehsm_uint8_t * storage_iv
- ehsm_uint32_t storage_iv_size
- ehsm_uint8_t * header
- ehsm_uint32_t header_size

3.221.1 Detailed Description

Definition at line 624 of file eHSM_Com_Struct_lp.h.

3.221.2 Member Data Documentation

3.221.2.1 check_version_flag

```
ehsm_uint8_t soc_image_upgrade_info::check_version_flag
```

Definition at line 632 of file eHSM_Com_Struct_lp.h.

3.221.2.2 cmd_input

```
ehsm_uint8_t* soc_image_upgrade_info::cmd_input
```

Definition at line 626 of file eHSM_Com_Struct_lp.h.

3.221.2.3 header

```
ehsm_uint8_t* soc_image_upgrade_info::header
```

Definition at line 649 of file eHSM_Com_Struct_lp.h.

3.221.2.4 header_size

```
ehsm_uint32_t soc_image_upgrade_info::header_size
```

Definition at line 650 of file eHSM_Com_Struct_lp.h.

3.221.2.5 mac_sign

```
ehsm_uint8_t* soc_image_upgrade_info::mac_sign
```

Definition at line 641 of file eHSM_Com_Struct_lp.h.

3.221.2.6 mac_sign_size

```
ehsm_uint32_t soc_image_upgrade_info::mac_sign_size
```

Definition at line 642 of file eHSM_Com_Struct_lp.h.

3.221.2.7 process_mode

```
ehsm_uint8_t soc_image_upgrade_info::process_mode
```

Definition at line 631 of file eHSM_Com_Struct_lp.h.

3.221.2.8 storage_alg

```
ehsm_uint8_t soc_image_upgrade_info::storage_alg
```

Definition at line 627 of file eHSM_Com_Struct_lp.h.

3.221.2.9 storage_encryption_flag

`ehsm_uint8_t soc_image_upgrade_info::storage_encryption_flag`

Definition at line 629 of file eHSM_Com_Struct_lp.h.

3.221.2.10 storage_image

`ehsm_uint8_t* soc_image_upgrade_info::storage_image`

Definition at line 639 of file eHSM_Com_Struct_lp.h.

3.221.2.11 storage_image_size

`ehsm_uint32_t soc_image_upgrade_info::storage_image_size`

Definition at line 640 of file eHSM_Com_Struct_lp.h.

3.221.2.12 storage_iv

`ehsm_uint8_t* soc_image_upgrade_info::storage_iv`

Definition at line 647 of file eHSM_Com_Struct_lp.h.

3.221.2.13 storage_iv_size

`ehsm_uint32_t soc_image_upgrade_info::storage_iv_size`

Definition at line 648 of file eHSM_Com_Struct_lp.h.

3.221.2.14 upgrade_alg

`ehsm_uint8_t soc_image_upgrade_info::upgrade_alg`

Definition at line 628 of file eHSM_Com_Struct_lp.h.

3.221.2.15 upgrade_decryption_flag

`ehsm_uint8_t soc_image_upgrade_info::upgrade_decryption_flag`

Definition at line 630 of file eHSM_Com_Struct_lp.h.

3.221.2.16 upgrade_image

`ehsm_uint8_t* soc_image_upgrade_info::upgrade_image`

Definition at line 637 of file eHSM_Com_Struct_lp.h.

3.221.2.17 upgrade_image_size

`ehsm_uint32_t soc_image_upgrade_info::upgrade_image_size`

Definition at line 638 of file eHSM_Com_Struct_lp.h.

3.221.2.18 upgrade_iv

`ehsm_uint8_t* soc_image_upgrade_info::upgrade_iv`

Definition at line 645 of file eHSM_Com_Struct_lp.h.

3.221.2.19 upgrade_iv_size

`ehsm_uint32_t soc_image_upgrade_info::upgrade_iv_size`

Definition at line 646 of file eHSM_Com_Struct_lp.h.

3.221.2.20 upgrade_pubkey

`ehsm_uint8_t* soc_image_upgrade_info::upgrade_pubkey`

Definition at line 643 of file eHSM_Com_Struct_lp.h.

3.221.2.21 upgrade_pubkey_size

```
ehsm_uint32_t soc_image_upgrade_info::upgrade_pubkey_size
```

Definition at line 644 of file eHSM_Com_Struct_Ip.h.

3.221.2.22 upgrade_sign

```
ehsm_uint8_t* soc_image_upgrade_info::upgrade_sign
```

Definition at line 633 of file eHSM_Com_Struct_Ip.h.

3.221.2.23 upgrade_sign_size

```
ehsm_uint32_t soc_image_upgrade_info::upgrade_sign_size
```

Definition at line 634 of file eHSM_Com_Struct_Ip.h.

3.221.2.24 upgrade_version

```
ehsm_uint8_t* soc_image_upgrade_info::upgrade_version
```

Definition at line 635 of file eHSM_Com_Struct_Ip.h.

3.221.2.25 upgrade_version_size

```
ehsm_uint32_t soc_image_upgrade_info::upgrade_version_size
```

Definition at line 636 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.222 soc_image_upgrade_input Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t storage_alg](#)
- [ehsm_uint8_t upgrade_alg](#)
- [ehsm_uint8_t storage_encryption_flag](#)
- [ehsm_uint8_t upgrade_decryption_flag](#)
- [ehsm_uint8_t process_mode](#)
- [ehsm_uint8_t check_version_flag](#)
- [ehsm_uint8_t rev](#) [2]
- [ehsm_uint8_t upgrade_sign_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t upgrade_sign_size](#) [4]
- [ehsm_uint8_t upgrade_version_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t upgrade_version_size](#) [4]
- [ehsm_uint8_t upgrade_image_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t upgrade_image_size](#) [4]
- [ehsm_uint8_t storage_image_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t storage_image_size](#) [4]
- [ehsm_uint8_t mac_sign_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t mac_sign_size](#) [4]
- [ehsm_uint8_t upgrade_pubkey_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t upgrade_pubkey_size](#) [4]
- [ehsm_uint8_t upgrade_iv_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t upgrade_iv_size](#) [4]
- [ehsm_uint8_t storage_iv_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t storage_iv_size](#) [4]
- [ehsm_uint8_t header_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t header_size](#) [4]
- [ehsm_uint8_t ctx_addr](#) [HOST_ADDRESS_SIZE]
- [ehsm_uint8_t ctx_size](#) [4]

3.222.1 Detailed Description

Definition at line 593 of file eHSM_Com_Struct_lp.h.

3.222.2 Member Data Documentation

3.222.2.1 check_version_flag

```
ehsm_uint8_t soc_image_upgrade_input::check_version_flag
```

Definition at line 600 of file eHSM_Com_Struct_lp.h.

3.222.2.2 ctx_addr

```
ehsm_uint8_t soc_image_upgrade_input::ctx_addr[HOST_ADDRESS_SIZE]
```

Definition at line 620 of file eHSM_Com_Struct_lp.h.

3.222.2.3 ctx_size

```
ehsm_uint8_t soc_image_upgrade_input::ctx_size[4]
```

Definition at line 621 of file eHSM_Com_Struct_lp.h.

3.222.2.4 header_addr

```
ehsm_uint8_t soc_image_upgrade_input::header_addr[HOST_ADDRESS_SIZE]
```

Definition at line 618 of file eHSM_Com_Struct_lp.h.

3.222.2.5 header_size

```
ehsm_uint8_t soc_image_upgrade_input::header_size[4]
```

Definition at line 619 of file eHSM_Com_Struct_lp.h.

3.222.2.6 mac_sign_addr

```
ehsm_uint8_t soc_image_upgrade_input::mac_sign_addr[HOST_ADDRESS_SIZE]
```

Definition at line 610 of file eHSM_Com_Struct_lp.h.

3.222.2.7 mac_sign_size

```
ehsm_uint8_t soc_image_upgrade_input::mac_sign_size[4]
```

Definition at line 611 of file eHSM_Com_Struct_lp.h.

3.222.2.8 process_mode

```
ehsm_uint8_t soc_image_upgrade_input::process_mode
```

Definition at line 599 of file eHSM_Com_Struct_lp.h.

3.222.2.9 rev

```
ehsm_uint8_t soc_image_upgrade_input::rev[2]
```

Definition at line 601 of file eHSM_Com_Struct_lp.h.

3.222.2.10 storage_alg

```
ehsm_uint8_t soc_image_upgrade_input::storage_alg
```

Definition at line 595 of file eHSM_Com_Struct_lp.h.

3.222.2.11 storage_encryption_flag

```
ehsm_uint8_t soc_image_upgrade_input::storage_encryption_flag
```

Definition at line 597 of file eHSM_Com_Struct_lp.h.

3.222.2.12 storage_image_addr

```
ehsm_uint8_t soc_image_upgrade_input::storage_image_addr[HOST_ADDRESS_SIZE]
```

Definition at line 608 of file eHSM_Com_Struct_lp.h.

3.222.2.13 storage_image_size

```
ehsm_uint8_t soc_image_upgrade_input::storage_image_size[4]
```

Definition at line 609 of file eHSM_Com_Struct_lp.h.

3.222.2.14 storage_iv_addr

```
ehsm_uint8_t soc_image_upgrade_input::storage_iv_addr[HOST_ADDRESS_SIZE]
```

Definition at line 616 of file eHSM_Com_Struct_lp.h.

3.222.2.15 storage_iv_size

```
ehsm_uint8_t soc_image_upgrade_input::storage_iv_size[4]
```

Definition at line 617 of file eHSM_Com_Struct_lp.h.

3.222.2.16 upgrade_alg

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_alg
```

Definition at line 596 of file eHSM_Com_Struct_lp.h.

3.222.2.17 upgrade_decryption_flag

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_decryption_flag
```

Definition at line 598 of file eHSM_Com_Struct_lp.h.

3.222.2.18 upgrade_image_addr

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_image_addr[HOST_ADDRESS_SIZE]
```

Definition at line 606 of file eHSM_Com_Struct_lp.h.

3.222.2.19 upgrade_image_size

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_image_size[4]
```

Definition at line 607 of file eHSM_Com_Struct_lp.h.

3.222.2.20 upgrade_iv_addr

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_iv_addr[HOST_ADDRESS_SIZE]
```

Definition at line 614 of file eHSM_Com_Struct_lp.h.

3.222.2.21 upgrade_iv_size

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_iv_size[4]
```

Definition at line 615 of file eHSM_Com_Struct_lp.h.

3.222.2.22 upgrade_pubkey_addr

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_pubkey_addr[HOST_ADDRESS_SIZE]
```

Definition at line 612 of file eHSM_Com_Struct_lp.h.

3.222.2.23 upgrade_pubkey_size

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_pubkey_size[4]
```

Definition at line 613 of file eHSM_Com_Struct_lp.h.

3.222.2.24 upgrade_sign_addr

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_sign_addr[HOST_ADDRESS_SIZE]
```

Definition at line 602 of file eHSM_Com_Struct_lp.h.

3.222.2.25 upgrade_sign_size

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_sign_size[4]
```

Definition at line 603 of file eHSM_Com_Struct_lp.h.

3.222.2.26 upgrade_version_addr

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_version_addr[HOST_ADDRESS_SIZE]
```

Definition at line 604 of file eHSM_Com_Struct_lp.h.

3.222.2.27 upgrade_version_size

```
ehsm_uint8_t soc_image_upgrade_input::upgrade_version_size[4]
```

Definition at line 605 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.223 soc_image_verify_info Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t](#) * cmd_input
- [ehsm_uint8_t](#) update_version_flag
- [ehsm_uint8_t](#) type
- [ehsm_uint8_t](#) storage_alg
- [ehsm_uint8_t](#) storage_encryption_flag
- [ehsm_uint8_t](#) * version
- [ehsm_uint32_t](#) version_size
- [ehsm_uint8_t](#) * pubkey
- [ehsm_uint32_t](#) pubkey_size
- [ehsm_uint8_t](#) * storage_image
- [ehsm_uint32_t](#) storage_image_size
- [ehsm_uint8_t](#) * header
- [ehsm_uint32_t](#) header_size
- [ehsm_uint8_t](#) * storage_iv
- [ehsm_uint32_t](#) storage_iv_size
- [ehsm_uint8_t](#) * storage_sign
- [ehsm_uint32_t](#) storage_sign_size

3.223.1 Detailed Description

Definition at line 691 of file eHSM_Com_Struct_Ip.h.

3.223.2 Member Data Documentation

3.223.2.1 cmd_input

```
ehsm_uint8_t* soc_image_verify_info::cmd_input
```

Definition at line 693 of file eHSM_Com_Struct_Ip.h.

3.223.2.2 header

```
ehsm_uint8_t* soc_image_verify_info::header
```

Definition at line 704 of file eHSM_Com_Struct_lp.h.

3.223.2.3 header_size

```
ehsm_uint32_t soc_image_verify_info::header_size
```

Definition at line 705 of file eHSM_Com_Struct_lp.h.

3.223.2.4 pubkey

```
ehsm_uint8_t* soc_image_verify_info::pubkey
```

Definition at line 700 of file eHSM_Com_Struct_lp.h.

3.223.2.5 pubkey_size

```
ehsm_uint32_t soc_image_verify_info::pubkey_size
```

Definition at line 701 of file eHSM_Com_Struct_lp.h.

3.223.2.6 storage_alg

```
ehsm_uint8_t soc_image_verify_info::storage_alg
```

Definition at line 696 of file eHSM_Com_Struct_lp.h.

3.223.2.7 storage_encryption_flag

```
ehsm_uint8_t soc_image_verify_info::storage_encryption_flag
```

Definition at line 697 of file eHSM_Com_Struct_lp.h.

3.223.2.8 storage_image

`ehsm_uint8_t* soc_image_verify_info::storage_image`

Definition at line 702 of file eHSM_Com_Struct_lp.h.

3.223.2.9 storage_image_size

`ehsm_uint32_t soc_image_verify_info::storage_image_size`

Definition at line 703 of file eHSM_Com_Struct_lp.h.

3.223.2.10 storage_iv

`ehsm_uint8_t* soc_image_verify_info::storage_iv`

Definition at line 706 of file eHSM_Com_Struct_lp.h.

3.223.2.11 storage_iv_size

`ehsm_uint32_t soc_image_verify_info::storage_iv_size`

Definition at line 707 of file eHSM_Com_Struct_lp.h.

3.223.2.12 storage_sign

`ehsm_uint8_t* soc_image_verify_info::storage_sign`

Definition at line 708 of file eHSM_Com_Struct_lp.h.

3.223.2.13 storage_sign_size

`ehsm_uint32_t soc_image_verify_info::storage_sign_size`

Definition at line 709 of file eHSM_Com_Struct_lp.h.

3.223.2.14 type

```
ehsm_uint8_t soc_image_verify_info::type
```

Definition at line 695 of file eHSM_Com_Struct_Ip.h.

3.223.2.15 update_version_flag

```
ehsm_uint8_t soc_image_verify_info::update_version_flag
```

Definition at line 694 of file eHSM_Com_Struct_Ip.h.

3.223.2.16 version

```
ehsm_uint8_t* soc_image_verify_info::version
```

Definition at line 698 of file eHSM_Com_Struct_Ip.h.

3.223.2.17 version_size

```
ehsm_uint32_t soc_image_verify_info::version_size
```

Definition at line 699 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

3.224 soc_image_verify_input Struct Reference

```
#include <eHSM_Com_Struct_Ip.h>
```

Public Attributes

- [ehsm_uint8_t update_version_flag](#)
- [ehsm_uint8_t type](#)
- [ehsm_uint8_t storage_alg](#)
- [ehsm_uint8_t storage_encryption_flag](#)
- [ehsm_uint8_t version_addr \[HOST_ADDRESS_SIZE\]](#)
- [ehsm_uint8_t version_size \[4\]](#)
- [ehsm_uint8_t pubkey_addr \[HOST_ADDRESS_SIZE\]](#)
- [ehsm_uint8_t pubkey_size \[4\]](#)
- [ehsm_uint8_t storage_image_addr \[HOST_ADDRESS_SIZE\]](#)
- [ehsm_uint8_t storage_image_size \[4\]](#)
- [ehsm_uint8_t header_addr \[HOST_ADDRESS_SIZE\]](#)
- [ehsm_uint8_t header_size \[4\]](#)
- [ehsm_uint8_t storage_iv_addr \[HOST_ADDRESS_SIZE\]](#)
- [ehsm_uint8_t storage_iv_size \[4\]](#)
- [ehsm_uint8_t storage_sign_addr \[HOST_ADDRESS_SIZE\]](#)
- [ehsm_uint8_t storage_sign_size \[4\]](#)

3.224.1 Detailed Description

Definition at line 671 of file eHSM_Com_Struct_lp.h.

3.224.2 Member Data Documentation

3.224.2.1 header_addr

```
ehsm_uint8_t soc_image_verify_input::header_addr[HOST_ADDRESS_SIZE]
```

Definition at line 683 of file eHSM_Com_Struct_lp.h.

3.224.2.2 header_size

```
ehsm_uint8_t soc_image_verify_input::header_size[4]
```

Definition at line 684 of file eHSM_Com_Struct_lp.h.

3.224.2.3 pubkey_addr

```
ehsm_uint8_t soc_image_verify_input::pubkey_addr[HOST_ADDRESS_SIZE]
```

Definition at line 679 of file eHSM_Com_Struct_lp.h.

3.224.2.4 pubkey_size

```
ehsm_uint8_t soc_image_verify_input::pubkey_size[4]
```

Definition at line 680 of file eHSM_Com_Struct_lp.h.

3.224.2.5 storage_alg

```
ehsm_uint8_t soc_image_verify_input::storage_alg
```

Definition at line 675 of file eHSM_Com_Struct_lp.h.

3.224.2.6 storage_encryption_flag

```
ehsm_uint8_t soc_image_verify_input::storage_encryption_flag
```

Definition at line 676 of file eHSM_Com_Struct_lp.h.

3.224.2.7 storage_image_addr

```
ehsm_uint8_t soc_image_verify_input::storage_image_addr[HOST_ADDRESS_SIZE]
```

Definition at line 681 of file eHSM_Com_Struct_lp.h.

3.224.2.8 storage_image_size

```
ehsm_uint8_t soc_image_verify_input::storage_image_size[4]
```

Definition at line 682 of file eHSM_Com_Struct_lp.h.

3.224.2.9 storage_iv_addr

```
ehsm_uint8_t soc_image_verify_input::storage_iv_addr[HOST_ADDRESS_SIZE]
```

Definition at line 685 of file eHSM_Com_Struct_lp.h.

3.224.2.10 storage_iv_size

```
ehsm_uint8_t soc_image_verify_input::storage_iv_size[4]
```

Definition at line 686 of file eHSM_Com_Struct_lp.h.

3.224.2.11 storage_sign_addr

```
ehsm_uint8_t soc_image_verify_input::storage_sign_addr[HOST_ADDRESS_SIZE]
```

Definition at line 687 of file eHSM_Com_Struct_lp.h.

3.224.2.12 storage_sign_size

```
ehsm_uint8_t soc_image_verify_input::storage_sign_size[4]
```

Definition at line 688 of file eHSM_Com_Struct_Ip.h.

3.224.2.13 type

```
ehsm_uint8_t soc_image_verify_input::type
```

Definition at line 674 of file eHSM_Com_Struct_Ip.h.

3.224.2.14 update_version_flag

```
ehsm_uint8_t soc_image_verify_input::update_version_flag
```

Definition at line 673 of file eHSM_Com_Struct_Ip.h.

3.224.2.15 version_addr

```
ehsm_uint8_t soc_image_verify_input::version_addr[HOST_ADDRESS_SIZE]
```

Definition at line 677 of file eHSM_Com_Struct_Ip.h.

3.224.2.16 version_size

```
ehsm_uint8_t soc_image_verify_input::version_size[4]
```

Definition at line 678 of file eHSM_Com_Struct_Ip.h.

The documentation for this struct was generated from the following file:

- [eHSM_Com_Struct_Ip.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_HSM.pdf File Reference

4.2 AC784xx_Hsm_Reg.h File Reference

This file provides HSM hardware integration functions.

```
#include "Device_Register.h"
#include "eHSM_Mailbox_Reg_Ip.h"
```

Macros

- `#define OTP_BASE_ADDR (0x01540000UL)`
- `#define OTP_LIFE_CYCLE_ADDR (OTP_BASE_ADDR + 0x00UL)`
- `#define OTP_UID_ADDR (OTP_BASE_ADDR + 0x04UL)`
- `#define OTP_HW_CTRL_FIELD_ADDR (OTP_BASE_ADDR + 0x18UL)`
- `#define OTP_FW_CTRL_FIELD_ADDR_L (OTP_BASE_ADDR + 0x20UL)`
- `#define OTP_FW_CTRL_FIELD_ADDR_H (OTP_BASE_ADDR + 0x24UL)`
- `#define OTP_HOST_CTRL_FIELD_ADDR_L (OTP_BASE_ADDR + 0x28UL)`
- `#define OTP_HOST_CTRL_FIELD_ADDR_H (OTP_BASE_ADDR + 0x2CUL)`
- `#define OTP_ERR_RSP_CTRL_ADDR (OTP_BASE_ADDR + 0x30UL)`
- `#define OTP_HSM_VERSION_ADDR (OTP_BASE_ADDR + 0x50UL)`
- `#define OTP_SOC_VERSION_ADDR (OTP_BASE_ADDR + 0x60UL)`
- `#define OTP_KEY_ADDR(a) (OTP_BASE_ADDR + 0x70UL + ((a)*10U*4UL))`
- `#define OTP_KEY_CRC(a) (OTP_BASE_ADDR + 0x94UL + ((a)*10U*4UL))`
- `#define OTP_HSM_ENABLE_ADDR (OTP_BASE_ADDR + 0x570UL)`
- `#define OTP_SECURE_BOOT_ADDR (OTP_BASE_ADDR + 0x574UL)`
- `#define OTP_KEY_ATTR_ENCODE_EACH_LENGTH (CONFIG_OTP_KEY_ATTR_LENGTH)`
- `#define OTP_KEY_ATTR_BYTE_LENGTH (4UL)`
- `#define OTP_VERSION_ENCODE_LENGTH (CONFIG_OTP_VERSION_LENGTH)`
- `#define OTP_VERSION_LENGTH (16UL)`
- `#define OTP_KEY_CRC_SIZE (4UL)`
- `#define OTP_KEY_SIZE (32UL + OTP_KEY_CRC_SIZE)`
- `#define OTP_KEY_ATTR_LC (OTP_BASE_ADDR + 0x600UL)`
- `#define OTP_SIZE (0xC00UL)`
- `#define SOC_CMD_IMAGE_UPGRADE_UPGRADE (0x00DFFF20U)`
- `#define SOC_CMD_IMAGE_VERIFY (0x00DDFF22U)`
- `#define SOC_CMD_GET_HSM_FW_VERSION (0xAC784301U)`
- `#define HSM_CMD_GET_RANDOM_KEY (0x00F7FF08U)`
- `bootloader cmd.`
- `#define HSM_CMD_ENCRYPT_KEY (0x00F6FF09U)`
- `#define HSM_CMD_BOOTROM_SET_BAUDRATE (0x00F5FF0AU)`

4.2.1 Detailed Description

This file provides HSM hardware integration functions.

4.2.2 Macro Definition Documentation

4.2.2.1 HSM_CMD_BOOTROM_SET_BAUDRATE

```
#define HSM_CMD_BOOTROM_SET_BAUDRATE (0x00F5FF0AU)
```

Definition at line 91 of file AC784xx_Hsm_Reg.h.

4.2.2.2 HSM_CMD_ENCRYPT_KEY

```
#define HSM_CMD_ENCRYPT_KEY (0x00F6FF09U)
```

Definition at line 90 of file AC784xx_Hsm_Reg.h.

4.2.2.3 HSM_CMD_GET_RANDOM_KEY

```
#define HSM_CMD_GET_RANDOM_KEY (0x00F7FF08U)
```

bootloader cmd.

Returns

none

Definition at line 89 of file AC784xx_Hsm_Reg.h.

4.2.2.4 OTP_BASE_ADDR

```
#define OTP_BASE_ADDR (0x01540000UL)
```

Definition at line 53 of file AC784xx_Hsm_Reg.h.

4.2.2.5 OTP_ERR_RSP_CTRL_ADDR

```
#define OTP_ERR_RSP_CTRL_ADDR (OTP_BASE_ADDR + 0x30UL)
```

Definition at line 61 of file AC784xx_Hsm_Reg.h.

4.2.2.6 OTP_FW_CTRL_FIELD_ADDR_H

```
#define OTP_FW_CTRL_FIELD_ADDR_H (OTP_BASE_ADDR + 0x24UL)
```

Definition at line 58 of file AC784xx_Hsm_Reg.h.

4.2.2.7 OTP_FW_CTRL_FIELD_ADDR_L

```
#define OTP_FW_CTRL_FIELD_ADDR_L (OTP_BASE_ADDR + 0x20UL)
```

Definition at line 57 of file AC784xx_Hsm_Reg.h.

4.2.2.8 OTP_HOST_CTRL_FIELD_ADDR_H

```
#define OTP_HOST_CTRL_FIELD_ADDR_H (OTP_BASE_ADDR + 0x2CUL)
```

Definition at line 60 of file AC784xx_Hsm_Reg.h.

4.2.2.9 OTP_HOST_CTRL_FIELD_ADDR_L

```
#define OTP_HOST_CTRL_FIELD_ADDR_L (OTP_BASE_ADDR + 0x28UL)
```

Definition at line 59 of file AC784xx_Hsm_Reg.h.

4.2.2.10 OTP_HSM_ENABLE_ADDR

```
#define OTP_HSM_ENABLE_ADDR (OTP_BASE_ADDR + 0x570UL)
```

Definition at line 66 of file AC784xx_Hsm_Reg.h.

4.2.2.11 OTP_HSM_VERSION_ADDR

```
#define OTP_HSM_VERSION_ADDR (OTP_BASE_ADDR + 0x50UL)
```

Definition at line 62 of file AC784xx_Hsm_Reg.h.

4.2.2.12 OTP_HW_CTRL_FIELD_ADDR

```
#define OTP_HW_CTRL_FIELD_ADDR (OTP_BASE_ADDR + 0x18UL)
```

Definition at line 56 of file AC784xx_Hsm_Reg.h.

4.2.2.13 OTP_KEY_ADDR

```
#define OTP_KEY_ADDR(  
    a ) (OTP_BASE_ADDR + 0x70UL + ((a)*10U*4UL))
```

Definition at line 64 of file AC784xx_Hsm_Reg.h.

4.2.2.14 OTP_KEY_ATTR_BYTE_LENGTH

```
#define OTP_KEY_ATTR_BYTE_LENGTH (4UL)
```

Definition at line 70 of file AC784xx_Hsm_Reg.h.

4.2.2.15 OTP_KEY_ATTR_ENCODE_EACH_LENGTH

```
#define OTP_KEY_ATTR_ENCODE_EACH_LENGTH (CONFIG_OTP_KEY_ATTR_LENGTH)
```

Definition at line 69 of file AC784xx_Hsm_Reg.h.

4.2.2.16 OTP_KEY_ATTR_LC

```
#define OTP_KEY_ATTR_LC (OTP_BASE_ADDR + 0x600UL)
```

Definition at line 78 of file AC784xx_Hsm_Reg.h.

4.2.2.17 OTP_KEY_CRC

```
#define OTP_KEY_CRC(  
    a ) (OTP_BASE_ADDR + 0x94UL + ((a)*10U*4UL))
```

Definition at line 65 of file AC784xx_Hsm_Reg.h.

4.2.2.18 OTP_KEY_CRC_SIZE

```
#define OTP_KEY_CRC_SIZE (4UL)
```

Definition at line 75 of file AC784xx_Hsm_Reg.h.

4.2.2.19 OTP_KEY_SIZE

```
#define OTP_KEY_SIZE (32UL + OTP_KEY_CRC_SIZE)
```

Definition at line 76 of file AC784xx_Hsm_Reg.h.

4.2.2.20 OTP_LIFE_CYCLE_ADDR

```
#define OTP_LIFE_CYCLE_ADDR (OTP_BASE_ADDR + 0x00UL)
```

Definition at line 54 of file AC784xx_Hsm_Reg.h.

4.2.2.21 OTP_SECURE_BOOT_ADDR

```
#define OTP_SECURE_BOOT_ADDR (OTP_BASE_ADDR + 0x574UL)
```

Definition at line 67 of file AC784xx_Hsm_Reg.h.

4.2.2.22 OTP_SIZE

```
#define OTP_SIZE (0xC00UL)
```

Definition at line 79 of file AC784xx_Hsm_Reg.h.

4.2.2.23 OTP_SOC_VERSION_ADDR

```
#define OTP_SOC_VERSION_ADDR (OTP_BASE_ADDR + 0x60UL)
```

Definition at line 63 of file AC784xx_Hsm_Reg.h.

4.2.2.24 OTP_UID_ADDR

```
#define OTP_UID_ADDR (OTP_BASE_ADDR + 0x04UL)
```

Definition at line 55 of file AC784xx_Hsm_Reg.h.

4.2.2.25 OTP_VERSION_ENCODE_LENGTH

```
#define OTP_VERSION_ENCODE_LENGTH (CONFIG_OTP_VERSION_LENGTH)
```

Definition at line 72 of file AC784xx_Hsm_Reg.h.

4.2.2.26 OTP_VERSION_LENGTH

```
#define OTP_VERSION_LENGTH (16UL)
```

Definition at line 73 of file AC784xx_Hsm_Reg.h.

4.2.2.27 SOC_CMD_GET_HSM_FW_VERSION

```
#define SOC_CMD_GET_HSM_FW_VERSION (0xAC784301U)
```

Definition at line 83 of file AC784xx_Hsm_Reg.h.

4.2.2.28 SOC_CMD_IMAGE_UPGRADE_UPGRADE

```
#define SOC_CMD_IMAGE_UPGRADE_UPGRADE (0x00DFFF20U)
```

Definition at line 81 of file AC784xx_Hsm_Reg.h.

4.2.2.29 SOC_CMD_IMAGE_VERIFY

```
#define SOC_CMD_IMAGE_VERIFY (0x00DDFF22U)
```

Definition at line 82 of file AC784xx_Hsm_Reg.h.

4.3 Asr_Standard_Types.h File Reference

```
#include "eHSM_Config_Ip.h"  
#include "eHSM_IntCfg_Ip.h"  
#include "Std_Types.h"
```

Classes

- struct [Crypto_JobInfoType](#)
- struct [Crypto_JobPrimitiveInputOutputType](#)
- struct [Crypto_AlgorithmInfoType](#)
- struct [Crypto_PrimitiveInfoType](#)
- struct [Crypto_JobPrimitiveInfoType](#)
- struct [Crypto_JobRedirectionInfoType](#)
- struct [Crypto_JobType](#)
- struct [CryptoKeyElementType](#)
- struct [CryptoPrimitive](#)
- struct [CryptoDriverObject](#)

Macros

- #define [E_OK](#) 0x00U
The service request succeeded.
- #define [E_NOT_OK](#) 0x01U
The service request failed.
- #define [CRYPTO_E_BUSY](#) 0x02U
The service request failed because the service is still busy.
- #define [CRYPTO_E_SMALL_BUFFER](#) 0x03U
The service request failed because the provided buffer is too small to store the result.
- #define [CRYPTO_E_ENTROPY_EXHAUSTION](#) 0x04U
The service request failed because the entropy of the random number generator is exhausted.
- #define [CRYPTO_E_QUEUE_FULL](#) 0x05U
The service request failed because the queue is full.
- #define [CRYPTO_E_KEY_READ_FAIL](#) 0x06U
The service request failed because read access failed.
- #define [CRYPTO_E_KEY_WRITE_FAIL](#) 0x07U
The service request failed because write access failed.
- #define [CRYPTO_E_KEY_NOT_AVAILABLE](#) 0x08U
The service request failed because the key is not available.
- #define [CRYPTO_E_KEY_NOT_VALID](#) 0x09U
The service request failed because at least one needed key element is invalid.
- #define [CRYPTO_E_KEY_SIZE_MISMATCH](#) 0x0AU
The service request failed because the key element is not partially accessible and the provided key element length is too short or too long for that key element.

- #define `CRYPTO_E_JOB_CANCELED` 0x0CU
The service request failed because the Job has been canceled.
- #define `CRYPTO_E_KEY_EMPTY` 0x0DU
The service request failed because of uninitialized source key element.
- #define `CRYPTO_KE_MAC_KEY` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_MAC_PROOF` ((ehsm_uint32_t)2U)
- #define `CRYPTO_KE_SIGNATURE_KEY` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_RANDOM_SEED_STATE` ((ehsm_uint32_t)3U)
- #define `CRYPTO_KE_RANDOM_ALGORITHM` ((ehsm_uint32_t)4U)
- #define `CRYPTO_KE_CIPHER_KEY` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_CIPHER_IV` ((ehsm_uint32_t)5U)
- #define `CRYPTO_KE_CIPHER_PROOF` ((ehsm_uint32_t)6U)
- #define `CRYPTO_KE_CIPHER_2NDKEY` ((ehsm_uint32_t)7U)
- #define `CYRPTO_KE_KEYEXCHANGE_SHAREDVALUE` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_KEYEXCHANGE_BASE` ((ehsm_uint32_t)8U)
- #define `CRYPTO_KE_KEYEXCHANGE_PRIVKEY` ((ehsm_uint32_t)9U)
- #define `CRYPTO_KE_KEYEXCHANGE_OWNPUKEY` ((ehsm_uint32_t)10U)
- #define `CRYPTO_KE_KEYEXCHANGE_ALGORITHM` ((ehsm_uint32_t)12U)
- #define `CRYPTO_KE_KEYDERIVATION_PASSWD` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_KEYDERIVATION_SALT` ((ehsm_uint32_t)13U)
- #define `CRYPTO_KE_KEYDERIVATION_ITERATIONS` ((ehsm_uint32_t)14U)
- #define `CRYPTO_KE_KEYDERIVATION_ALGORITHM` ((ehsm_uint32_t)15U)
- #define `CRYPTO_KE_KEYGENERATE_KEY` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_KEYGENERATE_SEED` ((ehsm_uint32_t)16U)
- #define `CRYPTO_KE_KEYGENERATE_ALGORITHM` ((ehsm_uint32_t)17U)
- #define `CRYPTO_KE_CERTIFICATE_DATA` ((ehsm_uint32_t)0U)
- #define `CRYPTO_KE_CERTIFICATE_SUBJECT_PUBLIC_KEY` ((ehsm_uint32_t)1U)
- #define `CRYPTO_KE_CERTIFICATE_PARSING_FORMAT` ((ehsm_uint32_t)18U)
- #define `CRYPTO_KE_CERTIFICATE_CURRENT_TIME` ((ehsm_uint32_t)19U)
- #define `CRYPTO_KE_CERTIFICATE_VERSION` ((ehsm_uint32_t)20U)
- #define `CRYPTO_KE_CERTIFICATE_SERIALNUMBER` ((ehsm_uint32_t)21U)
- #define `CRYPTO_KE_CERTIFICATE_SIGNATURE_ALGORITHM` ((ehsm_uint32_t)22U)
- #define `CRYPTO_KE_CERTIFICATE_ISSUER` ((ehsm_uint32_t)23U)
- #define `CRYPTO_KE_CERTIFICATE_VALIDITY_NOT_BEFORE` ((ehsm_uint32_t)24U)
- #define `CRYPTO_KE_CERTIFICATE_VALIDITY_NOT_AFTER` ((ehsm_uint32_t)25U)
- #define `CRYPTO_KE_CERTIFICATE_SUBJECT` ((ehsm_uint32_t)26U)
- #define `CRYPTO_KE_CERTIFICATE_EXTENSIONS` ((ehsm_uint32_t)27U)
- #define `CRYPTO_KE_CERTIFICATE_SIGNATURE` ((ehsm_uint32_t)28U)

Typedefs

- typedef `ehsm_uint32_t Std_HsmReturnType`
standard returned type.
- typedef void `Crypto_ConfigType`
Configuration data structure of eHSM.

Enumerations

- enum `Crypto_JobStateType` { `CRYPTO_JOBSTATE_IDLE` = 0x00U, `CRYPTO_JOBSTATE_ACTIVE` = 0x01U }
- enum `Crypto_AlgorithmModeType` {
`CRYPTO_ALGOMODE_NOT_SET` = 0x00U, `CRYPTO_ALGOMODE_ECB` = 0x01U, `CRYPTO_ALGOMODE_CBC` = 0x02U, `CRYPTO_ALGOMODE_CFB` = 0x03U,
`CRYPTO_ALGOMODE_OFB` = 0x04U, `CRYPTO_ALGOMODE_CTR` = 0x05U, `CRYPTO_ALGOMODE_GCM` = 0x06U, `CRYPTO_ALGOMODE_XTS` = 0x07U,
`CRYPTO_ALGOMODE_RSAES_OAEP` = 0x08U, `CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5` = 0x09U, `CRYPTO_ALGOMODE_RSASSA_PSS` = 0x0AU, `CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5` = 0x0BU,
`CRYPTO_ALGOMODE_8ROUNDS` = 0x0CU, `CRYPTO_ALGOMODE_12ROUNDS` = 0x0DU, `CRYPTO_ALGOMODE_20ROUNDS` = 0x0EU, `CRYPTO_ALGOMODE_HMAC` = 0x0FU,
`CRYPTO_ALGOMODE_CMAC` = 0x10U, `CRYPTO_ALGOMODE_GMAC` = 0x11U, `CRYPTO_ALGOMODE_CTR_DRBG` = 0x12U, `CRYPTO_ALGOMODE_SIPHASH_2_4` = 0x13U,
`CRYPTO_ALGOMODE_SIPHASH_4_8` = 0x14U, `CRYPTO_ALGOMODE_PXXR1` = 0x15U, `CRYPTO_ALGOMODE_CUSTOM` = 0xFFU, `CRYPTO_ALGOMODE_CCM` = 0x100U }
- enum `Crypto_VerifyResultType` { `CRYPTO_E_VER_OK` = 0x00U, `CRYPTO_E_VER_NOT_OK` = 0x01U }
- enum `Crypto_OperationModeType` {
`CRYPTO_OPERATIONMODE_START` = 0x01U, `CRYPTO_OPERATIONMODE_UPDATE` = 0x02U, `CRYPTO_OPERATIONMODE_STREAMSTART` = 0x03U, `CRYPTO_OPERATIONMODE_FINISH` = 0x04U,
`CRYPTO_OPERATIONMODE_SINGLECALL` = 0x07U }
- enum `Crypto_ServiceInfoType` {
`CRYPTO_HASH` = 0x00U, `CRYPTO_MACGENERATE` = 0x01U, `CRYPTO_MACVERIFY` = 0x02U, `CRYPTO_ENCRYPT` = 0x03U,
`CRYPTO_DECRYPT` = 0x04U, `CRYPTO_AEADENCRYPT` = 0x05U, `CRYPTO_AEADDECRYPT` = 0x06U, `CRYPTO_SIGNATUREGENERATE` = 0x07U,
`CRYPTO_SIGNATUREVERIFY` = 0x08U, `CRYPTO_RANDOMGENERATE` = 0x0BU, `CRYPTO_RANDOMSEED` = 0x0CU, `CRYPTO_KEYGENERATE` = 0x0DU,
`CRYPTO_KEYDERIVE` = 0x0EU, `CRYPTO_KEYEXCHANGEALCPUBVAL` = 0x0FU, `CRYPTO_KEYEXCHANGEALCSECRET` = 0x10U, `CRYPTO_CERTIFICATEPARSE` = 0x11U,
`CRYPTO_CERTIFICATEVERIFY` = 0x12U, `CRYPTO_KEYSETVALID` = 0x13U, `CRYPTO_KEYREMOVE` = 0x80U, `CRYPTO_KEYIMPORT` = 0x81U,
`CRYPTO_KEYEXPORT` = 0x82U }
- enum `Crypto_AlgorithmFamilyType` {
`CRYPTO_ALGOFAM_NOT_SET` = 0x00U, `CRYPTO_ALGOFAM_SHA1` = 0x01U, `CRYPTO_ALGOFAM_SHA2_224` = 0x02U, `CRYPTO_ALGOFAM_SHA2_256` = 0x03U,
`CRYPTO_ALGOFAM_SHA2_384` = 0x04U, `CRYPTO_ALGOFAM_SHA2_512` = 0x05U, `CRYPTO_ALGOFAM_SHA2_512_224` = 0x06U, `CRYPTO_ALGOFAM_SHA2_512_256` = 0x07U,
`CRYPTO_ALGOFAM_SHA3_224` = 0x08U, `CRYPTO_ALGOFAM_SHA3_256` = 0x09U, `CRYPTO_ALGOFAM_SHA3_384` = 0x0AU, `CRYPTO_ALGOFAM_SHA3_512` = 0x0BU,
`CRYPTO_ALGOFAM_SHAKE128` = 0x0CU, `CRYPTO_ALGOFAM_SHAKE256` = 0x0DU, `CRYPTO_ALGOFAM_RIPEMD160` = 0x0EU, `CRYPTO_ALGOFAM_BLAKE_1_256` = 0x0FU,
`CRYPTO_ALGOFAM_BLAKE_1_512` = 0x10U, `CRYPTO_ALGOFAM_BLAKE_2s_256` = 0x11U, `CRYPTO_ALGOFAM_BLAKE_2s_512` = 0x12U, `CRYPTO_ALGOFAM_3DES` = 0x13U,
`CRYPTO_ALGOFAM_AES` = 0x14U, `CRYPTO_ALGOFAM_CHACHA` = 0x15U, `CRYPTO_ALGOFAM_RSA` = 0x16U, `CRYPTO_ALGOFAM_ED25519` = 0x17U,
`CRYPTO_ALGOFAM_BRAINPOOL` = 0x18U, `CRYPTO_ALGOFAM_ECCNIST` = 0x19U, `CRYPTO_ALGOFAM_RNG` = 0x1BU, `CRYPTO_ALGOFAM_SIPHASH` = 0x1CU,
`CRYPTO_ALGOFAM_ECIES` = 0x1DU, `CRYPTO_ALGOFAM_ECCANSI` = 0x1EU, `CRYPTO_ALGOFAM_ECCSECP256K1` = 0x1FU, `CRYPTO_ALGOFAM_DRBG` = 0x20U,
`CRYPTO_ALGOFAM_FIPS186` = 0x21U, `CRYPTO_ALGOFAM_PADDING_PKCS7` = 0x22U, `CRYPTO_ALGOFAM_PADDING_ONETHZEROS` = 0x23U, `CRYPTO_ALGOFAM_PBKDF2` = 0x24U,
`CRYPTO_ALGOFAM_KDFX963` = 0x25U, `CRYPTO_ALGOFAM_DH` = 0x26U, `CRYPTO_ALGOFAM_CUSTOM` = 0xFFU, `CRYPTO_ALGOFAM_SM3` = 0x100U,
`CRYPTO_ALGOFAM_SM4` = 0x101U, `CRYPTO_ALGOFAM_SM2` = 0x102U, `CRYPTO_ALGOFAM_X25519` = 0x103U }
- enum `Crypto_ProcessingType` { `CRYPTO_PROCESSING_ASYNC` = 0x00U, `CRYPTO_PROCESSING_SYNC` = 0x01U }
- enum `Crypto_InputOutputRedirectionConfigType` {
`CRYPTO_REDIRECT_CONFIG_PRIMARY_INPUT` = 0x01, `CRYPTO_REDIRECT_CONFIG_SECONDARY_INPUT` = 0x02 }

- ```

 PUT = 0x02, CRYPTO_REDIRECT_CONFIG_TERTIARY_INPUT = 0x04, CRYPTO_REDIRECT_CONFIG_PRI↵
 MARY_OUTPUT = 0x10,
 CRYPTO_REDIRECT_CONFIG_SECONDARY_OUTPUT = 0x20 }
 • enum Crypto_KeyElementReadAccessType { CRYPTO_RA_DENIED = 0x01U, CRYPTO_RA_INTERNAL_COPY =
 0x02U, CRYPTO_RA_ALLOWED = 0x03U, CRYPTO_RA_ENCRYPTED = 0x04U }
 • enum Crypto_KeyElementWriteAccessType { CRYPTO_WA_DENIED = 0x01U, CRYPTO_WA_INTERNAL_COPY
 = 0x02U, CRYPTO_WA_ALLOWED = 0x03U, CRYPTO_WA_ENCRYPTED = 0x04U }
 • enum CryptoKeyFormatType {
 CRYPTO_KE_FORMAT_BIN_OCTET = 0x01U, CRYPTO_KE_FORMAT_BIN_SHEKEYS = 0x02U, CRYPTO_↵
 KE_FORMAT_BIN_IDENT_PRIVATEKEY_PKCS8 = 0x03U, CRYPTO_KE_FORMAT_BIN_IDENT_PUBLICKEY =
 0x04U,
 CRYPTO_KE_FORMAT_BIN_RSA_PRIVATEKEY = 0x05U, CRYPTO_KE_FORMAT_BIN_RSA_PUBLICKEY =
 0x06U, CRYPTO_KE_FORMAT_BIN_CERT_X509_V3 = 0x07U, CRYPTO_KE_FORMAT_BIN_CERT_CVC =
 0x08U }
 • enum CryptoDriverStateType { CRYPTO_DRIVER_UNINIT = 0x00U, CRYPTO_DRIVER_INITIALIZED = 0x01U }

```

*Enum containing the possible states of the Crypto driver.*

### 4.3.1 Macro Definition Documentation

#### 4.3.1.1 CRYPTO\_E\_BUSY

```
#define CRYPTO_E_BUSY 0x02U
```

The service request failed because the service is still busy.

Definition at line 38 of file Asr\_Standard\_Types.h.

#### 4.3.1.2 CRYPTO\_E\_ENTROPY\_EXHAUSTION

```
#define CRYPTO_E_ENTROPY_EXHAUSTION 0x04U
```

The service request failed because the entropy of the random number generator is exhausted.

#### Note

This is not supported.

Definition at line 50 of file Asr\_Standard\_Types.h.

#### 4.3.1.3 CRYPTO\_E\_JOB\_CANCELED

```
#define CRYPTO_E_JOB_CANCELED 0x0CU
```

The service request failed because the Job has been canceled.

Definition at line 85 of file Asr\_Standard\_Types.h.

#### 4.3.1.4 CRYPTO\_E\_KEY\_EMPTY

```
#define CRYPTO_E_KEY_EMPTY 0x0DU
```

The service request failed because of uninitialized source key element.

Definition at line 90 of file Asr\_Standard\_Types.h.

#### 4.3.1.5 CRYPTO\_E\_KEY\_NOT\_AVAILABLE

```
#define CRYPTO_E_KEY_NOT_AVAILABLE 0x08U
```

The service request failed because the key is not available.

Definition at line 70 of file Asr\_Standard\_Types.h.

#### 4.3.1.6 CRYPTO\_E\_KEY\_NOT\_VALID

```
#define CRYPTO_E_KEY_NOT_VALID 0x09U
```

The service request failed because at least one needed key element is invalid.

Definition at line 75 of file Asr\_Standard\_Types.h.

#### 4.3.1.7 CRYPTO\_E\_KEY\_READ\_FAIL

```
#define CRYPTO_E_KEY_READ_FAIL 0x06U
```

The service request failed because read access failed.

Definition at line 60 of file Asr\_Standard\_Types.h.

#### 4.3.1.8 CRYPTO\_E\_KEY\_SIZE\_MISMATCH

```
#define CRYPTO_E_KEY_SIZE_MISMATCH 0x0AU
```

The service request failed because the key element is not partially accessible and the provided key element length is too short or too long for that key element.

Definition at line 80 of file Asr\_Standard\_Types.h.

#### 4.3.1.9 CRYPTO\_E\_KEY\_WRITE\_FAIL

```
#define CRYPTO_E_KEY_WRITE_FAIL 0x07U
```

The service request failed because write access failed.

Definition at line 65 of file Asr\_Standard\_Types.h.

#### 4.3.1.10 CRYPTO\_E\_QUEUE\_FULL

```
#define CRYPTO_E_QUEUE_FULL 0x05U
```

The service request failed because the queue is full.

Definition at line 55 of file Asr\_Standard\_Types.h.

#### 4.3.1.11 CRYPTO\_E\_SMALL\_BUFFER

```
#define CRYPTO_E_SMALL_BUFFER 0x03U
```

The service request failed because the provided buffer is too small to store the result.

Definition at line 43 of file Asr\_Standard\_Types.h.

#### 4.3.1.12 CRYPTO\_KE\_CERTIFICATE\_CURRENT\_TIME

```
#define CRYPTO_KE_CERTIFICATE_CURRENT_TIME ((ehsm_uint32_t)19U)
```

Definition at line 136 of file Asr\_Standard\_Types.h.

#### 4.3.1.13 CRYPTO\_KE\_CERTIFICATE\_DATA

```
#define CRYPTO_KE_CERTIFICATE_DATA ((ehsm_uint32_t)0U)
```

Definition at line 133 of file Asr\_Standard\_Types.h.

#### 4.3.1.14 CRYPTO\_KE\_CERTIFICATE\_EXTENSIONS

```
#define CRYPTO_KE_CERTIFICATE_EXTENSIONS ((ehsm_uint32_t)27U)
```

Definition at line 144 of file Asr\_Standard\_Types.h.

#### 4.3.1.15 CRYPTO\_KE\_CERTIFICATE\_ISSUER

```
#define CRYPTO_KE_CERTIFICATE_ISSUER ((ehsm_uint32_t)23U)
```

Definition at line 140 of file Asr\_Standard\_Types.h.

#### 4.3.1.16 CRYPTO\_KE\_CERTIFICATE\_PARSING\_FORMAT

```
#define CRYPTO_KE_CERTIFICATE_PARSING_FORMAT ((ehsm_uint32_t)18U)
```

Definition at line 135 of file Asr\_Standard\_Types.h.

#### 4.3.1.17 CRYPTO\_KE\_CERTIFICATE\_SERIALNUMBER

```
#define CRYPTO_KE_CERTIFICATE_SERIALNUMBER ((ehsm_uint32_t)21U)
```

Definition at line 138 of file Asr\_Standard\_Types.h.

#### 4.3.1.18 CRYPTO\_KE\_CERTIFICATE\_SIGNATURE

```
#define CRYPTO_KE_CERTIFICATE_SIGNATURE ((ehsm_uint32_t)28U)
```

Definition at line 145 of file Asr\_Standard\_Types.h.

#### 4.3.1.19 CRYPTO\_KE\_CERTIFICATE\_SIGNATURE\_ALGORITHM

```
#define CRYPTO_KE_CERTIFICATE_SIGNATURE_ALGORITHM ((ehsm_uint32_t)22U)
```

Definition at line 139 of file Asr\_Standard\_Types.h.

#### 4.3.1.20 CRYPTO\_KE\_CERTIFICATE\_SUBJECT

```
#define CRYPTO_KE_CERTIFICATE_SUBJECT ((ehsm_uint32_t)26U)
```

Definition at line 143 of file Asr\_Standard\_Types.h.



#### 4.3.1.21 CRYPTO\_KE\_CERTIFICATE\_SUBJECT\_PUBLIC\_KEY

```
#define CRYPTO_KE_CERTIFICATE_SUBJECT_PUBLIC_KEY ((ehsm_uint32_t)1U)
```

Definition at line 134 of file Asr\_Standard\_Types.h.

#### 4.3.1.22 CRYPTO\_KE\_CERTIFICATE\_VALIDITY\_NOT\_AFTER

```
#define CRYPTO_KE_CERTIFICATE_VALIDITY_NOT_AFTER ((ehsm_uint32_t)25U)
```

Definition at line 142 of file Asr\_Standard\_Types.h.

#### 4.3.1.23 CRYPTO\_KE\_CERTIFICATE\_VALIDITY\_NOT\_BEFORE

```
#define CRYPTO_KE_CERTIFICATE_VALIDITY_NOT_BEFORE ((ehsm_uint32_t)24U)
```

Definition at line 141 of file Asr\_Standard\_Types.h.

#### 4.3.1.24 CRYPTO\_KE\_CERTIFICATE\_VERSION

```
#define CRYPTO_KE_CERTIFICATE_VERSION ((ehsm_uint32_t)20U)
```

Definition at line 137 of file Asr\_Standard\_Types.h.

#### 4.3.1.25 CRYPTO\_KE\_CIPHER\_2NDKEY

```
#define CRYPTO_KE_CIPHER_2NDKEY ((ehsm_uint32_t)7U)
```

Definition at line 108 of file Asr\_Standard\_Types.h.

#### 4.3.1.26 CRYPTO\_KE\_CIPHER\_IV

```
#define CRYPTO_KE_CIPHER_IV ((ehsm_uint32_t)5U)
```

Definition at line 105 of file Asr\_Standard\_Types.h.

#### 4.3.1.27 CRYPTO\_KE\_CIPHER\_KEY

```
#define CRYPTO_KE_CIPHER_KEY ((ehsm_uint32_t)1U)
```

Definition at line 104 of file Asr\_Standard\_Types.h.

#### 4.3.1.28 CRYPTO\_KE\_CIPHER\_PROOF

```
#define CRYPTO_KE_CIPHER_PROOF ((ehsm_uint32_t)6U)
```

Definition at line 106 of file Asr\_Standard\_Types.h.

#### 4.3.1.29 CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM

```
#define CRYPTO_KE_KEYDERIVATION_ALGORITHM ((ehsm_uint32_t)15U)
```

Definition at line 126 of file Asr\_Standard\_Types.h.

#### 4.3.1.30 CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS

```
#define CRYPTO_KE_KEYDERIVATION_ITERATIONS ((ehsm_uint32_t)14U)
```

Definition at line 125 of file Asr\_Standard\_Types.h.

#### 4.3.1.31 CRYPTO\_KE\_KEYDERIVATION\_PASSWD

```
#define CRYPTO_KE_KEYDERIVATION_PASSWD ((ehsm_uint32_t)1U)
```

Definition at line 123 of file Asr\_Standard\_Types.h.

#### 4.3.1.32 CRYPTO\_KE\_KEYDERIVATION\_SALT

```
#define CRYPTO_KE_KEYDERIVATION_SALT ((ehsm_uint32_t)13U)
```

Definition at line 124 of file Asr\_Standard\_Types.h.

#### 4.3.1.33 CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM

```
#define CRYPTO_KE_KEYEXCHANGE_ALGORITHM ((ehsm_uint32_t)12U)
```

Definition at line 119 of file Asr\_Standard\_Types.h.

#### 4.3.1.34 CRYPTO\_KE\_KEYEXCHANGE\_BASE

```
#define CRYPTO_KE_KEYEXCHANGE_BASE ((ehsm_uint32_t)8U)
```

Definition at line 114 of file Asr\_Standard\_Types.h.

#### 4.3.1.35 CRYPTO\_KE\_KEYEXCHANGE\_OWNPUBKEY

```
#define CRYPTO_KE_KEYEXCHANGE_OWNPUBKEY ((ehsm_uint32_t)10U)
```

Definition at line 118 of file Asr\_Standard\_Types.h.

#### 4.3.1.36 CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY

```
#define CRYPTO_KE_KEYEXCHANGE_PRIVKEY ((ehsm_uint32_t)9U)
```

Definition at line 116 of file Asr\_Standard\_Types.h.

#### 4.3.1.37 CRYPTO\_KE\_KEYGENERATE\_ALGORITHM

```
#define CRYPTO_KE_KEYGENERATE_ALGORITHM ((ehsm_uint32_t)17U)
```

Definition at line 131 of file Asr\_Standard\_Types.h.

#### 4.3.1.38 CRYPTO\_KE\_KEYGENERATE\_KEY

```
#define CRYPTO_KE_KEYGENERATE_KEY ((ehsm_uint32_t)1U)
```

Definition at line 129 of file Asr\_Standard\_Types.h.

#### 4.3.1.39 CRYPTO\_KE\_KEYGENERATE\_SEED

```
#define CRYPTO_KE_KEYGENERATE_SEED ((ehsm_uint32_t)16U)
```

Definition at line 130 of file Asr\_Standard\_Types.h.

#### 4.3.1.40 CRYPTO\_KE\_MAC\_KEY

```
#define CRYPTO_KE_MAC_KEY ((ehsm_uint32_t)1U)
```

Definition at line 94 of file Asr\_Standard\_Types.h.

#### 4.3.1.41 CRYPTO\_KE\_MAC\_PROOF

```
#define CRYPTO_KE_MAC_PROOF ((ehsm_uint32_t)2U)
```

Definition at line 95 of file Asr\_Standard\_Types.h.

#### 4.3.1.42 CRYPTO\_KE\_RANDOM\_ALGORITHM

```
#define CRYPTO_KE_RANDOM_ALGORITHM ((ehsm_uint32_t)4U)
```

Definition at line 101 of file Asr\_Standard\_Types.h.

#### 4.3.1.43 CRYPTO\_KE\_RANDOM\_SEED\_STATE

```
#define CRYPTO_KE_RANDOM_SEED_STATE ((ehsm_uint32_t)3U)
```

Definition at line 100 of file Asr\_Standard\_Types.h.

#### 4.3.1.44 CRYPTO\_KE\_SIGNATURE\_KEY

```
#define CRYPTO_KE_SIGNATURE_KEY ((ehsm_uint32_t)1U)
```

Definition at line 97 of file Asr\_Standard\_Types.h.

#### 4.3.1.45 CYRPTO\_KEY\_KEYEXCHANGE\_SHAREDVALUE

```
#define CYRPTO_KEY_KEYEXCHANGE_SHAREDVALUE ((ehsm_uint32_t)1U)
```

Definition at line 112 of file Asr\_Standard\_Types.h.

#### 4.3.1.46 E\_NOT\_OK

```
#define E_NOT_OK 0x01U
```

The service request failed.

Definition at line 33 of file Asr\_Standard\_Types.h.

#### 4.3.1.47 E\_OK

```
#define E_OK 0x00U
```

The service request succeeded.

Definition at line 28 of file Asr\_Standard\_Types.h.

### 4.3.2 Typedef Documentation

#### 4.3.2.1 Crypto\_ConfigType

```
typedef void Crypto_ConfigType
```

Configuration data structure of eHSM.

Definition at line 154 of file Asr\_Standard\_Types.h.

#### 4.3.2.2 Std\_HsmReturnType

```
typedef ehsm_uint32_t Std_HsmReturnType
```

standard returned type.

Definition at line 23 of file Asr\_Standard\_Types.h.

### 4.3.3 Enumeration Type Documentation

#### 4.3.3.1 Crypto\_AlgorithmFamilyType

```
enum Crypto_AlgorithmFamilyType
```

## Enumerator

|                                     |  |
|-------------------------------------|--|
| CRYPTO_ALGOFAM_NOT_SET              |  |
| CRYPTO_ALGOFAM_SHA1                 |  |
| CRYPTO_ALGOFAM_SHA2_224             |  |
| CRYPTO_ALGOFAM_SHA2_256             |  |
| CRYPTO_ALGOFAM_SHA2_384             |  |
| CRYPTO_ALGOFAM_SHA2_512             |  |
| CRYPTO_ALGOFAM_SHA2_512_224         |  |
| CRYPTO_ALGOFAM_SHA2_512_256         |  |
| CRYPTO_ALGOFAM_SHA3_224             |  |
| CRYPTO_ALGOFAM_SHA3_256             |  |
| CRYPTO_ALGOFAM_SHA3_384             |  |
| CRYPTO_ALGOFAM_SHA3_512             |  |
| CRYPTO_ALGOFAM_SHAKE128             |  |
| CRYPTO_ALGOFAM_SHAKE256             |  |
| CRYPTO_ALGOFAM_RIPEMD160            |  |
| CRYPTO_ALGOFAM_BLAKE_1_256          |  |
| CRYPTO_ALGOFAM_BLAKE_1_512          |  |
| CRYPTO_ALGOFAM_BLAKE_2s_256         |  |
| CRYPTO_ALGOFAM_BLAKE_2s_512         |  |
| CRYPTO_ALGOFAM_3DES                 |  |
| CRYPTO_ALGOFAM_AES                  |  |
| CRYPTO_ALGOFAM_CHACHA               |  |
| CRYPTO_ALGOFAM_RSA                  |  |
| CRYPTO_ALGOFAM_ED25519              |  |
| CRYPTO_ALGOFAM_BRAINPOOL            |  |
| CRYPTO_ALGOFAM_ECCNIST              |  |
| CRYPTO_ALGOFAM_RNG                  |  |
| CRYPTO_ALGOFAM_SIPHASH              |  |
| CRYPTO_ALGOFAM_ECIES                |  |
| CRYPTO_ALGOFAM_ECCANSI              |  |
| CRYPTO_ALGOFAM_ECCSEC               |  |
| CRYPTO_ALGOFAM_DRBG                 |  |
| CRYPTO_ALGOFAM_FIPS186              |  |
| CRYPTO_ALGOFAM_PADDING_PKCS7        |  |
| CRYPTO_ALGOFAM_PADDING_ONEWITHZEROS |  |
| CRYPTO_ALGOFAM_PBKDF2               |  |
| CRYPTO_ALGOFAM_KDFX963              |  |
| CRYPTO_ALGOFAM_DH                   |  |
| CRYPTO_ALGOFAM_CUSTOM               |  |
| CRYPTO_ALGOFAM_SM3                  |  |
| CRYPTO_ALGOFAM_SM4                  |  |
| CRYPTO_ALGOFAM_SM2                  |  |
| CRYPTO_ALGOFAM_X25519               |  |

Definition at line 246 of file Asr\_Standard\_Types.h.

## 4.3.3.2 Crypto\_AlgorithmModeType

```
enum Crypto_AlgorithmModeType
```

## Enumerator

|                                   |  |
|-----------------------------------|--|
| CRYPTO_ALGOMODE_NOT_SET           |  |
| CRYPTO_ALGOMODE_ECB               |  |
| CRYPTO_ALGOMODE_CBC               |  |
| CRYPTO_ALGOMODE_CFB               |  |
| CRYPTO_ALGOMODE_OFB               |  |
| CRYPTO_ALGOMODE_CTR               |  |
| CRYPTO_ALGOMODE_GCM               |  |
| CRYPTO_ALGOMODE_XTS               |  |
| CRYPTO_ALGOMODE_RSAES_OAEP        |  |
| CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5  |  |
| CRYPTO_ALGOMODE_RSASSA_PSS        |  |
| CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5 |  |
| CRYPTO_ALGOMODE_8ROUNDS           |  |
| CRYPTO_ALGOMODE_12ROUNDS          |  |
| CRYPTO_ALGOMODE_20ROUNDS          |  |
| CRYPTO_ALGOMODE_HMAC              |  |
| CRYPTO_ALGOMODE_CMAC              |  |
| CRYPTO_ALGOMODE_GMAC              |  |
| CRYPTO_ALGOMODE_CTRDRBG           |  |
| CRYPTO_ALGOMODE_SIPHASH_2_4       |  |
| CRYPTO_ALGOMODE_SIPHASH_4_8       |  |
| CRYPTO_ALGOMODE_PXXR1             |  |
| CRYPTO_ALGOMODE_CUSTOM            |  |
| CRYPTO_ALGOMODE_CCM               |  |

Definition at line 171 of file Asr\_Standard\_Types.h.

## 4.3.3.3 Crypto\_InputOutputRedirectionConfigType

```
enum Crypto_InputOutputRedirectionConfigType
```

## Enumerator

|                                         |  |
|-----------------------------------------|--|
| CRYPTO_REDIRECT_CONFIG_PRIMARY_INPUT    |  |
| CRYPTO_REDIRECT_CONFIG_SECONDARY_INPUT  |  |
| CRYPTO_REDIRECT_CONFIG_TERTIARY_INPUT   |  |
| CRYPTO_REDIRECT_CONFIG_PRIMARY_OUTPUT   |  |
| CRYPTO_REDIRECT_CONFIG_SECONDARY_OUTPUT |  |

Definition at line 359 of file Asr\_Standard\_Types.h.

## 4.3.3.4 Crypto\_JobStateType

```
enum Crypto_JobStateType
```

**Enumerator**

|                        |  |
|------------------------|--|
| CRYPTO_JOBSTATE_IDLE   |  |
| CRYPTO_JOBSTATE_ACTIVE |  |

Definition at line 164 of file Asr\_Standard\_Types.h.

**4.3.3.5 Crypto\_KeyElementReadAccessType**

enum [Crypto\\_KeyElementReadAccessType](#)

**Enumerator**

|                         |  |
|-------------------------|--|
| CRYPTO_RA_DENIED        |  |
| CRYPTO_RA_INTERNAL_COPY |  |
| CRYPTO_RA_ALLOWED       |  |
| CRYPTO_RA_ENCRYPTED     |  |

Definition at line 413 of file Asr\_Standard\_Types.h.

**4.3.3.6 Crypto\_KeyElementWriteAccessType**

enum [Crypto\\_KeyElementWriteAccessType](#)

**Enumerator**

|                         |  |
|-------------------------|--|
| CRYPTO_WA_DENIED        |  |
| CRYPTO_WA_INTERNAL_COPY |  |
| CRYPTO_WA_ALLOWED       |  |
| CRYPTO_WA_ENCRYPTED     |  |

Definition at line 421 of file Asr\_Standard\_Types.h.

**4.3.3.7 Crypto\_OperationModeType**

enum [Crypto\\_OperationModeType](#)

**Enumerator**

|                                  |  |
|----------------------------------|--|
| CRYPTO_OPERATIONMODE_START       |  |
| CRYPTO_OPERATIONMODE_UPDATE      |  |
| CRYPTO_OPERATIONMODE_STREAMSTART |  |
| CRYPTO_OPERATIONMODE_FINISH      |  |
| CRYPTO_OPERATIONMODE_SINGLECALL  |  |



Definition at line 208 of file Asr\_Standard\_Types.h.

#### 4.3.3.8 Crypto\_ProcessingType

enum [Crypto\\_ProcessingType](#)

##### Enumerator

|                         |  |
|-------------------------|--|
| CRYPTO_PROCESSING_ASYNC |  |
| CRYPTO_PROCESSING_SYNC  |  |

Definition at line 338 of file Asr\_Standard\_Types.h.

#### 4.3.3.9 Crypto\_ServiceInfoType

enum [Crypto\\_ServiceInfoType](#)

##### Enumerator

|                             |  |
|-----------------------------|--|
| CRYPTO_HASH                 |  |
| CRYPTO_MACGENERATE          |  |
| CRYPTO_MACVERIFY            |  |
| CRYPTO_ENCRYPT              |  |
| CRYPTO_DECRYPT              |  |
| CRYPTO_AEADENCRYPT          |  |
| CRYPTO_AEADDECRYPT          |  |
| CRYPTO_SIGNATUREGENERATE    |  |
| CRYPTO_SIGNATUREVERIFY      |  |
| CRYPTO_RANDOMGENERATE       |  |
| CRYPTO_RANDOMSEED           |  |
| CRYPTO_KEYGENERATE          |  |
| CRYPTO_KEYDERIVE            |  |
| CRYPTO_KEYEXCHANGEALCPUBVAL |  |
| CRYPTO_KEYEXCHANGEALCSECRET |  |
| CRYPTO_CERTIFICATEPARSE     |  |
| CRYPTO_CERTIFICATEVERIFY    |  |
| CRYPTO_KEYSETVALID          |  |
| CRYPTO_KEYREMOVE            |  |
| CRYPTO_KEYIMPORT            |  |
| CRYPTO_KEYEXPORT            |  |

Definition at line 218 of file Asr\_Standard\_Types.h.

#### 4.3.3.10 Crypto\_VerifyResultType

enum `Crypto_VerifyResultType`

## Enumerator

|                     |  |
|---------------------|--|
| CRYPTO_E_VER_OK     |  |
| CRYPTO_E_VER_NOT_OK |  |

Definition at line 200 of file Asr\_Standard\_Types.h.

## 4.3.3.11 CryptoDriverStateType

enum [CryptoDriverStateType](#)

Enum containing the possible states of the Crypto driver.

## Enumerator

|                           |  |
|---------------------------|--|
| CRYPTO_DRIVER_UNINIT      |  |
| CRYPTO_DRIVER_INITIALIZED |  |

Definition at line 444 of file Asr\_Standard\_Types.h.

## 4.3.3.12 CryptoKeyFormatType

enum [CryptoKeyFormatType](#)

## Enumerator

|                                             |  |
|---------------------------------------------|--|
| CRYPTO_KE_FORMAT_BIN_OCTET                  |  |
| CRYPTO_KE_FORMAT_BIN_SHEKEYS                |  |
| CRYPTO_KE_FORMAT_BIN_IDENT_PRIVATEKEY_PKCS8 |  |
| CRYPTO_KE_FORMAT_BIN_IDENT_PUBLICKEY        |  |
| CRYPTO_KE_FORMAT_BIN_RSA_PRIVATEKEY         |  |
| CRYPTO_KE_FORMAT_BIN_RSA_PUBLICKEY          |  |
| CRYPTO_KE_FORMAT_BIN_CERT_X509_V3           |  |
| CRYPTO_KE_FORMAT_BIN_CERT_CVC               |  |

Definition at line 429 of file Asr\_Standard\_Types.h.

## 4.4 eHSM\_Com\_Struct\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
```

## Classes

- struct [ehsm\\_key\\_flags\\_element\\_st](#)
- struct [ehsm\\_key\\_usages\\_st](#)
- struct [ehsm\\_create\\_random\\_key\\_param](#)
- struct [ehsm\\_key\\_derived\\_param](#)
- struct [ehsm\\_evita\\_key\\_import\\_st](#)
- struct [ehsm\\_evita\\_key\\_export](#)
- struct [sm2\\_ext\\_param](#)
- struct [ehsm\\_create\\_dh\\_key\\_param](#)
- struct [ehsm\\_get\\_pub\\_from\\_priv\\_param](#)
- struct [ehsm\\_key\\_remove\\_param](#)
- struct [ehsm\\_key\\_status\\_param](#)
- struct [ehsm\\_key\\_copy\\_param](#)
- struct [ehsm\\_she\\_key\\_param](#)
- struct [ehsm\\_she\\_key\\_host\\_param](#)
- struct [ehsm\\_she\\_plain\\_key\\_param](#)
- struct [ehsm\\_she\\_plain\\_key\\_host\\_param](#)
- struct [ehsm\\_keyexchange\\_key\\_info](#)
- struct [ehsm\\_crypto\\_randomgenerate\\_param](#)
- struct [ehsm\\_get\\_challenge\\_st](#)
- struct [ehsm\\_debug\\_auth\\_st](#)
- struct [ehsm\\_fw\\_random\\_key](#)
- struct [ehsm\\_fw\\_encrypt\\_key](#)
- struct [ehsm\\_image](#)
- struct [ehsm\\_image\\_verify\\_st](#)
- struct [soc\\_image\\_upgrade\\_input](#)
- struct [soc\\_image\\_upgrade\\_info](#)
- struct [ehsm\\_soc\\_image\\_verify\\_st](#)
- struct [soc\\_image\\_verify\\_input](#)
- struct [soc\\_image\\_verify\\_info](#)
- struct [ehsm\\_gen\\_dh\\_key\\_param\\_st](#)
- struct [ehsm\\_gen\\_key\\_param\\_st](#)
- struct [ehsm\\_create\\_evita\\_key\\_param](#)
- struct [ehsm\\_gen\\_sm9\\_master\\_key\\_param](#)
- struct [ehsm\\_gen\\_sm9\\_userpriv\\_key\\_param](#)
- struct [ehsm\\_gen\\_sm9\\_key\\_param](#)
- struct [ehsm\\_exchange\\_sm9\\_key\\_param](#)
- struct [ehsm\\_sm9\\_wrap\\_key\\_param](#)
- struct [ehsm\\_sm9\\_unwrap\\_key\\_param](#)
- struct [ehsm\\_sm9\\_exckey\\_gen\\_tmpkey\\_param](#)
- struct [ehsm\\_sm9\\_inexport\\_key\\_param](#)
- struct [ehsm\\_sm9\\_gen\\_mast\\_pubkey](#)
- struct [ehsm\\_sm9\\_gen\\_tmp\\_pubkey\\_param](#)
- struct [ehsm\\_otp\\_read\\_param\\_st](#)
- struct [ehsm\\_otp\\_write\\_param\\_st](#)
- struct [ehsm\\_storage\\_area\\_param\\_st](#)
- struct [ehsm\\_change\\_control\\_field\\_st](#)
- struct [ehsm\\_she\\_get\\_id\\_param\\_st](#)
- struct [ehsm\\_get\\_emu\\_status\\_param\\_st](#)
- struct [ehsm\\_emu\\_status\\_st](#)
- struct [ehsm\\_evita\\_memory\\_info\\_st](#)
- struct [ehsm\\_module\\_status\\_st](#)

*The structure to store the information of ehsm status. The parameter alg, key\_handle, key\_auth\_size, key\_auth\_value, sign←\_size and sign are only valid when type is not EHSM\_GET\_STATUS\_SHE.*

- struct [ehsm\\_certificate\\_verify\\_st](#)
- struct [ehsm\\_sensor\\_init\\_param\\_st](#)

## Macros

- `#define OTP_CONTROL_FILED_BYTE_SIZE` (8)
- `#define EHSM_SHE_M1_MAX_SIZE` (32)
- `#define EHSM_SHE_M2_MAX_SIZE` (32)
- `#define EHSM_SHE_M3_MAX_SIZE` (16)
- `#define EHSM_SHE_M4_MAX_SIZE` (48)
- `#define EHSM_SHE_M5_MAX_SIZE` (16)
- `#define EHSM_EVITA_AUTH_VALUE_MAX_SIZE` (32)
- `#define DEFAULT_RSAKEY_E_SIZE` 17
- `#define EHSM_GET_STATUS_SHE` 0x0
- `#define EHSM_GET_STATUS_SBB` 0x1
- `#define EHSM_GET_STATUS_MEM` 0x2
- `#define EHSM_GET_STATUS_ERRORS` 0x4
- `#define EHSM_SELF_TEST_SKE_DES` (0x1U << 0)
- `#define EHSM_SELF_TEST_SKE_TDES` (0x1U << 1)
- `#define EHSM_SELF_TEST_SKE_AES` (0x1U << 2)
- `#define EHSM_SELF_TEST_SKE_SM4` (0x1U << 3)
- `#define EHSM_SELF_TEST_PKE_RSA` (0x1U << 8)
- `#define EHSM_SELF_TEST_PKE_ECC` (0x1U << 9)
- `#define EHSM_SELF_TEST_PKE_SM2` (0x1U << 10)
- `#define EHSM_SELF_TEST_PKE_SM9` (0x1U << 11)
- `#define EHSM_SELF_TEST_HASH_MD5` (0x1U << 16)
- `#define EHSM_SELF_TEST_HASH_SHA1` (0x1U << 17)
- `#define EHSM_SELF_TEST_HASH_SHA2` (0x1U << 18)
- `#define EHSM_SELF_TEST_HASH_SHA3` (0x1U << 19)
- `#define EHSM_SELF_TEST_HASH_SM3` (0x1U << 20)
- `#define EHSM_SELF_TEST_HASH_SHA256` (0x1U << 21)
- `#define EHSM_SELF_TEST_TRNG` (0x1U << 24)
- `#define EHSM_SELF_TEST_SKE`
- `#define EHSM_SELF_TEST_PKE`
- `#define EHSM_SELF_TEST_HASH`
- `#define EHSM_SELF_TEST_ALL`
- `#define EHSM_FAST_CMAC_AES128` 1
- `#define EHSM_FAST_CMAC_SM4` 2
- `#define EHSM_FAST_CMAC_GEN` 0
- `#define EHSM_FAST_CMAC_VERIFY` 1
- `#define EHSM_FAST_CMAC_SHE_KEY` EHSM\_CMD\_CIPHER\_KEY\_TYPE\_SHE
- `#define EHSM_FAST_CMAC_EVITA_KEY` EHSM\_CMD\_CIPHER\_KEY\_TYPE\_EVITA
- `#define SECURE_BOOT_TYPE_IMAGE_VERIFY` (0x01U)
- `#define SECURE_BOOT_TYPE_SECURE_BOOT` (0x02U)
- `#define IMAGE_SIGNATURE_MAX_LENGTH` (256U)
- `#define IMAGE_PUBLIC_KEY_MAX_LENGTH` (64U+256U)
- `#define EHSM_CODE_VERIFY_FALG` (0x01)
- `#define SOC_CODE_VERIFY_FALG` (0x02)
- `#define CODE_VALID_FLAG` (0x8E97645DUL)
- `#define UPGRADE_VALID_FLAG` (0x71689BA2UL)
- `#define IMAGE_DECRYPT_CODE` (0x5A)
- `#define IMAGE_ENCRYPT_CODE` (0x5C)
- `#define IMAGE_ANALYSIS_CODE` (0x55)
- `#define SOC_BOOT_TYPE_SEQUENTIAL` 0x0U
- `#define SOC_BOOT_TYPE_PARALLEL` 0x1U

## Typedefs

- typedef struct ehsm\_create\_random\_key\_param ehsm\_create\_random\_key\_param\_st
- typedef struct ehsm\_key\_derived\_param ehsm\_key\_derived\_param\_st
- typedef struct ehsm\_evita\_key\_export ehsm\_evita\_key\_export\_st
- typedef struct sm2\_ext\_param sm2\_ext\_param\_st
- typedef struct ehsm\_create\_dh\_key\_param ehsm\_create\_dh\_key\_param\_st
- typedef struct ehsm\_get\_pub\_from\_priv\_param ehsm\_get\_pub\_from\_priv\_param\_st
- typedef struct ehsm\_key\_remove\_param ehsm\_key\_remove\_param\_st
- typedef struct ehsm\_key\_status\_param ehsm\_key\_status\_param\_st
- typedef struct ehsm\_key\_copy\_param ehsm\_key\_copy\_param\_st
- typedef struct ehsm\_she\_key\_param ehsm\_she\_key\_param\_st
- typedef struct ehsm\_she\_key\_host\_param ehsm\_she\_key\_host\_param\_st
- typedef struct ehsm\_she\_plain\_key\_param ehsm\_she\_plain\_key\_param\_st
- typedef struct ehsm\_she\_plain\_key\_host\_param ehsm\_she\_plain\_key\_host\_param\_st
- typedef struct ehsm\_keyexchange\_key\_info ehsm\_keyexchange\_key\_info\_st
- typedef struct ehsm\_crypto\_randomgenerate\_param ehsm\_crypto\_randomgenerate\_param\_st
- typedef struct ehsm\_fw\_random\_key ehsm\_fw\_random\_key\_st
- typedef struct ehsm\_fw\_encrypt\_key ehsm\_fw\_encrypt\_key\_st
- typedef struct ehsm\_image ehsm\_image\_upgrade\_st
- typedef struct soc\_image\_upgrade\_input ehsm\_soc\_image\_upgrade\_input\_st
- typedef struct soc\_image\_upgrade\_info ehsm\_soc\_image\_upgrade\_info\_st
- typedef struct soc\_image\_verify\_input ehsm\_soc\_image\_verify\_input\_st
- typedef struct soc\_image\_verify\_info ehsm\_soc\_image\_verify\_info\_st
- typedef struct ehsm\_create\_evita\_key\_param ehsm\_create\_evita\_key\_param\_st
- typedef struct ehsm\_gen\_sm9\_master\_key\_param ehsm\_gen\_sm9\_master\_key\_param\_st
- typedef struct ehsm\_gen\_sm9\_userpriv\_key\_param ehsm\_gen\_sm9\_userpriv\_key\_param\_st
- typedef struct ehsm\_gen\_sm9\_key\_param ehsm\_gen\_sm9\_key\_param\_st
- typedef struct ehsm\_exchange\_sm9\_key\_param ehsm\_exchange\_sm9\_key\_param\_st
- typedef struct ehsm\_sm9\_wrap\_key\_param ehsm\_sm9\_wrap\_key\_param\_st
- typedef struct ehsm\_sm9\_unwrap\_key\_param ehsm\_sm9\_unwrap\_key\_param\_st
- typedef struct ehsm\_sm9\_exckey\_gen\_tmpkey\_param ehsm\_sm9\_exckey\_gen\_tmpkey\_st
- typedef struct ehsm\_sm9\_inexport\_key\_param ehsm\_sm9\_inexport\_key\_param\_st
- typedef struct ehsm\_sm9\_gen\_mast\_pubkey ehsm\_sm9\_gen\_mast\_pubkey\_st
- typedef struct ehsm\_sm9\_gen\_tmp\_pubkey\_param ehsm\_sm9\_gen\_tmp\_pubkey\_st
- typedef struct ehsm\_emu\_status\_st ehsm\_emu\_status\_st
- typedef struct ehsm\_evita\_memory\_info\_st ehsm\_evita\_memory\_info\_st

## Enumerations

- enum ehsm\_key\_mem\_type\_e { EHSM\_EVITA\_KEY\_TYPE\_NVM = 0U, EHSM\_EVITA\_KEY\_TYPE\_RAM = 1U }
- enum ehsm\_dh\_mode\_e { EHSM\_DH\_MODE\_KEY\_HANDLE, EHSM\_DH\_MODE\_KEY\_PUB\_KEY, EHSM\_DH\_MODE\_RAW\_PUB\_KEY }
- enum ehsm\_rsa\_key\_type\_e { EHSM\_RSA\_KEY\_TYPE\_COMMON, EHSM\_RSA\_KEY\_TYPE\_CRT }
- enum sm2\_key\_exchange\_role\_e { SM2\_KEY\_EXCHANGE\_ROLE\_SPONSOR = 0, SM2\_KEY\_EXCHANGE\_ROLE\_RESPONSOR }
- enum ehsm\_uart\_baudrate\_e { EHSM\_UART\_BAUDRATE\_9600 = 1, EHSM\_UART\_BAUDRATE\_19200, EHSM\_UART\_BAUDRATE\_38400, EHSM\_UART\_BAUDRATE\_57600, EHSM\_UART\_BAUDRATE\_115200, EHSM\_UART\_BAUDRATE\_INVALID }
- enum crypto\_key\_derive\_type\_e { CRYPTO\_KEY\_DERIVE\_USER\_PASSWD = 1, CRYPTO\_KEY\_DERIVE\_USER\_KEYHANDLE }
- enum ehsm\_challenge\_type\_e { EHSM\_CHALLENGE\_TYPE\_INVALID = 0, EHSM\_CHALLENGE\_TYPE\_TIME\_SYNC = 1, EHSM\_CHALLENGE\_TYPE\_EHSM\_DEBUG = 2, EHSM\_CHALLENGE\_TYPE\_SHE\_DEBUG = 3, EHSM\_CHALLENGE\_TYPE\_SOC\_DEBUG = 4, EHSM\_CHALLENGE\_TYPE\_USER\_AUTH = 5, EHSM\_CHALLENGE\_TYPE\_MAX }

- enum ehsm\_debug\_auth\_alg\_e { EHSM\_DEBUG\_AUTH\_ALG\_SM2\_WITH\_SM3 = 1, EHSM\_DEBUG\_AUTH\_ALG\_ECCSECP256R1\_WITH\_SHA256 = 2, EHSM\_DEBUG\_AUTH\_ALG\_SM4\_CMAC = 3, EHSM\_DEBUG\_AUTH\_ALG\_AES128\_CMAC = 4 }
- enum ehsm\_fw\_random\_key\_type\_e { EHSM\_FW\_RANDOM\_KEY\_TYPE\_SYMMETRIC\_KEY = 0x01, EHSM\_FW\_RANDOM\_KEY\_TYPE\_SM2\_PRIVATE\_KEY = 0x02, EHSM\_FW\_RANDOM\_KEY\_TYPE\_SECP256R1\_PRIVATE\_KEY = 0x04 }
- enum ehsm\_fw\_random\_key\_slot\_e { EHSM\_FW\_RANDOM\_KEY\_SLOT\_DEVICE\_ROOT\_KEY = 0x01, EHSM\_FW\_RANDOM\_KEY\_SLOT\_SOC\_FW\_VERIFY\_KEY = 0x02, EHSM\_FW\_RANDOM\_KEY\_SLOT\_SOC\_ENC\_KEY = 0x04, EHSM\_FW\_RANDOM\_KEY\_SLOT\_SOC\_PRIVATE\_KEY = 0x08, EHSM\_FW\_RANDOM\_KEY\_SLOT\_USER\_ROOT\_KEY = 0x10 }
- enum ehsm\_fw\_encrypt\_key\_type\_e { EHSM\_FW\_ENCRYPT\_KEY\_TYPE\_SYMMETRIC\_KEY = 0x01, EHSM\_FW\_ENCRYPT\_KEY\_TYPE\_PUBLIC\_KEY\_HASH = 0x02 }
- enum ehsm\_fw\_encrypt\_key\_slot\_e { EHSM\_FW\_ENCRYPT\_KEY\_SLOT\_SOC\_DEBUG\_KEY = 0x01, EHSM\_FW\_ENCRYPT\_KEY\_SLOT\_SOC\_FW\_VERIFY\_KEY = 0x02, EHSM\_FW\_ENCRYPT\_KEY\_SLOT\_SOC\_UPGRADE\_ENC\_KEY = 0x04, EHSM\_FW\_ENCRYPT\_KEY\_SLOT\_SOC\_UPGRADE\_VERIFY\_KEY = 0x08, EHSM\_FW\_ENCRYPT\_KEY\_SLOT\_USER\_DEBUG\_KEY = 0x10 }
- enum ehsm\_image\_process\_mode\_e { EHSM\_IMAGE\_PROCESS\_MODE\_INIT = 0x01, EHSM\_IMAGE\_PROCESS\_MODE\_UPDATE = 0x02, EHSM\_IMAGE\_PROCESS\_MODE\_FINISH = 0x04, EHSM\_IMAGE\_PROCESS\_MODE\_ONEPASS = 0x08 }
- enum ehsm\_code\_verify\_alg\_e { CODE\_VERIFY\_ALG\_RSA = 0U, CODE\_VERIFY\_ALG\_SM2, CODE\_VERIFY\_ALG\_AES128\_CMAC, CODE\_VERIFY\_ALG\_SM4\_CMAC, CODE\_VERIFY\_ALG\_INVALID = 0xFFU }
- enum ehsm\_code\_upgrade\_alg\_e { CODE\_UPGRADE\_ALG\_RSA = 0U, CODE\_UPGRADE\_ALG\_SM2 = 1U, CODE\_UPGRADE\_ALG\_AES128\_GCM = 2U, CODE\_UPGRADE\_ALG\_SM4\_GCM = 3U, CODE\_UPGRADE\_ALG\_AES128\_CMAC = 4U, CODE\_UPGRADE\_ALG\_SM4\_CMAC = 5U, CODE\_UPGRADE\_ALG\_INVALID = 0xFFU }
- enum ehsm\_sm9\_exchg\_key\_role\_e { EHSM\_SM9\_EXCHG\_KEY\_ROLE\_SELF = 1u, EHSM\_SM9\_EXCHG\_KEY\_ROLE\_PEER }
- enum ehsm\_sm9\_master\_key\_type\_e { EHSM\_GEN\_SM9\_SIGN\_MASTER\_KEY = 0U, EHSM\_GEN\_SM9\_ENC\_MASTER\_KEY = 1U, EHSM\_GEN\_SM9\_EXCHG\_MASTER\_KEY = 2U, EHSM\_GEN\_SM9\_INVALID\_MASTER\_KEY }
- enum ehsm\_sm9\_user\_privkey\_type\_e { EHSM\_GEN\_SM9\_SIGN\_USERPRIV\_KEY = 0U, EHSM\_GEN\_SM9\_ENC\_USERPRIV\_KEY = 1U, EHSM\_GEN\_SM9\_EXCHG\_USERPRIV\_KEY = 2U, EHSM\_GEN\_SM9\_EXCHG\_USERTMP\_KEY = 3U, EHSM\_GEN\_SM9\_INVALID\_USERPRIV\_KEY }
- enum ehsm\_gen\_sm9\_key\_type\_e { EHSM\_GEN\_SM9\_MASTER\_KEY = 0U, EHSM\_GEN\_SM9\_PRIV\_KEY = 1U, EHSM\_GEN\_SM9\_INVALID\_KEY }
- enum ehsm\_lifecycle\_e { EHSM\_LIFE\_CYCLE\_UNNORMAL\_MODE = 0x00, EHSM\_LIFE\_CYCLE\_TEST\_MODE = 0x01, EHSM\_LIFE\_CYCLE\_DEV\_MODE = 0x02, EHSM\_LIFE\_CYCLE\_MANU\_MODE = 0x03, EHSM\_LIFE\_CYCLE\_USER\_MODE = 0x04, EHSM\_LIFE\_CYCLE\_DEBUG\_MODE = 0x05, EHSM\_LIFE\_CYCLE\_DESTROY\_MODE = 0x06 }
- enum ehsm\_control\_field\_type\_e { EHSM\_CONTROL\_FIELD\_TYPE\_HW, EHSM\_CONTROL\_FIELD\_TYPE\_EHSM, EHSM\_CONTROL\_FIELD\_TYPE\_SOC }
- enum ehsm\_api\_type\_e { EHSM\_API\_TYPE\_SHE = 1, EHSM\_API\_TYPE\_EVITA, EHSM\_API\_TYPE\_AUTOSAR, EHSM\_API\_TYPE\_EXT, EHSM\_API\_TYPE\_INVALID }

#### 4.4.1 Macro Definition Documentation

#### 4.4.1.1 CODE\_VALID\_FLAG

```
#define CODE_VALID_FLAG (0x8E97645DUL)
```

Definition at line 107 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.2 DEFAULT\_RSAKEY\_E\_SIZE

```
#define DEFAULT_RSAKEY_E_SIZE 17
```

Definition at line 27 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.3 EHSM\_CODE\_VERIFY\_FALG

```
#define EHSM_CODE_VERIFY_FALG (0x01)
```

Definition at line 104 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.4 EHSM\_EVITA\_AUTH\_VALUE\_MAX\_SIZE

```
#define EHSM_EVITA_AUTH_VALUE_MAX_SIZE (32)
```

Definition at line 26 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.5 EHSM\_FAST\_CMAC\_AES128

```
#define EHSM_FAST_CMAC_AES128 1
```

Definition at line 87 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.6 EHSM\_FAST\_CMAC\_EVITA\_KEY

```
#define EHSM_FAST_CMAC_EVITA_KEY EHSM_CMD_CIPHER_KEY_TYPE_EVITA
```

Definition at line 96 of file eHSM\_Com\_Struct\_lp.h.



#### 4.4.1.7 EHSM\_FAST\_CMACE\_GEN

```
#define EHSM_FAST_CMACE_GEN 0
```

Definition at line 91 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.8 EHSM\_FAST\_CMACE\_SHE\_KEY

```
#define EHSM_FAST_CMACE_SHE_KEY EHSM_CMD_CIPHER_KEY_TYPE_SHE
```

Definition at line 95 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.9 EHSM\_FAST\_CMACE\_SM4

```
#define EHSM_FAST_CMACE_SM4 2
```

Definition at line 88 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.10 EHSM\_FAST\_CMACE\_VERIFY

```
#define EHSM_FAST_CMACE_VERIFY 1
```

Definition at line 92 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.11 EHSM\_GET\_STATUS\_ERRORS

```
#define EHSM_GET_STATUS_ERRORS 0x4
```

Definition at line 36 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.12 EHSM\_GET\_STATUS\_MEM

```
#define EHSM_GET_STATUS_MEM 0x2
```

Definition at line 34 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.13 EHSM\_GET\_STATUS\_SBB

```
#define EHSM_GET_STATUS_SBB 0x1
```

Definition at line 32 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.14 EHSM\_GET\_STATUS\_SHE

```
#define EHSM_GET_STATUS_SHE 0x0
```

Definition at line 30 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.15 EHSM\_SELF\_TEST\_ALL

```
#define EHSM_SELF_TEST_ALL
```

**Value:**

```
(EHSM_SELF_TEST_TRNG | \
 EHSM_SELF_TEST_SKE | \
 EHSM_SELF_TEST_PKE | \
 EHSM_SELF_TEST_HASH)
```

Definition at line 81 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.16 EHSM\_SELF\_TEST\_HASH

```
#define EHSM_SELF_TEST_HASH
```

**Value:**

```
(EHSM_SELF_TEST_HASH_MD5 | EHSM_SELF_TEST_HASH_SHA1 |
 EHSM_SELF_TEST_HASH_SHA2 | \
 EHSM_SELF_TEST_HASH_SHA3 |
 EHSM_SELF_TEST_HASH_SM3 | EHSM_SELF_TEST_HASH_SHA256)
```

Definition at line 77 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.17 EHSM\_SELF\_TEST\_HASH\_MD5

```
#define EHSM_SELF_TEST_HASH_MD5 (0x1U << 16)
```

Definition at line 58 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.1.18 EHSM\_SELF\_TEST\_HASH\_SHA1

```
#define EHSM_SELF_TEST_HASH_SHA1 (0x1U << 17)
```

Definition at line 59 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.19 EHSM\_SELF\_TEST\_HASH\_SHA2

```
#define EHSM_SELF_TEST_HASH_SHA2 (0x1U << 18)
```

Definition at line 60 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.20 EHSM\_SELF\_TEST\_HASH\_SHA256

```
#define EHSM_SELF_TEST_HASH_SHA256 (0x1U << 21)
```

Definition at line 63 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.21 EHSM\_SELF\_TEST\_HASH\_SHA3

```
#define EHSM_SELF_TEST_HASH_SHA3 (0x1U << 19)
```

Definition at line 61 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.22 EHSM\_SELF\_TEST\_HASH\_SM3

```
#define EHSM_SELF_TEST_HASH_SM3 (0x1U << 20)
```

Definition at line 62 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.23 EHSM\_SELF\_TEST\_PKE

```
#define EHSM_SELF_TEST_PKE
```

**Value:**

```
(EHSM_SELF_TEST_PKE_RSA | EHSM_SELF_TEST_PKE_ECC | \
 EHSM_SELF_TEST_PKE_SM2 | \
 EHSM_SELF_TEST_PKE_SM9)
```

Definition at line 73 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.24 EHSM\_SELF\_TEST\_PKE\_ECC

```
#define EHSM_SELF_TEST_PKE_ECC (0x1U << 9)
```

Definition at line 51 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.25 EHSM\_SELF\_TEST\_PKE\_RSA

```
#define EHSM_SELF_TEST_PKE_RSA (0x1U << 8)
```

Definition at line 49 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.26 EHSM\_SELF\_TEST\_PKE\_SM2

```
#define EHSM_SELF_TEST_PKE_SM2 (0x1U << 10)
```

Definition at line 53 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.27 EHSM\_SELF\_TEST\_PKE\_SM9

```
#define EHSM_SELF_TEST_PKE_SM9 (0x1U << 11)
```

Definition at line 55 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.28 EHSM\_SELF\_TEST\_SKE

```
#define EHSM_SELF_TEST_SKE
```

**Value:**

```
(EHSM_SELF_TEST_SKE_DES | EHSM_SELF_TEST_SKE_TDES | \
 EHSM_SELF_TEST_SKE_AES |
 EHSM_SELF_TEST_SKE_SM4)
```

Definition at line 69 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.29 EHSM\_SELF\_TEST\_SKE\_AES

```
#define EHSM_SELF_TEST_SKE_AES (0x1U << 2)
```

Definition at line 44 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.30 EHSM\_SELF\_TEST\_SKE\_DES

```
#define EHSM_SELF_TEST_SKE_DES (0x1U << 0)
```

Definition at line 40 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.31 EHSM\_SELF\_TEST\_SKE\_SM4

```
#define EHSM_SELF_TEST_SKE_SM4 (0x1U << 3)
```

Definition at line 46 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.32 EHSM\_SELF\_TEST\_SKE\_TDES

```
#define EHSM_SELF_TEST_SKE_TDES (0x1U << 1)
```

Definition at line 42 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.33 EHSM\_SELF\_TEST\_TRNG

```
#define EHSM_SELF_TEST_TRNG (0x1U << 24)
```

Definition at line 66 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.34 EHSM\_SHE\_M1\_MAX\_SIZE

```
#define EHSM_SHE_M1_MAX_SIZE (32)
```

Definition at line 20 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.35 EHSM\_SHE\_M2\_MAX\_SIZE

```
#define EHSM_SHE_M2_MAX_SIZE (32)
```

Definition at line 21 of file eHSM\_Com\_Struct\_lp.h.

**4.4.1.36 EHSM\_SHE\_M3\_MAX\_SIZE**

```
#define EHSM_SHE_M3_MAX_SIZE (16)
```

Definition at line 22 of file eHSM\_Com\_Struct\_lp.h.

**4.4.1.37 EHSM\_SHE\_M4\_MAX\_SIZE**

```
#define EHSM_SHE_M4_MAX_SIZE (48)
```

Definition at line 23 of file eHSM\_Com\_Struct\_lp.h.

**4.4.1.38 EHSM\_SHE\_M5\_MAX\_SIZE**

```
#define EHSM_SHE_M5_MAX_SIZE (16)
```

Definition at line 24 of file eHSM\_Com\_Struct\_lp.h.

**4.4.1.39 IMAGE\_ANALYSIS\_CODE**

```
#define IMAGE_ANALYSIS_CODE (0x55)
```

Definition at line 112 of file eHSM\_Com\_Struct\_lp.h.

**4.4.1.40 IMAGE\_DECRYPT\_CODE**

```
#define IMAGE_DECRYPT_CODE (0x5A)
```

Definition at line 110 of file eHSM\_Com\_Struct\_lp.h.

**4.4.1.41 IMAGE\_ENCRYPT\_CODE**

```
#define IMAGE_ENCRYPT_CODE (0x5C)
```

Definition at line 111 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.42 IMAGE\_PUBLIC\_KEY\_MAX\_LENGTH

```
#define IMAGE_PUBLIC_KEY_MAX_LENGTH (64U+256U)
```

Definition at line 102 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.43 IMAGE\_SIGNATURE\_MAX\_LENGTH

```
#define IMAGE_SIGNATURE_MAX_LENGTH (256U)
```

Definition at line 101 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.44 OTP\_CONTROL\_FILED\_BYTE\_SIZE

```
#define OTP_CONTROL_FILED_BYTE_SIZE (8)
```

Definition at line 18 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.45 SECURE\_BOOT\_TYPE\_IMAGE\_VERIFY

```
#define SECURE_BOOT_TYPE_IMAGE_VERIFY (0x01U)
```

Definition at line 98 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.46 SECURE\_BOOT\_TYPE\_SECURE\_BOOT

```
#define SECURE_BOOT_TYPE_SECURE_BOOT (0x02U)
```

Definition at line 99 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.47 SOC\_BOOT\_TYPE\_PARALLEL

```
#define SOC_BOOT_TYPE_PARALLEL 0x1U
```

Definition at line 119 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.48 SOC\_BOOT\_TYPE\_SEQUENTIAL

```
#define SOC_BOOT_TYPE_SEQUENTIAL 0x0U
```

Definition at line 118 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.49 SOC\_CODE\_VERIFY\_FALG

```
#define SOC_CODE_VERIFY_FALG (0x02)
```

Definition at line 105 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.1.50 UPGRADE\_VALID\_FLAG

```
#define UPGRADE_VALID_FLAG (0x71689BA2UL)
```

Definition at line 108 of file eHSM\_Com\_Struct\_lp.h.

### 4.4.2 Typedef Documentation

#### 4.4.2.1 ehsm\_create\_dh\_key\_param\_st

```
typedef struct ehsm_create_dh_key_param ehsm_create_dh_key_param_st
```

#### 4.4.2.2 ehsm\_create\_evita\_key\_param\_st

```
typedef struct ehsm_create_evita_key_param ehsm_create_evita_key_param_st
```

#### 4.4.2.3 ehsm\_create\_random\_key\_param\_st

```
typedef struct ehsm_create_random_key_param ehsm_create_random_key_param_st
```

#### 4.4.2.4 ehsm\_crypto\_randomgenerate\_param\_st

```
typedef struct ehsm_crypto_randomgenerate_param ehsm_crypto_randomgenerate_param_st
```



#### 4.4.2.5 ehsm\_emu\_status\_st

```
typedef struct ehsm_emu_status_st_ ehsm_emu_status_st
```

#### 4.4.2.6 ehsm\_evita\_key\_export\_st

```
typedef struct ehsm_evita_key_export ehsm_evita_key_export_st
```

#### 4.4.2.7 ehsm\_evita\_memory\_info\_st

```
typedef struct ehsm_evita_memory_info_st_ ehsm_evita_memory_info_st
```

#### 4.4.2.8 ehsm\_exchange\_sm9\_key\_param\_st

```
typedef struct ehsm_exchange_sm9_key_param ehsm_exchange_sm9_key_param_st
```

#### 4.4.2.9 ehsm\_fw\_encrypt\_key\_st

```
typedef struct ehsm_fw_encrypt_key ehsm_fw_encrypt_key_st
```

#### 4.4.2.10 ehsm\_fw\_random\_key\_st

```
typedef struct ehsm_fw_random_key ehsm_fw_random_key_st
```

#### 4.4.2.11 ehsm\_gen\_sm9\_key\_param\_st

```
typedef struct ehsm_gen_sm9_key_param ehsm_gen_sm9_key_param_st
```

#### 4.4.2.12 ehsm\_gen\_sm9\_master\_key\_param\_st

```
typedef struct ehsm_gen_sm9_master_key_param ehsm_gen_sm9_master_key_param_st
```

#### 4.4.2.13 ehsm\_gen\_sm9\_userpriv\_key\_param\_st

```
typedef struct ehsm_gen_sm9_userpriv_key_param ehsm_gen_sm9_userpriv_key_param_st
```

#### 4.4.2.14 ehsm\_get\_pub\_from\_priv\_param\_st

```
typedef struct ehsm_get_pub_from_priv_param ehsm_get_pub_from_priv_param_st
```

#### 4.4.2.15 ehsm\_image\_upgrade\_st

```
typedef struct ehsm_image ehsm_image_upgrade_st
```

#### 4.4.2.16 ehsm\_key\_copy\_param\_st

```
typedef struct ehsm_key_copy_param ehsm_key_copy_param_st
```

#### 4.4.2.17 ehsm\_key\_derived\_param\_st

```
typedef struct ehsm_key_derived_param ehsm_key_derived_param_st
```

#### 4.4.2.18 ehsm\_key\_remove\_param\_st

```
typedef struct ehsm_key_remove_param ehsm_key_remove_param_st
```

#### 4.4.2.19 ehsm\_key\_status\_param\_st

```
typedef struct ehsm_key_status_param ehsm_key_status_param_st
```

#### 4.4.2.20 ehsm\_keyexchange\_key\_info\_st

```
typedef struct ehsm_keyexchange_key_info ehsm_keyexchange_key_info_st
```

**4.4.2.21 ehsm\_she\_key\_host\_param\_st**

```
typedef struct ehsm_she_key_host_param ehsm_she_key_host_param_st
```

**4.4.2.22 ehsm\_she\_key\_param\_st**

```
typedef struct ehsm_she_key_param ehsm_she_key_param_st
```

**4.4.2.23 ehsm\_she\_plain\_key\_host\_param\_st**

```
typedef struct ehsm_she_plain_key_host_param ehsm_she_plain_key_host_param_st
```

**4.4.2.24 ehsm\_she\_plain\_key\_param\_st**

```
typedef struct ehsm_she_plain_key_param ehsm_she_plain_key_param_st
```

**4.4.2.25 ehsm\_sm9\_exckey\_gen\_tmpkey\_st**

```
typedef struct ehsm_sm9_exckey_gen_tmpkey_param ehsm_sm9_exckey_gen_tmpkey_st
```

**4.4.2.26 ehsm\_sm9\_gen\_mast\_pubkey\_st**

```
typedef struct ehsm_sm9_gen_mast_pubkey ehsm_sm9_gen_mast_pubkey_st
```

**4.4.2.27 ehsm\_sm9\_gen\_tmp\_pubkey\_st**

```
typedef struct ehsm_sm9_gen_tmp_pubkey_param ehsm_sm9_gen_tmp_pubkey_st
```

**4.4.2.28 ehsm\_sm9\_inexport\_key\_param\_st**

```
typedef struct ehsm_sm9_inexport_key_param ehsm_sm9_inexport_key_param_st
```

#### 4.4.2.29 ehsm\_sm9\_unwrap\_key\_param\_st

```
typedef struct ehsm_sm9_unwrap_key_param ehsm_sm9_unwrap_key_param_st
```

#### 4.4.2.30 ehsm\_sm9\_wrap\_key\_param\_st

```
typedef struct ehsm_sm9_wrap_key_param ehsm_sm9_wrap_key_param_st
```

#### 4.4.2.31 ehsm\_soc\_image\_upgrade\_info\_st

```
typedef struct soc_image_upgrade_info ehsm_soc_image_upgrade_info_st
```

#### 4.4.2.32 ehsm\_soc\_image\_upgrade\_input\_st

```
typedef struct soc_image_upgrade_input ehsm_soc_image_upgrade_input_st
```

#### 4.4.2.33 ehsm\_soc\_image\_verify\_info\_st

```
typedef struct soc_image_verify_info ehsm_soc_image_verify_info_st
```

#### 4.4.2.34 ehsm\_soc\_image\_verify\_input\_st

```
typedef struct soc_image_verify_input ehsm_soc_image_verify_input_st
```

#### 4.4.2.35 sm2\_ext\_param\_st

```
typedef struct sm2_ext_param sm2_ext_param_st
```

### 4.4.3 Enumeration Type Documentation

#### 4.4.3.1 crypto\_key\_derive\_type\_e

```
enum crypto_key_derive_type_e
```

**Enumerator**

|                                  |  |
|----------------------------------|--|
| CRYPTO_KEY_DERIVE_USER_PASSWD    |  |
| CRYPTO_KEY_DERIVE_USER_KEYHANDLE |  |

Definition at line 223 of file eHSM\_Com\_Struct\_Ip.h.

**4.4.3.2 ehsm\_api\_type\_e**

enum `ehsm_api_type_e`

**Enumerator**

|                       |  |
|-----------------------|--|
| EHSM_API_TYPE_SHE     |  |
| EHSM_API_TYPE_EVITA   |  |
| EHSM_API_TYPE_AUTOSAR |  |
| EHSM_API_TYPE_EXT     |  |
| EHSM_API_TYPE_INVALID |  |

Definition at line 1034 of file eHSM\_Com\_Struct\_Ip.h.

**4.4.3.3 ehsm\_challenge\_type\_e**

enum `ehsm_challenge_type_e`

**Enumerator**

|                                |  |
|--------------------------------|--|
| EHSM_CHALLENGE_TYPE_INVALID    |  |
| EHSM_CHALLENGE_TYPE_TIME_SYNC  |  |
| EHSM_CHALLENGE_TYPE_EHSM_DEBUG |  |
| EHSM_CHALLENGE_TYPE_SHE_DEBUG  |  |
| EHSM_CHALLENGE_TYPE_SOC_DEBUG  |  |
| EHSM_CHALLENGE_TYPE_USER_AUTH  |  |
| EHSM_CHALLENGE_TYPE_MAX        |  |

Definition at line 482 of file eHSM\_Com\_Struct\_Ip.h.

**4.4.3.4 ehsm\_code\_upgrade\_alg\_e**

enum `ehsm_code_upgrade_alg_e`

**Enumerator**

|                      |  |
|----------------------|--|
| CODE_UPGRADE_ALG_RSA |  |
|----------------------|--|

## Enumerator

|                              |  |
|------------------------------|--|
| CODE_UPGRADE_ALG_SM2         |  |
| CODE_UPGRADE_ALG_AES128_GCM  |  |
| CODE_UPGRADE_ALG_SM4_GCM     |  |
| CODE_UPGRADE_ALG_AES128_CMAC |  |
| CODE_UPGRADE_ALG_SM4_CMAC    |  |
| CODE_UPGRADE_ALG_INVALID     |  |

Definition at line 721 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.5 ehsm\_code\_verify\_alg\_e

```
enum ehsm_code_verify_alg_e
```

## Enumerator

|                             |  |
|-----------------------------|--|
| CODE_VERIFY_ALG_RSA         |  |
| CODE_VERIFY_ALG_SM2         |  |
| CODE_VERIFY_ALG_AES128_CMAC |  |
| CODE_VERIFY_ALG_SM4_CMAC    |  |
| CODE_VERIFY_ALG_INVALID     |  |

Definition at line 712 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.6 ehsm\_control\_field\_type\_e

```
enum ehsm_control_field_type_e
```

## Enumerator

|                              |  |
|------------------------------|--|
| EHSM_CONTROL_FIELD_TYPE_HW   |  |
| EHSM_CONTROL_FIELD_TYPE_EHSM |  |
| EHSM_CONTROL_FIELD_TYPE_SOC  |  |

Definition at line 949 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.7 ehsm\_debug\_auth\_alg\_e

```
enum ehsm_debug_auth_alg_e
```

## Enumerator

|                                  |  |
|----------------------------------|--|
| EHSM_DEBUG_AUTH_ALG_SM2_WITH_SM3 |  |
|----------------------------------|--|

## Enumerator

|                                              |  |
|----------------------------------------------|--|
| EHSM_DEBUG_AUTH_ALG_ECCSECP256R1_WITH_SHA256 |  |
| EHSM_DEBUG_AUTH_ALG_SM4_CMAC                 |  |
| EHSM_DEBUG_AUTH_ALG_AES128_CMAC              |  |

Definition at line 493 of file eHSM\_Com\_Struct\_Ip.h.

## 4.4.3.8 ehsm\_dh\_mode\_e

enum [ehsm\\_dh\\_mode\\_e](#)

## Enumerator

|                          |  |
|--------------------------|--|
| EHSM_DH_MODE_KEY_HANDLE  |  |
| EHSM_DH_MODE_KEY_PUB_KEY |  |
| EHSM_DH_MODE_RAW_PUB_KEY |  |

Definition at line 160 of file eHSM\_Com\_Struct\_Ip.h.

## 4.4.3.9 ehsm\_fw\_encrypt\_key\_slot\_e

enum [ehsm\\_fw\\_encrypt\\_key\\_slot\\_e](#)

## Enumerator

|                                                 |  |
|-------------------------------------------------|--|
| EHSM_FW_ENCRYPT_KEY_SLOT_SOC_DEBUG_KEY          |  |
| EHSM_FW_ENCRYPT_KEY_SLOT_SOC_FW_VERIFY_KEY      |  |
| EHSM_FW_ENCRYPT_KEY_SLOT_SOC_UPGRADE_ENC_KEY    |  |
| EHSM_FW_ENCRYPT_KEY_SLOT_SOC_UPGRADE_VERIFY_KEY |  |
| EHSM_FW_ENCRYPT_KEY_SLOT_USER_DEBUG_KEY         |  |

Definition at line 540 of file eHSM\_Com\_Struct\_Ip.h.

## 4.4.3.10 ehsm\_fw\_encrypt\_key\_type\_e

enum [ehsm\\_fw\\_encrypt\\_key\\_type\\_e](#)

## Enumerator

|                                          |  |
|------------------------------------------|--|
| EHSM_FW_ENCRYPT_KEY_TYPE_SYMMETRIC_KEY   |  |
| EHSM_FW_ENCRYPT_KEY_TYPE_PUBLIC_KEY_HASH |  |

Definition at line 534 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.3.11 ehsm\_fw\_random\_key\_slot\_e

enum [ehsm\\_fw\\_random\\_key\\_slot\\_e](#)

##### Enumerator

|                                           |  |
|-------------------------------------------|--|
| EHSM_FW_RANDOM_KEY_SLOT_DEVICE_ROOT_KEY   |  |
| EHSM_FW_RANDOM_KEY_SLOT_SOC_FW_VERIFY_KEY |  |
| EHSM_FW_RANDOM_KEY_SLOT_SOC_ENC_KEY       |  |
| EHSM_FW_RANDOM_KEY_SLOT_SOC_PRIVATE_KEY   |  |
| EHSM_FW_RANDOM_KEY_SLOT_USER_ROOT_KEY     |  |

Definition at line 525 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.3.12 ehsm\_fw\_random\_key\_type\_e

enum [ehsm\\_fw\\_random\\_key\\_type\\_e](#)

##### Enumerator

|                                               |  |
|-----------------------------------------------|--|
| EHSM_FW_RANDOM_KEY_TYPE_SYMMETRIC_KEY         |  |
| EHSM_FW_RANDOM_KEY_TYPE_SM2_PRIVATE_KEY       |  |
| EHSM_FW_RANDOM_KEY_TYPE_SECP256R1_PRIVATE_KEY |  |

Definition at line 518 of file eHSM\_Com\_Struct\_lp.h.

#### 4.4.3.13 ehsm\_gen\_sm9\_key\_type\_e

enum [ehsm\\_gen\\_sm9\\_key\\_type\\_e](#)

##### Enumerator

|                          |  |
|--------------------------|--|
| EHSM_GEN_SM9_MASTER_KEY  |  |
| EHSM_GEN_SM9_PRIV_KEY    |  |
| EHSM_GEN_SM9_INVALID_KEY |  |

Definition at line 794 of file eHSM\_Com\_Struct\_lp.h.



## 4.4.3.14 ehsm\_image\_process\_mode\_e

```
enum ehsm_image_process_mode_e
```

## Enumerator

|                                 |  |
|---------------------------------|--|
| EHSM_IMAGE_PROCESS_MODE_INIT    |  |
| EHSM_IMAGE_PROCESS_MODE_UPDATE  |  |
| EHSM_IMAGE_PROCESS_MODE_FINISH  |  |
| EHSM_IMAGE_PROCESS_MODE_ONEPASS |  |

Definition at line 563 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.15 ehsm\_key\_mem\_type\_e

```
enum ehsm_key_mem_type_e
```

## Enumerator

|                         |  |
|-------------------------|--|
| EHSM_EVITA_KEY_TYPE_NVM |  |
| EHSM_EVITA_KEY_TYPE_RAM |  |

Definition at line 154 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.16 ehsm\_lifecycle\_e

```
enum ehsm_lifecycle_e
```

## Enumerator

|                               |  |
|-------------------------------|--|
| EHSM_LIFE_CYCLE_UNNORMAL_MODE |  |
| EHSM_LIFE_CYCLE_TEST_MODE     |  |
| EHSM_LIFE_CYCLE_DEV_MODE      |  |
| EHSM_LIFE_CYCLE_MANU_MODE     |  |
| EHSM_LIFE_CYCLE_USER_MODE     |  |
| EHSM_LIFE_CYCLE_DEBUG_MODE    |  |
| EHSM_LIFE_CYCLE_DESTROY_MODE  |  |

Definition at line 938 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.17 ehsm\_rsa\_key\_type\_e

```
enum ehsm_rsa_key_type_e
```

## Enumerator

|                          |  |
|--------------------------|--|
| EHSM_RSA_KEY_TYPE_COMMON |  |
| EHSM_RSA_KEY_TYPE_CRT    |  |

Definition at line 170 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.18 ehsm\_SM9\_exchg\_key\_role\_e

```
enum ehsm_SM9_exchg_key_role_e
```

## Enumerator

|                              |  |
|------------------------------|--|
| EHSM_SM9_EXCHG_KEY_ROLE_SELF |  |
| EHSM_SM9_EXCHG_KEY_ROLE_PEER |  |

Definition at line 771 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.19 ehsm\_sm9\_master\_key\_type\_e

```
enum ehsm_sm9_master_key_type_e
```

## Enumerator

|                                 |  |
|---------------------------------|--|
| EHSM_GEN_SM9_SIGN_MASTER_KEY    |  |
| EHSM_GEN_SM9_ENC_MASTER_KEY     |  |
| EHSM_GEN_SM9_EXCHG_MASTER_KEY   |  |
| EHSM_GEN_SM9_INVALID_MASTER_KEY |  |

Definition at line 777 of file eHSM\_Com\_Struct\_lp.h.

## 4.4.3.20 ehsm\_sm9\_user\_privkey\_type\_e

```
enum ehsm_sm9_user_privkey_type_e
```

## Enumerator

|                                   |  |
|-----------------------------------|--|
| EHSM_GEN_SM9_SIGN_USERPRIV_KEY    |  |
| EHSM_GEN_SM9_ENC_USERPRIV_KEY     |  |
| EHSM_GEN_SM9_EXCHG_USERPRIV_KEY   |  |
| EHSM_GEN_SM9_EXCHG_USERTMP_KEY    |  |
| EHSM_GEN_SM9_INVALID_USERPRIV_KEY |  |

Definition at line 785 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.3.21 ehsm\_uart\_baudrate\_e

```
enum ehsm_uart_baudrate_e
```

##### Enumerator

|                            |  |
|----------------------------|--|
| EHSM_UART_BAUDRATE_9600    |  |
| EHSM_UART_BAUDRATE_19200   |  |
| EHSM_UART_BAUDRATE_38400   |  |
| EHSM_UART_BAUDRATE_57600   |  |
| EHSM_UART_BAUDRATE_115200  |  |
| EHSM_UART_BAUDRATE_INVALID |  |

Definition at line 183 of file eHSM\_Com\_Struct\_Ip.h.

#### 4.4.3.22 sm2\_key\_exchange\_role\_e

```
enum sm2_key_exchange_role_e
```

##### Enumerator

|                                 |  |
|---------------------------------|--|
| SM2_KEY_EXCHANGE_ROLE_SPONSOR   |  |
| SM2_KEY_EXCHANGE_ROLE_RESPONSOR |  |

Definition at line 178 of file eHSM\_Com\_Struct\_Ip.h.

## 4.5 eHSM\_Compt\_Bitmap.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include <stddef.h>
#include "eHSM_Compt_Bitmap.h"
#include "eHSM_Types_Ip.h"
```

## Functions

- `bitmap_st * ehsm_bitmap_init` (void \*bit\_ptr, ehsm\_uint32\_t bit\_max)  
*initialize a bitmap [in] bits of the bitmap [in] memory for bitmap\_st*
- void `ehsm_bitmap_reset` (bitmap\_st \*bmap)  
*clear all the bits of the bitmap [in] bitmap that will be reset*
- void `ehsm_bitmap_set` (bitmap\_st \*bmap, ehsm\_uint32\_t bit)

- set a bit of the bitmap [in] bitmap that will be set*
  - void `ehsm_bitmap_clr` (`bitmap_st *bmap`, `ehsm_uint32_t bit`)
- clear a bit of the bitmap [in] bitmap that will be clear*
  - `ehsm_uint32_t ehsm_bitmap_count` (`bitmap_st *bmap`)
- get the bits number that has been set [in] bitmap object*
  - `ehsm_uint32_t ehsm_bitmap_first` (`bitmap_st *bmap`)
- get the first bits index that have been set [in] bitmap object*
  - `ehsm_uint32_t ehsm_bitmap_last` (`bitmap_st *bmap`)
- get the last bits index that have been set [in] bitmap object*

## 4.5.1 Function Documentation

### 4.5.1.1 ehsm\_bitmap\_clr()

```
void ehsm_bitmap_clr (
 bitmap_st * bmap,
 ehsm_uint32_t bit)
```

clear a bit of the bitmap [in] bitmap that will be clear

Definition at line 87 of file eHSM\_Compt\_Bitmap.c.

### 4.5.1.2 ehsm\_bitmap\_count()

```
ehsm_uint32_t ehsm_bitmap_count (
 bitmap_st * bmap)
```

get the bits number that has been set [in] bitmap object

#### Returns

number of bits that has been set

Definition at line 98 of file eHSM\_Compt\_Bitmap.c.

### 4.5.1.3 ehsm\_bitmap\_first()

```
ehsm_uint32_t ehsm_bitmap_first (
 bitmap_st * bmap)
```

get the first bits index that have been set [in] bitmap object

#### Returns

the first bits index that have been set, no bit has been when return 0

Definition at line 119 of file eHSM\_Compt\_Bitmap.c.

#### 4.5.1.4 ehsm\_bitmap\_init()

```
bitmap_st* ehsm_bitmap_init (
 void * bit_ptr,
 ehsm_uint32_t bit_max)
```

initialize a bitmap [in] bits of the bitmap [in] memory for bitmap\_st

##### Returns

Failure when return NULL, otherwise success

Definition at line 49 of file eHSM\_Compt\_Bitmap.c.

#### 4.5.1.5 ehsm\_bitmap\_last()

```
ehsm_uint32_t ehsm_bitmap_last (
 bitmap_st * bmap)
```

get the last bits index that have been set [in] bitmap object

##### Returns

the last bits index that have been set, no bit has been when return 0

Definition at line 147 of file eHSM\_Compt\_Bitmap.c.

#### 4.5.1.6 ehsm\_bitmap\_reset()

```
void ehsm_bitmap_reset (
 bitmap_st * bmap)
```

clear all the bits of the bitmap [in] bitmap that will be reset

Definition at line 65 of file eHSM\_Compt\_Bitmap.c.

#### 4.5.1.7 ehsm\_bitmap\_set()

```
void ehsm_bitmap_set (
 bitmap_st * bmap,
 ehsm_uint32_t bit)
```

set a bit of the bitmap [in] bitmap that will be set

Definition at line 76 of file eHSM\_Compt\_Bitmap.c.

## 4.6 eHSM\_Compt\_Bitmap.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Types_Ip.h"
```

### Classes

- struct [bitmap](#)

### Typedefs

- typedef struct [bitmap](#) [bitmap\\_st](#)

### Functions

- [bitmap\\_st \\* ehsm\\_bitmap\\_create](#) ([ehsm\\_uint32\\_t](#) bit\_max)  
*create a bitmap [in] bits of the bitmap*
- [bitmap\\_st \\* ehsm\\_bitmap\\_init](#) (void \*bit\_ptr, [ehsm\\_uint32\\_t](#) bit\_max)  
*initialize a bitmap [in] bits of the bitmap [in] memory for bitmap\_st*
- void [ehsm\\_bitmap\\_destroy](#) ([bitmap\\_st](#) \*bmap)  
*destroy a bitmap [in] bitmap the will be destroyed*
- void [ehsm\\_bitmap\\_reset](#) ([bitmap\\_st](#) \*bmap)  
*clear all the bits of the bitmap [in] bitmap that will be reset*
- void [ehsm\\_bitmap\\_set](#) ([bitmap\\_st](#) \*bmap, [ehsm\\_uint32\\_t](#) bit)  
*set a bit of the bitmap [in] bitmap that will be set*
- void [ehsm\\_bitmap\\_clr](#) ([bitmap\\_st](#) \*bmap, [ehsm\\_uint32\\_t](#) bit)  
*clear a bit of the bitmap [in] bitmap that will be clear*
- [ehsm\\_uint32\\_t ehsm\\_bitmap\\_count](#) ([bitmap\\_st](#) \*bmap)  
*get the bits number that has been set [in] bitmap object*
- [ehsm\\_uint32\\_t ehsm\\_bitmap\\_first](#) ([bitmap\\_st](#) \*bmap)  
*get the first bits index that have been set [in] bitmap object*
- [ehsm\\_uint32\\_t ehsm\\_bitmap\\_last](#) ([bitmap\\_st](#) \*bmap)  
*get the last bits index that have been set [in] bitmap object*

### 4.6.1 Typedef Documentation

#### 4.6.1.1 [bitmap\\_st](#)

```
typedef struct bitmap bitmap_st
```

### 4.6.2 Function Documentation

#### 4.6.2.1 ehsm\_bitmap\_clr()

```
void ehsm_bitmap_clr (
 bitmap_st * bmap,
 ehsm_uint32_t bit)
```

clear a bit of the bitmap [in] bitmap that will be clear

Definition at line 87 of file eHSM\_Compt\_Bitmap.c.

#### 4.6.2.2 ehsm\_bitmap\_count()

```
ehsm_uint32_t ehsm_bitmap_count (
 bitmap_st * bmap)
```

get the bits number that has been set [in] bitmap object

##### Returns

number of bits that has been set

Definition at line 98 of file eHSM\_Compt\_Bitmap.c.

#### 4.6.2.3 ehsm\_bitmap\_create()

```
bitmap_st* ehsm_bitmap_create (
 ehsm_uint32_t bit_max)
```

create a bitmap [in] bits of the bitmap

##### Returns

Failure when return NULL, otherwise success

#### 4.6.2.4 ehsm\_bitmap\_destroy()

```
void ehsm_bitmap_destroy (
 bitmap_st * bmap)
```

destroy a bitmap [in] bitmap the will be destroyed

#### 4.6.2.5 ehsm\_bitmap\_first()

```
ehsm_uint32_t ehsm_bitmap_first (
 bitmap_st * bmap)
```

get the first bits index that have been set [in] bitmap object

##### Returns

the first bits index that have been set, no bit has been when return 0

Definition at line 119 of file eHSM\_Compt\_Bitmap.c.

#### 4.6.2.6 ehsm\_bitmap\_init()

```
bitmap_st* ehsm_bitmap_init (
 void * bit_ptr,
 ehsm_uint32_t bit_max)
```

initialize a bitmap [in] bits of the bitmap [in] memory for bitmap\_st

##### Returns

Failure when return NULL, otherwise success

Definition at line 49 of file eHSM\_Compt\_Bitmap.c.

#### 4.6.2.7 ehsm\_bitmap\_last()

```
ehsm_uint32_t ehsm_bitmap_last (
 bitmap_st * bmap)
```

get the last bits index that have been set [in] bitmap object

##### Returns

the last bits index that have been set, no bit has been when return 0

Definition at line 147 of file eHSM\_Compt\_Bitmap.c.

#### 4.6.2.8 ehsm\_bitmap\_reset()

```
void ehsm_bitmap_reset (
 bitmap_st * bmap)
```

clear all the bits of the bitmap [in] bitmap that will be reset

Definition at line 65 of file eHSM\_Compt\_Bitmap.c.



## 4.6.2.9 ehsm\_bitmap\_set()

```
void ehsm_bitmap_set (
 bitmap_st * bmap,
 ehsm_uint32_t bit)
```

set a bit of the bitmap [in] bitmap that will be set

Definition at line 76 of file eHSM\_Compt\_Bitmap.c.

## 4.7 eHSM\_Compt\_List.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
```

## Classes

- struct [dlist\\_head](#)

## Macros

- #define [OFFSET](#)(TYPE, MEMBER) ((size\_t) &((TYPE \*)0)->MEMBER)
- #define [container\\_of](#)(ptr, type, member) (type \*)((char \*)ptr -[OFFSET](#)(type,member))
- #define [DLIST\\_POISON1](#) ((void \*) 0x00100100)
- #define [DLIST\\_POISON2](#) ((void \*) 0x00200200)
- #define [dlist\\_entry](#)(ptr, type, member) [container\\_of](#)(ptr, type, member)
- #define [DLIST\\_HEAD\\_INIT](#)(name) { &(name), &(name) }
- #define [DLIST\\_HEAD](#)(name) struct [dlist\\_head](#) name = [DLIST\\_HEAD\\_INIT](#)(name)
- #define [dlist\\_for\\_each](#)(pos, head) for (pos = (head)->next; pos != (head); pos = pos->next)
- #define [dlist\\_for\\_each\\_safe](#)(pos, n, head)

## 4.7.1 Macro Definition Documentation

## 4.7.1.1 container\_of

```
#define container_of(
 ptr,
 type,
 member) (type *) ((char *)ptr -OFFSET(type,member))
```

Definition at line 19 of file eHSM\_Compt\_List.h.

#### 4.7.1.2 dlist\_entry

```
#define dlist_entry(
 ptr,
 type,
 member) container_of(ptr, type, member)
```

Definition at line 24 of file eHSM\_Compt\_List.h.

#### 4.7.1.3 dlist\_for\_each

```
#define dlist_for_each(
 pos,
 head) for (pos = (head)->next; pos != (head); pos = pos->next)
```

Definition at line 32 of file eHSM\_Compt\_List.h.

#### 4.7.1.4 dlist\_for\_each\_safe

```
#define dlist_for_each_safe(
 pos,
 n,
 head)
```

**Value:**

```
for (pos = (head)->next, n = pos->next; pos != (head); \
 pos = n, n = pos->next)
```

Definition at line 88 of file eHSM\_Compt\_List.h.

#### 4.7.1.5 DLIST\_HEAD

```
#define DLIST_HEAD(
 name) struct dlist_head name = DLIST_HEAD_INIT(name)
```

Definition at line 29 of file eHSM\_Compt\_List.h.

#### 4.7.1.6 DLIST\_HEAD\_INIT

```
#define DLIST_HEAD_INIT(
 name) { &(name), &(name) }
```

Definition at line 27 of file eHSM\_Compt\_List.h.

## 4.7.1.7 DLIST\_POISON1

```
#define DLIST_POISON1 ((void *) 0x00100100)
```

Definition at line 21 of file eHSM\_Compt\_List.h.

## 4.7.1.8 DLIST\_POISON2

```
#define DLIST_POISON2 ((void *) 0x00200200)
```

Definition at line 22 of file eHSM\_Compt\_List.h.

## 4.7.1.9 OFFSET

```
#define OFFSET(
 TYPE,
 MEMBER) ((size_t) &((TYPE *)0)->MEMBER)
```

Definition at line 17 of file eHSM\_Compt\_List.h.

## 4.8 eHSM\_Config\_Ip.h File Reference

## Macros

- #define [CONFIG\\_EHSM\\_AUTOSAR](#)  
*configuration for AUTOSAR support*
- #define [CONFIG\\_EHSM\\_EVITA](#)  
*configuration for EVITA support*
- #define [CONFIG\\_EHSM\\_SHE](#)  
*configuration for SHE support*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CRYPTOBJ\\_PKE\\_QUEUE\\_SIZE](#) 10U  
*configuration for SE support*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CRYPTOBJ\\_TRNG\\_QUEUE\\_SIZE](#) 10U
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CRYPTOBJ\\_HASH\\_QUEUE\\_SIZE](#) 10U
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CRYPTOBJ\\_K\\_QUEUE\\_SIZE](#) 10U
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CRYPTOBJ\\_SYSMGR\\_QUEUE\\_SIZE](#) 10U
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CRYPTOBJ\\_SKE\\_QUEUE\\_SIZE](#) 10U
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_CMD\\_QUEUE\\_SIZE](#) 2U  
*command queue size in eHSM for each crypto object*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_DEFAULT\\_CMD\\_TIMEOUT](#) 0x1FFCFFFFU  
*time out for Synchronous command, millisecond.*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_RSA\\_K\\_CMD\\_TIMEOUT](#) 0x1FFFFFFFU  
*time out for RSA key generation.*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_MAILBOX\\_TIMEOUT](#) 100U  
*time out for Synchronous command, millisecond.*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_JTAG\\_TIMEOUT](#) 1000U  
*time out for jtag channel of mailbox, millisecond.*

- #define `CONFIG_EHSM_ARCH_HOST_MAILBOX_POLLING`  
*configuration host driver using polling mechanism to read mailbox data, if not defined using interrupt to read mailbox data*
- #define `CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_KEY\_MATERIAL*
- #define `CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_EXT\_SHE\_KEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_MAC\_PROOF*
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_PROOF*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PARTIAL_ACCESS` (`true`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_IV*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_K_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_MAC\_KEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_K_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_K_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_MAC_K_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_KEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_CURVE\_ID*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_CIPHER\_ALG*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_KDF\_ALG*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_PARTIAL_ACCESS` (`false`)  
*configuration for key element CRYPTO\_KE\_CIPHER\_MAC\_ALG*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)

- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_CIPHER\_MAC\_ALG*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_CIPHER\_MAC\_ALG*
- #define `CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_CIPHER\_2NDKEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_PARTIAL_ACCESS` (`true`)
- configuration for key element CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_BASE*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_OWNPUKEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUKEY_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_PERSIST` (`false`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_PARTIAL_ACCESS` (`false`)
- configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_PEERPUKEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUKEY_PERSIST` (`false`)

- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINFO*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE*
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_SIGNATURE\_KEY*
- #define `CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_READ_ACCESS` (`CRYPTO_RA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_WRITE_ACCESS` (`CRYPTO_WA_ALLOWED`)
- #define `CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PERSIST` (false)
- #define `CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PARTIAL_ACCESS` (false)  
*configuration for key element CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED*

- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_KEY*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_PARTIAL\_ACCESS (true)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_PASSWD*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_PARTIAL\_ACCESS (true)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_SALT*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYDERIVATION\_TYPE*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYGENERATE\_KEY*



- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYGENERATE\_KEYINFO*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYGENERATE\_SEED*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYGENERATE\_ALGORITHM*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_IMPORT\_KEY*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_EXPORT\_KEY*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_EXPORT\_KEY\_BLOB*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_KEY\_STATUS*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_REMOVE\_KEY*



- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_PARTIAL\_ACCESS (false)
- configuration for key element CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PARTIAL\_ACCESS (true)
- configuration for key element CRYPTO\_KE\_CERTIFICATE\_DATA*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_PARTIAL\_ACCESS (true)
- configuration for key element CRYPTO\_KE\_CERTIFICATE\_SUBJECT\_PUBLIC\_KEY*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PARTIAL\_ACCESS (true)
- configuration for key element CRYPTO\_KE\_CERTIFICATE\_SIGNATURE*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PERSIST (false)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_PARTIAL\_ACCESS (true)
- configuration for key element CRYPTO\_KE\_CERTIFICATE\_SIGNEDDATA*
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_READ\_ACCESS (CRYPTO\_RA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_WRITE\_ACCESS (CRYPTO\_WA\_ALLOWED)
- #define CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_PERSIST (false)

## 4.8.1 Macro Definition Documentation

### 4.8.1.1 CONFIG\_EHSM\_ARCH\_HOST\_MAILBOX\_POLLING

```
#define CONFIG_EHSM_ARCH_HOST_MAILBOX_POLLING
```

configuration host driver using polling mechanism to read mailbox data, if not defined using interrupt to read mailbox data

Definition at line 83 of file eHSM\_Config\_Ip.h.

#### 4.8.1.2 CONFIG\_EHSM\_ARCH\_V\_CMD\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CMD_QUEUE_SIZE 2U
```

command queue size in eHSM for each crypto object

Definition at line 57 of file eHSM\_Config\_Ip.h.

#### 4.8.1.3 CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_HASH\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CRYPT_OBJ_HASH_QUEUE_SIZE 10U
```

Definition at line 49 of file eHSM\_Config\_Ip.h.

#### 4.8.1.4 CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_K\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CRYPT_OBJ_K_QUEUE_SIZE 10U
```

Definition at line 50 of file eHSM\_Config\_Ip.h.

#### 4.8.1.5 CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_PKE\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CRYPT_OBJ_PKE_QUEUE_SIZE 10U
```

configuration for SE support

configuration for crypto object queue size

Definition at line 47 of file eHSM\_Config\_Ip.h.

#### 4.8.1.6 CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_SKE\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CRYPT_OBJ_SKE_QUEUE_SIZE 10U
```

Definition at line 52 of file eHSM\_Config\_Ip.h.

#### 4.8.1.7 CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_SYSMGR\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CRYPT_OBJ_SYSMGR_QUEUE_SIZE 10U
```

Definition at line 51 of file eHSM\_Config\_Ip.h.

#### 4.8.1.8 CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_TRNG\_QUEUE\_SIZE

```
#define CONFIG_EHSM_ARCH_V_CRYPTOBJ_TRNG_QUEUE_SIZE 10U
```

Definition at line 48 of file eHSM\_Config\_Ip.h.

#### 4.8.1.9 CONFIG\_EHSM\_ARCH\_V\_DEFAULT\_CMD\_TIMEOUT

```
#define CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT 0x1FFCFFFU
```

time out for Synchronous command, millisecond.

Definition at line 62 of file eHSM\_Config\_Ip.h.

#### 4.8.1.10 CONFIG\_EHSM\_ARCH\_V\_JTAG\_TIMEOUT

```
#define CONFIG_EHSM_ARCH_V_JTAG_TIMEOUT 1000U
```

time out for jtag channel of mailbox, millisecond.

Definition at line 77 of file eHSM\_Config\_Ip.h.

#### 4.8.1.11 CONFIG\_EHSM\_ARCH\_V\_MAILBOX\_TIMEOUT

```
#define CONFIG_EHSM_ARCH_V_MAILBOX_TIMEOUT 100U
```

time out for Synchronous command, millisecond.

Definition at line 72 of file eHSM\_Config\_Ip.h.

#### 4.8.1.12 CONFIG\_EHSM\_ARCH\_V\_RSA\_K\_CMD\_TIMEOUT

```
#define CONFIG_EHSM_ARCH_V_RSA_K_CMD_TIMEOUT 0x1FFFFFFU
```

time out for RSA key generation.

Definition at line 67 of file eHSM\_Config\_Ip.h.

#### 4.8.1.13 CONFIG\_EHSM\_AUTOSAR

```
#define CONFIG_EHSM_AUTOSAR
```

configuration for AUTOSAR support

Definition at line 22 of file eHSM\_Config\_Ip.h.

#### 4.8.1.14 CONFIG\_EHSM\_EVITA

```
#define CONFIG_EHSM_EVITA
```

configuration for EVITA support

Definition at line 27 of file eHSM\_Config\_Ip.h.

#### 4.8.1.15 CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_PARTIAL\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_MAC\_ALG

Definition at line 189 of file eHSM\_Config\_Ip.h.

#### 4.8.1.16 CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_PERSIST

```
#define CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_PERSIST (false)
```

Definition at line 192 of file eHSM\_Config\_Ip.h.

#### 4.8.1.17 CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_READ\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 190 of file eHSM\_Config\_Ip.h.

#### 4.8.1.18 CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_WRITE\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_AEAD_TAG_SIZE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 191 of file eHSM\_Config\_Ip.h.

#### 4.8.1.19 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PARTIAL\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_CERTIFICATE\_DATA

Definition at line 493 of file eHSM\_Config\_Ip.h.

#### 4.8.1.20 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PERSIST

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_PERSIST (false)
```

Definition at line 496 of file eHSM\_Config\_Ip.h.

#### 4.8.1.21 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_READ\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 494 of file eHSM\_Config\_Ip.h.

#### 4.8.1.22 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_WRITE\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_DATA_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 495 of file eHSM\_Config\_Ip.h.

#### 4.8.1.23 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PARTIAL\_ACCESS

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_CERTIFICATE\_SIGNATURE

Definition at line 509 of file eHSM\_Config\_Ip.h.

#### 4.8.1.24 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PERSIST

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_PERSIST (false)
```

Definition at line 512 of file eHSM\_Config\_Ip.h.

**4.8.1.25 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 510 of file eHSM\_Config\_Ip.h.

**4.8.1.26 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNATURE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 511 of file eHSM\_Config\_Ip.h.

**4.8.1.27 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_CERTIFICATE\_SIGNEDDATA

Definition at line 517 of file eHSM\_Config\_Ip.h.

**4.8.1.28 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_PERSIST (false)
```

Definition at line 520 of file eHSM\_Config\_Ip.h.

**4.8.1.29 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 518 of file eHSM\_Config\_Ip.h.

**4.8.1.30 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNEDDATA\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGNEDDATA_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 519 of file eHSM\_Config\_Ip.h.

**4.8.1.31 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_CERTIFICATE\_SUBJECT\_PUBLIC\_KEY

Definition at line 501 of file eHSM\_Config\_Ip.h.

**4.8.1.32 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_PERSIST (false)
```

Definition at line 504 of file eHSM\_Config\_Ip.h.

**4.8.1.33 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 502 of file eHSM\_Config\_Ip.h.

**4.8.1.34 CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SUBJECT\_PUBLIC\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJECT_PUBLIC_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 503 of file eHSM\_Config\_Ip.h.

**4.8.1.35 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKEY\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_2NDKEY

Definition at line 197 of file eHSM\_Config\_Ip.h.

**4.8.1.36 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKEY\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_PERSIST (false)
```

Definition at line 200 of file eHSM\_Config\_Ip.h.

**4.8.1.37 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKEY\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 198 of file eHSM\_Config\_Ip.h.

**4.8.1.38 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKEY\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 199 of file eHSM\_Config\_Ip.h.

**4.8.1.39 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHER\_ALG\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_CIPHER\_ALG

Definition at line 157 of file eHSM\_Config\_Ip.h.

**4.8.1.40 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHER\_ALG\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_PERSIST (false)
```

Definition at line 160 of file eHSM\_Config\_Ip.h.

**4.8.1.41 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHER\_ALG\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 158 of file eHSM\_Config\_Ip.h.

**4.8.1.42 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHER\_ALG\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_ALG_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 159 of file eHSM\_Config\_Ip.h.



**4.8.1.43 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURVE\_ID\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_CURVE\_ID

Definition at line 149 of file eHSM\_Config\_Ip.h.

**4.8.1.44 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURVE\_ID\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_PERSIST (false)
```

Definition at line 152 of file eHSM\_Config\_Ip.h.

**4.8.1.45 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURVE\_ID\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 150 of file eHSM\_Config\_Ip.h.

**4.8.1.46 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURVE\_ID\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 151 of file eHSM\_Config\_Ip.h.

**4.8.1.47 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_CIPHER\_IV

Definition at line 125 of file eHSM\_Config\_Ip.h.

**4.8.1.48 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PERSIST (false)
```

Definition at line 128 of file eHSM\_Config\_Ip.h.

**4.8.1.49 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 126 of file eHSM\_Config\_Ip.h.

**4.8.1.50 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 127 of file eHSM\_Config\_Ip.h.

**4.8.1.51 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_KEY

Definition at line 141 of file eHSM\_Config\_Ip.h.

**4.8.1.52 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_PERSIST (false)
```

Definition at line 144 of file eHSM\_Config\_Ip.h.

**4.8.1.53 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 142 of file eHSM\_Config\_Ip.h.

**4.8.1.54 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 143 of file eHSM\_Config\_Ip.h.

**4.8.1.55 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_KDF\_ALG

Definition at line 165 of file eHSM\_Config\_Ip.h.

**4.8.1.56 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_PERSIST (false)
```

Definition at line 168 of file eHSM\_Config\_Ip.h.

**4.8.1.57 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 166 of file eHSM\_Config\_Ip.h.

**4.8.1.58 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_KDF_ALG_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 167 of file eHSM\_Config\_Ip.h.

**4.8.1.59 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_MAC\_ALG

Definition at line 173 of file eHSM\_Config\_Ip.h.

**4.8.1.60 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_PERSIST (false)
```

Definition at line 176 of file eHSM\_Config\_Ip.h.

**4.8.1.61 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 174 of file eHSM\_Config\_Ip.h.

**4.8.1.62 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_ALG_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 175 of file eHSM\_Config\_Ip.h.

**4.8.1.63 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_MAC\_ALG

Definition at line 181 of file eHSM\_Config\_Ip.h.

**4.8.1.64 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_PERSIST (false)
```

Definition at line 184 of file eHSM\_Config\_Ip.h.

**4.8.1.65 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 182 of file eHSM\_Config\_Ip.h.

**4.8.1.66 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_MAC_SIZE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 183 of file eHSM\_Config\_Ip.h.

**4.8.1.67 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_CIPHER\_PROOF

Definition at line 117 of file eHSM\_Config\_Ip.h.

**4.8.1.68 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_PERSIST (false)
```

Definition at line 120 of file eHSM\_Config\_Ip.h.

**4.8.1.69 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 118 of file eHSM\_Config\_Ip.h.

**4.8.1.70 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_CIPHER_PROOF_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 119 of file eHSM\_Config\_Ip.h.

**4.8.1.71 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY

Definition at line 477 of file eHSM\_Config\_Ip.h.

**4.8.1.72 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_PERSIST (false)
```

Definition at line 480 of file eHSM\_Config\_Ip.h.

**4.8.1.73 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 478 of file eHSM\_Config\_Ip.h.

**4.8.1.74 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 479 of file eHSM\_Config\_Ip.h.

**4.8.1.75 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE

Definition at line 485 of file eHSM\_Config\_Ip.h.

**4.8.1.76 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_PERSIST (false)
```

Definition at line 488 of file eHSM\_Config\_Ip.h.

**4.8.1.77 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 486 of file eHSM\_Config\_Ip.h.

**4.8.1.78 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 487 of file eHSM\_Config\_Ip.h.

**4.8.1.79 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_EXPORT\_KEY\_BLOB

Definition at line 453 of file eHSM\_Config\_Ip.h.

**4.8.1.80 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_PERSIST (false)
```

Definition at line 456 of file eHSM\_Config\_Ip.h.

**4.8.1.81 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 454 of file eHSM\_Config\_Ip.h.

**4.8.1.82 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 455 of file eHSM\_Config\_Ip.h.

**4.8.1.83 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_EXPORT\_KEY

Definition at line 445 of file eHSM\_Config\_Ip.h.

**4.8.1.84 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_PERSIST (false)
```

Definition at line 448 of file eHSM\_Config\_Ip.h.

**4.8.1.85 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 446 of file eHSM\_Config\_Ip.h.

**4.8.1.86 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 447 of file eHSM\_Config\_Ip.h.

**4.8.1.87 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_IMPORT\_KEY

Definition at line 429 of file eHSM\_Config\_Ip.h.

**4.8.1.88 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_PERSIST (false)
```

Definition at line 432 of file eHSM\_Config\_Ip.h.

**4.8.1.89 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 430 of file eHSM\_Config\_Ip.h.

**4.8.1.90 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 431 of file eHSM\_Config\_Ip.h.



**4.8.1.91 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE

Definition at line 437 of file eHSM\_Config\_Ip.h.

**4.8.1.92 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_PERSIST (false)
```

Definition at line 440 of file eHSM\_Config\_Ip.h.

**4.8.1.93 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 438 of file eHSM\_Config\_Ip.h.

**4.8.1.94 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 439 of file eHSM\_Config\_Ip.h.

**4.8.1.95 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_EXT\_SHE\_KEY

Definition at line 101 of file eHSM\_Config\_Ip.h.

**4.8.1.96 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_PERSIST (false)
```

Definition at line 104 of file eHSM\_Config\_Ip.h.

**4.8.1.97 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 102 of file eHSM\_Config\_Ip.h.

**4.8.1.98 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_EXT_SHE_KEY_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 103 of file eHSM\_Config\_Ip.h.

**4.8.1.99 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEY\_MATERIAL

Definition at line 93 of file eHSM\_Config\_Ip.h.

**4.8.1.100 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PERSIST (false)
```

Definition at line 96 of file eHSM\_Config\_Ip.h.

**4.8.1.101 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 94 of file eHSM\_Config\_Ip.h.

**4.8.1.102 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 95 of file eHSM\_Config\_Ip.h.

**4.8.1.103 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_STATUS_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEY\_STATUS

Definition at line 461 of file eHSM\_Config\_Ip.h.

**4.8.1.104 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_STATUS_PERSIST (false)
```

Definition at line 464 of file eHSM\_Config\_Ip.h.

**4.8.1.105 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_STATUS_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 462 of file eHSM\_Config\_Ip.h.

**4.8.1.106 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_K_STATUS_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 463 of file eHSM\_Config\_Ip.h.

**4.8.1.107 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM

Definition at line 365 of file eHSM\_Config\_Ip.h.

**4.8.1.108 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_PERSIST (false)
```

Definition at line 368 of file eHSM\_Config\_Ip.h.

**4.8.1.109 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 366 of file eHSM\_Config\_Ip.h.

**4.8.1.110 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ALGORITHM_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 367 of file eHSM\_Config\_Ip.h.

**4.8.1.111 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS

Definition at line 357 of file eHSM\_Config\_Ip.h.

**4.8.1.112 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_PERSIST (false)
```

Definition at line 360 of file eHSM\_Config\_Ip.h.

**4.8.1.113 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 358 of file eHSM\_Config\_Ip.h.

**4.8.1.114 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_ITERATIONS_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 359 of file eHSM\_Config\_Ip.h.

**4.8.1.115 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_KEY

Definition at line 333 of file eHSM\_Config\_Ip.h.

**4.8.1.116 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_PERSIST (false)
```

Definition at line 336 of file eHSM\_Config\_Ip.h.

**4.8.1.117 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 334 of file eHSM\_Config\_Ip.h.

**4.8.1.118 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 335 of file eHSM\_Config\_Ip.h.

**4.8.1.119 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE

Definition at line 373 of file eHSM\_Config\_Ip.h.

**4.8.1.120 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_PERSIST (false)
```

Definition at line 376 of file eHSM\_Config\_Ip.h.

**4.8.1.121 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 374 of file eHSM\_Config\_Ip.h.

**4.8.1.122 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_KHANDLE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 375 of file eHSM\_Config\_Ip.h.

**4.8.1.123 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_PASSWD

Definition at line 341 of file eHSM\_Config\_Ip.h.

**4.8.1.124 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_PERSIST (false)
```

Definition at line 344 of file eHSM\_Config\_Ip.h.

**4.8.1.125 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 342 of file eHSM\_Config\_Ip.h.

**4.8.1.126 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_PASSWD_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 343 of file eHSM\_Config\_Ip.h.

**4.8.1.127 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_PARTIAL_ACCESS (true)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_SALT

Definition at line 349 of file eHSM\_Config\_Ip.h.

**4.8.1.128 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_PERSIST (false)
```

Definition at line 352 of file eHSM\_Config\_Ip.h.

**4.8.1.129 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 350 of file eHSM\_Config\_Ip.h.

**4.8.1.130 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_SALT_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 351 of file eHSM\_Config\_Ip.h.

**4.8.1.131 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYDERIVATION\_TYPE

Definition at line 381 of file eHSM\_Config\_Ip.h.

**4.8.1.132 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_PERSIST (false)
```

Definition at line 384 of file eHSM\_Config\_Ip.h.

**4.8.1.133 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 382 of file eHSM\_Config\_Ip.h.

**4.8.1.134 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KDERIVATION_TYPE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 383 of file eHSM\_Config\_Ip.h.

**4.8.1.135 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM

Definition at line 237 of file eHSM\_Config\_Ip.h.

**4.8.1.136 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_PERSIST (false)
```

Definition at line 240 of file eHSM\_Config\_Ip.h.

**4.8.1.137 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 238 of file eHSM\_Config\_Ip.h.

**4.8.1.138 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_ALGORITHM_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 239 of file eHSM\_Config\_Ip.h.



**4.8.1.139 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_BASE

Definition at line 213 of file eHSM\_Config\_Ip.h.

**4.8.1.140 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_PERSIST (false)
```

Definition at line 216 of file eHSM\_Config\_Ip.h.

**4.8.1.141 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 214 of file eHSM\_Config\_Ip.h.

**4.8.1.142 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_BASE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 215 of file eHSM\_Config\_Ip.h.

**4.8.1.143 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYINFO\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO

Definition at line 253 of file eHSM\_Config\_Ip.h.

**4.8.1.144 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYINFO\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_PERSIST (false)
```

Definition at line 256 of file eHSM\_Config\_Ip.h.

**4.8.1.145 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYINFO\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 254 of file eHSM\_Config\_Ip.h.

**4.8.1.146 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYINFO\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_KEYINFO_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 255 of file eHSM\_Config\_Ip.h.

**4.8.1.147 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNPUBKEY\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUBKEY_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_OWNPUBKEY

Definition at line 229 of file eHSM\_Config\_Ip.h.

**4.8.1.148 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNPUBKEY\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUBKEY_PERSIST (false)
```

Definition at line 232 of file eHSM\_Config\_Ip.h.

**4.8.1.149 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNPUBKEY\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUBKEY_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 230 of file eHSM\_Config\_Ip.h.

**4.8.1.150 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNPUBKEY\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_OWNPUBKEY_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 231 of file eHSM\_Config\_Ip.h.

**4.8.1.151 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEERPUBKEY\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUBKEY_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_PEERPUBKEY

Definition at line 245 of file eHSM\_Config\_Ip.h.

**4.8.1.152 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEERPUBKEY\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUBKEY_PERSIST (false)
```

Definition at line 248 of file eHSM\_Config\_Ip.h.

**4.8.1.153 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEERPUBKEY\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUBKEY_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 246 of file eHSM\_Config\_Ip.h.

**4.8.1.154 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEERPUBKEY\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PEERPUBKEY_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 247 of file eHSM\_Config\_Ip.h.

**4.8.1.155 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVKEY\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY

Definition at line 221 of file eHSM\_Config\_Ip.h.

**4.8.1.156 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVKEY\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_PERSIST (false)
```

Definition at line 224 of file eHSM\_Config\_Ip.h.

**4.8.1.157 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVKEY\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 222 of file eHSM\_Config\_Ip.h.

**4.8.1.158 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVKEY\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PRIVKEY_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 223 of file eHSM\_Config\_Ip.h.

**4.8.1.159 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBKTYPE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE

Definition at line 261 of file eHSM\_Config\_Ip.h.

**4.8.1.160 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBKTYPE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_PERSIST (false)
```

Definition at line 264 of file eHSM\_Config\_Ip.h.

**4.8.1.161 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBKTYPE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 262 of file eHSM\_Config\_Ip.h.

**4.8.1.162 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBKTYPE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_PUBKTYPE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 263 of file eHSM\_Config\_Ip.h.

**4.8.1.163 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAREDVALUE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_PARTIAL_ACCESS (true)
```

configuration for key element CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE

Definition at line 205 of file eHSM\_Config\_Ip.h.

**4.8.1.164 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAREDVALUE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_PERSIST (false)
```

Definition at line 208 of file eHSM\_Config\_Ip.h.

**4.8.1.165 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAREDVALUE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 206 of file eHSM\_Config\_Ip.h.

**4.8.1.166 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAREDVALUE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SHAREDVALUE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 207 of file eHSM\_Config\_Ip.h.

**4.8.1.167 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_LOCALTMPKINFO\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINFO

Definition at line 269 of file eHSM\_Config\_Ip.h.

**4.8.1.168 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_LOCALTMPKINFO\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_PERSIST (false)
```

Definition at line 272 of file eHSM\_Config\_Ip.h.

**4.8.1.169 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_LOCALTMPKINFO\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 270 of file eHSM\_Config\_Ip.h.

**4.8.1.170 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_LOCALTMPKINFO\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_LOCALTMPKINFO_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 271 of file eHSM\_Config\_Ip.h.

**4.8.1.171 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_PEERTMPPUBK\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK

Definition at line 277 of file eHSM\_Config\_Ip.h.

**4.8.1.172 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_PEERTMPPUBK\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_PERSIST (false)
```

Definition at line 280 of file eHSM\_Config\_Ip.h.

**4.8.1.173 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_PEERTMPPUBK\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 278 of file eHSM\_Config\_Ip.h.

**4.8.1.174 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_PEERTMPPUBK\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_PEERTMPPUBK_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 279 of file eHSM\_Config\_Ip.h.

**4.8.1.175 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_ROLE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE

Definition at line 301 of file eHSM\_Config\_Ip.h.

**4.8.1.176 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_ROLE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_PERSIST (false)
```

Definition at line 304 of file eHSM\_Config\_Ip.h.

**4.8.1.177 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_ROLE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 302 of file eHSM\_Config\_Ip.h.

**4.8.1.178 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_ROLE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_ROLE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 303 of file eHSM\_Config\_Ip.h.

**4.8.1.179 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_S1\_S2\_VALUE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE

Definition at line 285 of file eHSM\_Config\_Ip.h.

**4.8.1.180 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_S1\_S2\_VALUE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_PERSIST (false)
```

Definition at line 288 of file eHSM\_Config\_Ip.h.

**4.8.1.181 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_S1\_S2\_VALUE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 286 of file eHSM\_Config\_Ip.h.

**4.8.1.182 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_S1\_S2\_VALUE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_S1_S2_VALUE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 287 of file eHSM\_Config\_Ip.h.

**4.8.1.183 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_SA\_SB\_VALUE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE

Definition at line 293 of file eHSM\_Config\_Ip.h.

**4.8.1.184 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_SA\_SB\_VALUE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_PERSIST (false)
```

Definition at line 296 of file eHSM\_Config\_Ip.h.

**4.8.1.185 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_SA\_SB\_VALUE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 294 of file eHSM\_Config\_Ip.h.

**4.8.1.186 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_SA\_SB\_VALUE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KEXCHANGE_SM2_SA_SB_VALUE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 295 of file eHSM\_Config\_Ip.h.



**4.8.1.187 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYGENERATE\_ALGORITHM

Definition at line 413 of file eHSM\_Config\_Ip.h.

**4.8.1.188 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_PERSIST (false)
```

Definition at line 416 of file eHSM\_Config\_Ip.h.

**4.8.1.189 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 414 of file eHSM\_Config\_Ip.h.

**4.8.1.190 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGORITHM\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_ALGORITHM_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 415 of file eHSM\_Config\_Ip.h.

**4.8.1.191 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO

Definition at line 421 of file eHSM\_Config\_Ip.h.

**4.8.1.192 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_PERSIST (false)
```

Definition at line 424 of file eHSM\_Config\_Ip.h.

**4.8.1.193 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 422 of file eHSM\_Config\_Ip.h.

**4.8.1.194 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_INFO\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_DH_K_INFO_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 423 of file eHSM\_Config\_Ip.h.

**4.8.1.195 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYGENERATE\_KEY

Definition at line 389 of file eHSM\_Config\_Ip.h.

**4.8.1.196 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_PERSIST (false)
```

Definition at line 392 of file eHSM\_Config\_Ip.h.

**4.8.1.197 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 390 of file eHSM\_Config\_Ip.h.

**4.8.1.198 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 391 of file eHSM\_Config\_Ip.h.

**4.8.1.199 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYGENERATE\_KEYINFO

Definition at line 397 of file eHSM\_Config\_Ip.h.

**4.8.1.200 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_PERSIST (false)
```

Definition at line 400 of file eHSM\_Config\_Ip.h.

**4.8.1.201 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 398 of file eHSM\_Config\_Ip.h.

**4.8.1.202 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_KINFO_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 399 of file eHSM\_Config\_Ip.h.

**4.8.1.203 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_KEYGENERATE\_SEED

Definition at line 405 of file eHSM\_Config\_Ip.h.

**4.8.1.204 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_PERSIST (false)
```

Definition at line 408 of file eHSM\_Config\_Ip.h.

**4.8.1.205 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 406 of file eHSM\_Config\_Ip.h.

**4.8.1.206 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 407 of file eHSM\_Config\_Ip.h.

**4.8.1.207 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_MAC\_KEY

Definition at line 133 of file eHSM\_Config\_Ip.h.

**4.8.1.208 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_K_PERSIST (false)
```

Definition at line 136 of file eHSM\_Config\_Ip.h.

**4.8.1.209 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 134 of file eHSM\_Config\_Ip.h.

**4.8.1.210 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 135 of file eHSM\_Config\_Ip.h.

**4.8.1.211 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_PARTIAL\_ACCESSRTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_PARTIAL_ACCESSRTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_MAC\_PROOF

Definition at line 109 of file eHSM\_Config\_Ip.h.

**4.8.1.212 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_PERSIST (false)
```

Definition at line 112 of file eHSM\_Config\_Ip.h.

**4.8.1.213 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 110 of file eHSM\_Config\_Ip.h.

**4.8.1.214 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_MAC_PROOF_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 111 of file eHSM\_Config\_Ip.h.

**4.8.1.215 CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_REMOVE\_KEY

Definition at line 469 of file eHSM\_Config\_Ip.h.

**4.8.1.216 CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_PERSIST (false)
```

Definition at line 472 of file eHSM\_Config\_Ip.h.

**4.8.1.217 CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 470 of file eHSM\_Config\_Ip.h.

**4.8.1.218 CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 471 of file eHSM\_Config\_Ip.h.

**4.8.1.219 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_SIGNATURE\_KEY

Definition at line 309 of file eHSM\_Config\_Ip.h.

**4.8.1.220 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PERSIST (false)
```

Definition at line 312 of file eHSM\_Config\_Ip.h.

**4.8.1.221 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 310 of file eHSM\_Config\_Ip.h.

**4.8.1.222 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 311 of file eHSM\_Config\_Ip.h.

**4.8.1.223 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE

Definition at line 325 of file eHSM\_Config\_Ip.h.

**4.8.1.224 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_PERSIST (false)
```

Definition at line 328 of file eHSM\_Config\_Ip.h.

**4.8.1.225 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 326 of file eHSM\_Config\_Ip.h.

**4.8.1.226 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_CRT\_MODE\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 327 of file eHSM\_Config\_Ip.h.

**4.8.1.227 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_PARTIAL\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PARTIAL_ACCESS (false)
```

configuration for key element CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED

Definition at line 317 of file eHSM\_Config\_Ip.h.

**4.8.1.228 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_PERSIST**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PERSIST (false)
```

Definition at line 320 of file eHSM\_Config\_Ip.h.

**4.8.1.229 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_READ\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_READ_ACCESS (CRYPTO_RA_ALLOWED)
```

Definition at line 318 of file eHSM\_Config\_Ip.h.

**4.8.1.230 CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMESTAMPED\_WRITE\_ACCESS**

```
#define CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_WRITE_ACCESS (CRYPTO_WA_ALLOWED)
```

Definition at line 319 of file eHSM\_Config\_Ip.h.

**4.8.1.231 CONFIG\_EHSM\_SHE**

```
#define CONFIG_EHSM_SHE
```

configuration for SHE support

Definition at line 32 of file eHSM\_Config\_Ip.h.

**4.9 eHSM\_Debug\_Ip.h File Reference**

```
#include "eHSM_IntCfg_Ip.h"
#include "stdio.h"
#include "string.h"
```

**Macros**

- #define [filename](#)(p) strchr(p, '\\')? strchr(p, '\\')+1 : p
- #define [CONFIG\\_HOST\\_V\\_LOG\\_ERR](#) 0
- #define [CONFIG\\_HOST\\_V\\_LOG\\_WARN](#) 1
- #define [CONFIG\\_HOST\\_V\\_LOG\\_DEBUG](#) 2
- #define [CONFIG\\_HOST\\_V\\_LOG\\_INFO](#) 3
- #define [CONFIG\\_HOST\\_V\\_LOG\\_LEVEL](#) [CONFIG\\_HOST\\_V\\_LOG\\_INFO](#)
- #define [HOST\\_LOG](#)(level, fmt, ...)
- #define [PURE\\_LOG](#)(level, fmt, ...)
- #define [HOST\\_LOG\\_ERROR](#)(...) [HOST\\_LOG](#)([CONFIG\\_HOST\\_V\\_LOG\\_ERR](#), ##\_\_VA\_ARGS\_\_)
- #define [HOST\\_LOG\\_WARN](#)(...) [HOST\\_LOG](#)([CONFIG\\_HOST\\_V\\_LOG\\_WARN](#), ##\_\_VA\_ARGS\_\_)
- #define [HOST\\_LOG\\_DEBUG](#)(...) [HOST\\_LOG](#)([CONFIG\\_HOST\\_V\\_LOG\\_DEBUG](#), ##\_\_VA\_ARGS\_\_)
- #define [HOST\\_LOG\\_INFO](#)(...) [HOST\\_LOG](#)([CONFIG\\_HOST\\_V\\_LOG\\_INFO](#), ##\_\_VA\_ARGS\_\_)
- #define [HOST\\_PURE\\_LOG](#)(...) [PURE\\_LOG](#)([CONFIG\\_HOST\\_V\\_LOG\\_INFO](#), ##\_\_VA\_ARGS\_\_)
- #define [CONFIG\\_HOST\\_COMMON\\_DEBUG\\_ENABLE](#)
- #define [COMMON\\_LOG\\_ERROR](#)(fmt, args...) [HOST\\_LOG\\_ERROR](#)(fmt, ##args)
- #define [COMMON\\_LOG\\_WARN](#)(fmt, args...) [HOST\\_LOG\\_WARN](#)(fmt, ##args)
- #define [COMMON\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_LOG\\_DEBUG](#)(fmt, ##args)
- #define [COMMON\\_LOG\\_INFO](#)(fmt, args...) [HOST\\_LOG\\_INFO](#)(fmt, ##args)
- #define [COMMON\\_PURE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_PURE\\_LOG](#)(fmt, ##args)



- #define [CONFIG\\_HOST\\_EVITA\\_DEBUG\\_ENABLE](#)
- #define [EVITA\\_LOG\\_ERROR](#)(fmt, args...) [HOST\\_LOG\\_ERROR](#)(fmt, ##args)
- #define [EVITA\\_LOG\\_WARN](#)(fmt, args...) [HOST\\_LOG\\_WARN](#)(fmt, ##args)
- #define [EVITA\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_LOG\\_DEBUG](#)(fmt, ##args)
- #define [EVITA\\_LOG\\_INFO](#)(fmt, args...) [HOST\\_LOG\\_INFO](#)(fmt, ##args)
- #define [EVITA\\_PURE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_PURE\\_LOG](#)(fmt, ##args)
- #define [CONFIG\\_HOST\\_AUTOSAR\\_DEBUG\\_ENABLE](#)
- #define [AUTOSAR\\_LOG\\_ERROR](#)(fmt, args...) [HOST\\_LOG\\_ERROR](#)(fmt, ##args)
- #define [AUTOSAR\\_LOG\\_WARN](#)(fmt, args...) [HOST\\_LOG\\_WARN](#)(fmt, ##args)
- #define [AUTOSAR\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_LOG\\_DEBUG](#)(fmt, ##args)
- #define [AUTOSAR\\_LOG\\_INFO](#)(fmt, args...) [HOST\\_LOG\\_INFO](#)(fmt, ##args)
- #define [AUTOSAR\\_PURE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_PURE\\_LOG](#)(fmt, ##args)
- #define [CONFIG\\_HOST\\_SHE\\_DEBUG\\_ENABLE](#)
- #define [SHE\\_LOG\\_ERROR](#)(fmt, args...) [HOST\\_LOG\\_ERROR](#)(fmt, ##args)
- #define [SHE\\_LOG\\_WARN](#)(fmt, args...) [HOST\\_LOG\\_WARN](#)(fmt, ##args)
- #define [SHE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_LOG\\_DEBUG](#)(fmt, ##args)
- #define [SHE\\_LOG\\_INFO](#)(fmt, args...) [HOST\\_LOG\\_INFO](#)(fmt, ##args)
- #define [SHE\\_PURE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_PURE\\_LOG](#)(fmt, ##args)
- #define [CONFIG\\_HOST\\_CUSTOM\\_DEBUG\\_ENABLE](#)
- #define [CUSTOM\\_LOG\\_ERROR](#)(fmt, args...) [HOST\\_LOG\\_ERROR](#)(fmt, ##args)
- #define [CUSTOM\\_LOG\\_WARN](#)(fmt, args...) [HOST\\_LOG\\_WARN](#)(fmt, ##args)
- #define [CUSTOM\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_LOG\\_DEBUG](#)(fmt, ##args)
- #define [CUSTOM\\_LOG\\_INFO](#)(fmt, args...) [HOST\\_LOG\\_INFO](#)(fmt, ##args)
- #define [CUSTOM\\_PURE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_PURE\\_LOG](#)(fmt, ##args)
- #define [CONFIG\\_HOST\\_PERFORMANCE\\_DEBUG\\_ENABLE](#)
- #define [PERFORMANCE\\_LOG\\_ERROR](#)(fmt, args...) [HOST\\_LOG\\_ERROR](#)(fmt, ##args)
- #define [PERFORMANCE\\_LOG\\_WARN](#)(fmt, args...) [HOST\\_LOG\\_WARN](#)(fmt, ##args)
- #define [PERFORMANCE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_LOG\\_DEBUG](#)(fmt, ##args)
- #define [PERFORMANCE\\_LOG\\_INFO](#)(fmt, args...) [HOST\\_LOG\\_INFO](#)(fmt, ##args)
- #define [PERFORMANCE\\_PURE\\_LOG\\_DEBUG](#)(fmt, args...) [HOST\\_PURE\\_LOG](#)(fmt, ##args)

## Functions

- void [Debug\\_Printf](#) (const char \*format,...)

## 4.9.1 Macro Definition Documentation

### 4.9.1.1 AUTOSAR\_LOG\_DEBUG

```
#define AUTOSAR_LOG_DEBUG (
 fmt,
 args...) HOST_LOG_DEBUG(fmt, ##args)
```

Definition at line 153 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.2 AUTOSAR\_LOG\_ERROR

```
#define AUTOSAR_LOG_ERROR(
 fmt,
 args...) HOST_LOG_ERROR(fmt, ##args)
```

Definition at line 151 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.3 AUTOSAR\_LOG\_INFO

```
#define AUTOSAR_LOG_INFO(
 fmt,
 args...) HOST_LOG_INFO(fmt, ##args)
```

Definition at line 154 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.4 AUTOSAR\_LOG\_WARN

```
#define AUTOSAR_LOG_WARN(
 fmt,
 args...) HOST_LOG_WARN(fmt, ##args)
```

Definition at line 152 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.5 AUTOSAR\_PURE\_LOG\_DEBUG

```
#define AUTOSAR_PURE_LOG_DEBUG(
 fmt,
 args...) HOST_PURE_LOG(fmt, ##args)
```

Definition at line 155 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.6 COMMON\_LOG\_DEBUG

```
#define COMMON_LOG_DEBUG(
 fmt,
 args...) HOST_LOG_DEBUG(fmt, ##args)
```

Definition at line 123 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.7 COMMON\_LOG\_ERROR

```
#define COMMON_LOG_ERROR(
 fmt,
 args...) HOST_LOG_ERROR(fmt, ##args)
```

Definition at line 121 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.8 COMMON\_LOG\_INFO

```
#define COMMON_LOG_INFO(
 fmt,
 args...) HOST_LOG_INFO(fmt, ##args)
```

Definition at line 124 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.9 COMMON\_LOG\_WARN

```
#define COMMON_LOG_WARN(
 fmt,
 args...) HOST_LOG_WARN(fmt, ##args)
```

Definition at line 122 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.10 COMMON\_PURE\_LOG\_DEBUG

```
#define COMMON_PURE_LOG_DEBUG(
 fmt,
 args...) HOST_PURE_LOG(fmt, ##args)
```

Definition at line 125 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.11 CONFIG\_HOST\_AUTOSAR\_DEBUG\_ENABLE

```
#define CONFIG_HOST_AUTOSAR_DEBUG_ENABLE
```

Definition at line 149 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.12 CONFIG\_HOST\_COMMON\_DEBUG\_ENABLE

```
#define CONFIG_HOST_COMMON_DEBUG_ENABLE
```

Definition at line 119 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.13 CONFIG\_HOST\_CUSTOM\_DEBUG\_ENABLE

```
#define CONFIG_HOST_CUSTOM_DEBUG_ENABLE
```

Definition at line 179 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.14 CONFIG\_HOST\_EVITA\_DEBUG\_ENABLE

```
#define CONFIG_HOST_EVITA_DEBUG_ENABLE
```

Definition at line 134 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.15 CONFIG\_HOST\_PERFORMANCE\_DEBUG\_ENABLE

```
#define CONFIG_HOST_PERFORMANCE_DEBUG_ENABLE
```

Definition at line 194 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.16 CONFIG\_HOST\_SHE\_DEBUG\_ENABLE

```
#define CONFIG_HOST_SHE_DEBUG_ENABLE
```

Definition at line 164 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.17 CONFIG\_HOST\_V\_LOG\_DEBUG

```
#define CONFIG_HOST_V_LOG_DEBUG 2
```

Definition at line 55 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.18 CONFIG\_HOST\_V\_LOG\_ERR

```
#define CONFIG_HOST_V_LOG_ERR 0
```

Definition at line 53 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.19 CONFIG\_HOST\_V\_LOG\_INFO

```
#define CONFIG_HOST_V_LOG_INFO 3
```

Definition at line 56 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.20 CONFIG\_HOST\_V\_LOG\_LEVEL

```
#define CONFIG_HOST_V_LOG_LEVEL CONFIG_HOST_V_LOG_INFO
```

Definition at line 57 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.21 CONFIG\_HOST\_V\_LOG\_WARN

```
#define CONFIG_HOST_V_LOG_WARN 1
```

Definition at line 54 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.22 CUSTOM\_LOG\_DEBUG

```
#define CUSTOM_LOG_DEBUG(
 fmt,
 args...) HOST_LOG_DEBUG(fmt, ##args)
```

Definition at line 183 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.23 CUSTOM\_LOG\_ERROR

```
#define CUSTOM_LOG_ERROR(
 fmt,
 args...) HOST_LOG_ERROR(fmt, ##args)
```

Definition at line 181 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.24 CUSTOM\_LOG\_INFO

```
#define CUSTOM_LOG_INFO(
 fmt,
 args...) HOST_LOG_INFO(fmt, ##args)
```

Definition at line 184 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.25 CUSTOM\_LOG\_WARN

```
#define CUSTOM_LOG_WARN(
 fmt,
 args...) HOST_LOG_WARN(fmt, ##args)
```

Definition at line 182 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.26 CUSTOM\_PURE\_LOG\_DEBUG

```
#define CUSTOM_PURE_LOG_DEBUG(
 fmt,
 args...) HOST_PURE_LOG(fmt, ##args)
```

Definition at line 185 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.27 EVITA\_LOG\_DEBUG

```
#define EVITA_LOG_DEBUG(
 fmt,
 args...) HOST_LOG_DEBUG(fmt, ##args)
```

Definition at line 138 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.28 EVITA\_LOG\_ERROR

```
#define EVITA_LOG_ERROR(
 fmt,
 args...) HOST_LOG_ERROR(fmt, ##args)
```

Definition at line 136 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.29 EVITA\_LOG\_INFO

```
#define EVITA_LOG_INFO(
 fmt,
 args...) HOST_LOG_INFO(fmt, ##args)
```

Definition at line 139 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.30 EVITA\_LOG\_WARN

```
#define EVITA_LOG_WARN(
 fmt,
 args...) HOST_LOG_WARN(fmt, ##args)
```

Definition at line 137 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.31 EVITA\_PURE\_LOG\_DEBUG

```
#define EVITA_PURE_LOG_DEBUG(
 fmt,
 args...) HOST_PURE_LOG(fmt, ##args)
```

Definition at line 140 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.32 filename

```
#define filename(
 p) strrchr(p, '\\')? strrchr(p, '\\')+1 : p
```

Definition at line 23 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.33 HOST\_LOG

```
#define HOST_LOG(
 level,
 fmt,
 ...)
```

Definition at line 82 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.34 HOST\_LOG\_DEBUG

```
#define HOST_LOG_DEBUG(
 ...) HOST_LOG(CONFIG_HOST_V_LOG_DEBUG, ##__VA_ARGS__)
```

Definition at line 100 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.35 HOST\_LOG\_ERROR

```
#define HOST_LOG_ERROR(
 ...) HOST_LOG(CONFIG_HOST_V_LOG_ERR, ##__VA_ARGS__)
```

Definition at line 88 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.36 HOST\_LOG\_INFO

```
#define HOST_LOG_INFO(
 ...) HOST_LOG(CONFIG_HOST_V_LOG_INFO, ##__VA_ARGS__)
```

Definition at line 106 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.37 HOST\_LOG\_WARN

```
#define HOST_LOG_WARN(
 ...) HOST_LOG(CONFIG_HOST_V_LOG_WARN, ##__VA_ARGS__)
```

Definition at line 94 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.38 HOST\_PURE\_LOG

```
#define HOST_PURE_LOG(
 ...) PURE_LOG(CONFIG_HOST_V_LOG_INFO, ##__VA_ARGS__)
```

Definition at line 112 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.39 PERFORMANCE\_LOG\_DEBUG

```
#define PERFORMANCE_LOG_DEBUG(
 fmt,
 args...) HOST_LOG_DEBUG(fmt, ##args)
```

Definition at line 198 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.40 PERFORMANCE\_LOG\_ERROR

```
#define PERFORMANCE_LOG_ERROR(
 fmt,
 args...) HOST_LOG_ERROR(fmt, ##args)
```

Definition at line 196 of file eHSM\_Debug\_Ip.h.



#### 4.9.1.41 PERFORMANCE\_LOG\_INFO

```
#define PERFORMANCE_LOG_INFO(
 fmt,
 args...) HOST_LOG_INFO(fmt, ##args)
```

Definition at line 199 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.42 PERFORMANCE\_LOG\_WARN

```
#define PERFORMANCE_LOG_WARN(
 fmt,
 args...) HOST_LOG_WARN(fmt, ##args)
```

Definition at line 197 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.43 PERFORMANCE\_PURE\_LOG\_DEBUG

```
#define PERFORMANCE_PURE_LOG_DEBUG(
 fmt,
 args...) HOST_PURE_LOG(fmt, ##args)
```

Definition at line 200 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.44 PURE\_LOG

```
#define PURE_LOG(
 level,
 fmt,
 ...)
```

Definition at line 83 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.45 SHE\_LOG\_DEBUG

```
#define SHE_LOG_DEBUG(
 fmt,
 args...) HOST_LOG_DEBUG(fmt, ##args)
```

Definition at line 168 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.46 SHE\_LOG\_ERROR

```
#define SHE_LOG_ERROR(
 fmt,
 args...) HOST_LOG_ERROR(fmt, ##args)
```

Definition at line 166 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.47 SHE\_LOG\_INFO

```
#define SHE_LOG_INFO(
 fmt,
 args...) HOST_LOG_INFO(fmt, ##args)
```

Definition at line 169 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.48 SHE\_LOG\_WARN

```
#define SHE_LOG_WARN(
 fmt,
 args...) HOST_LOG_WARN(fmt, ##args)
```

Definition at line 167 of file eHSM\_Debug\_Ip.h.

#### 4.9.1.49 SHE\_PURE\_LOG\_DEBUG

```
#define SHE_PURE_LOG_DEBUG(
 fmt,
 args...) HOST_PURE_LOG(fmt, ##args)
```

Definition at line 170 of file eHSM\_Debug\_Ip.h.

### 4.9.2 Function Documentation

#### 4.9.2.1 Debug\_Printf()

```
void Debug_Printf (
 const char * format,
 ...)
```

## 4.10 eHSM\_Dspt\_CryObj\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include "eHSM_Dspt_CryObj_Ip.h"
#include "eHSM_Dspt_lp.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
#include "eHSM_Exclusive_Area.h"
```

### Functions

- void [hw\\_interrupt\\_disable](#) ()
- void [hw\\_interrupt\\_enable](#) ()
- [ehsm\\_uint32\\_t ehsm\\_crypto\\_object\\_init](#) (void)
- [ehsm\\_uint32\\_t ehsm\\_add\\_cmd\\_to\\_priority\\_queue](#) (ehsm\_cmd\_req\_st \*cmd)
- [ehsm\\_uint32\\_t ehsm\\_del\\_cmd\\_from\\_priority\\_queue](#) (ehsm\_cmd\_req\_st \*cmd)
- [ehsm\\_uint32\\_t ehsm\\_add\\_cmd\\_to\\_sent\\_queue](#) (ehsm\_cmd\_req\_st \*cmd)
- [ehsm\\_uint32\\_t ehsm\\_fetch\\_cmd\\_from\\_crypto\\_object](#) (ehsm\_cmd\_req\_st \*\*cmd)
- [ehsm\\_uint32\\_t ehsm\\_crypto\\_object\\_get\\_cmd\\_done](#) (crypto\_object\_type\_e object\_type, ehsm\_cmd\_req\_st \*\*cmd)
- [ehsm\\_bool\\_t ehsm\\_crypto\\_object\\_is\\_free](#) (crypto\_object\_type\_e object\_type)
- [ehsm\\_uint32\\_t ehsm\\_del\\_cmd\\_from\\_sent\\_queue](#) (ehsm\_cmd\_req\_st \*cmd)
- [Crypto\\_JobType \\* ehsm\\_crypto\\_object\\_get\\_job](#) (crypto\_object\_type\_e object\_type, ehsm\_uint32\_t jobId)

### 4.10.1 Function Documentation

#### 4.10.1.1 ehsm\_add\_cmd\_to\_priority\_queue()

```
ehsm_uint32_t ehsm_add_cmd_to_priority_queue (
 ehsm_cmd_req_st * cmd)
```

add cmd to priority queue of crypto object

Definition at line 131 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.2 ehsm\_add\_cmd\_to\_sent\_queue()

```
ehsm_uint32_t ehsm_add_cmd_to_sent_queue (
 ehsm_cmd_req_st * cmd)
```

add cmd to sent queue of crypto object

Definition at line 220 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.3 ehsm\_crypto\_object\_get\_cmd\_done()

```
ehsm_uint32_t ehsm_crypto_object_get_cmd_done (
 crypto_object_type_e object_type,
 ehsm_cmd_req_st ** cmd)
```

get cmd from crypto object which has been done

Definition at line 298 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.4 ehsm\_crypto\_object\_get\_job()

```
Crypto_JobType* ehsm_crypto_object_get_job (
 crypto_object_type_e object_type,
 ehsm_uint32_t jobId)
```

Definition at line 377 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.5 ehsm\_crypto\_object\_init()

```
ehsm_uint32_t ehsm_crypto_object_init (
 void)
```

crypto object initialization

##### Returns

0 for success, negtive values for error.

Definition at line 120 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.6 ehsm\_crypto\_object\_is\_free()

```
ehsm_bool_t ehsm_crypto_object_is_free (
 crypto_object_type_e object_type)
```

check whether crypto object is free

Definition at line 337 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.7 ehsm\_del\_cmd\_from\_priority\_queue()

```
ehsm_uint32_t ehsm_del_cmd_from_priority_queue (
 ehsm_cmd_req_st * cmd)
```

delete cmd from priority queue of crypto object

Definition at line 185 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.8 ehsm\_del\_cmd\_from\_sent\_queue()

```
ehsm_uint32_t ehsm_del_cmd_from_sent_queue (
 ehsm_cmd_req_st * cmd)
```

Definition at line 355 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.9 ehsm\_fetch\_cmd\_from\_crypto\_object()

```
ehsm_uint32_t ehsm_fetch_cmd_from_crypto_object (
 ehsm_cmd_req_st ** cmd)
```

fetch a command from priority queue of crypto object

Definition at line 261 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.10.1.10 hw\_interrupt\_disable()

```
void hw_interrupt_disable ()
```

#### 4.10.1.11 hw\_interrupt\_enable()

```
void hw_interrupt_enable ()
```

## 4.11 eHSM\_Dspt\_CryObj\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "stdbool.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Compt_List.h"
```

## Classes

- struct [crypto\\_object](#)

## Typedefs

- typedef struct [crypto\\_object](#) [crypto\\_object\\_st](#)

## Enumerations

- enum [crypto\\_object\\_state\\_e](#) { [CRYPTO\\_OBJECT\\_STATE\\_FREE](#) = 0, [CRYPTO\\_OBJECT\\_STATE\\_BUSY](#) }

## Functions

- [ehsm\\_uint32\\_t](#) [ehsm\\_crypto\\_object\\_init](#) (void)
- [ehsm\\_uint32\\_t](#) [ehsm\\_crypto\\_object\\_submit\\_cmd](#) ([ehsm\\_cmd\\_req\\_st](#) \*cmd)
- [ehsm\\_uint32\\_t](#) [ehsm\\_fetch\\_cmd\\_from\\_crypto\\_object](#) ([ehsm\\_cmd\\_req\\_st](#) \*\*cmd)
- [ehsm\\_bool\\_t](#) [ehsm\\_crypto\\_object\\_is\\_free](#) ([crypto\\_object\\_type\\_e](#) object\_type)
- [ehsm\\_uint32\\_t](#) [ehsm\\_add\\_cmd\\_to\\_sent\\_queue](#) ([ehsm\\_cmd\\_req\\_st](#) \*cmd)
- [ehsm\\_uint32\\_t](#) [ehsm\\_add\\_cmd\\_to\\_priority\\_queue](#) ([ehsm\\_cmd\\_req\\_st](#) \*cmd)
- [ehsm\\_uint32\\_t](#) [ehsm\\_del\\_cmd\\_from\\_priority\\_queue](#) ([ehsm\\_cmd\\_req\\_st](#) \*cmd)
- [ehsm\\_uint32\\_t](#) [ehsm\\_crypto\\_object\\_get\\_cmd\\_done](#) ([crypto\\_object\\_type\\_e](#) object\_type, [ehsm\\_cmd\\_req\\_st](#) \*\*cmd)
- [ehsm\\_uint32\\_t](#) [ehsm\\_del\\_cmd\\_from\\_sent\\_queue](#) ([ehsm\\_cmd\\_req\\_st](#) \*cmd)
- [Crypto\\_JobType](#) \* [ehsm\\_crypto\\_object\\_get\\_job](#) ([crypto\\_object\\_type\\_e](#) object\_type, [ehsm\\_uint32\\_t](#) jobId)

### 4.11.1 Typedef Documentation

#### 4.11.1.1 [crypto\\_object\\_st](#)

```
typedef struct crypto_object crypto_object_st
```

### 4.11.2 Enumeration Type Documentation

#### 4.11.2.1 [crypto\\_object\\_state\\_e](#)

```
enum crypto_object_state_e
```

##### Enumerator

|                                          |  |
|------------------------------------------|--|
| <a href="#">CRYPTO_OBJECT_STATE_FREE</a> |  |
| <a href="#">CRYPTO_OBJECT_STATE_BUSY</a> |  |

Definition at line 24 of file eHSM\_Dspt\_CryObj\_Ip.h.

### 4.11.3 Function Documentation

#### 4.11.3.1 ehsm\_add\_cmd\_to\_priority\_queue()

```
ehsm_uint32_t ehsm_add_cmd_to_priority_queue (
 ehsm_cmd_req_st * cmd)
```

add cmd to priority queue of crypto object

Definition at line 131 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.11.3.2 ehsm\_add\_cmd\_to\_sent\_queue()

```
ehsm_uint32_t ehsm_add_cmd_to_sent_queue (
 ehsm_cmd_req_st * cmd)
```

add cmd to sent queue of crypto object

Definition at line 220 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.11.3.3 ehsm\_crypto\_object\_get\_cmd\_done()

```
ehsm_uint32_t ehsm_crypto_object_get_cmd_done (
 crypto_object_type_e object_type,
 ehsm_cmd_req_st ** cmd)
```

get cmd from crypto object which has been done

Definition at line 298 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.11.3.4 ehsm\_crypto\_object\_get\_job()

```
Crypto_JobType* ehsm_crypto_object_get_job (
 crypto_object_type_e object_type,
 ehsm_uint32_t jobId)
```

Definition at line 377 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.11.3.5 ehsm\_crypto\_object\_init()

```
ehsm_uint32_t ehsm_crypto_object_init (
 void)
```

crypto object initialization

##### Returns

0 for success, negtive values for error.

Definition at line 120 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.11.3.6 ehsm\_crypto\_object\_is\_free()

```
ehsm_bool_t ehsm_crypto_object_is_free (
 crypto_object_type_e object_type)
```

check whether crypto object is free

Definition at line 337 of file eHSM\_Dspt\_CryObj\_Ip.c.

#### 4.11.3.7 ehsm\_crypto\_object\_submit\_cmd()

```
ehsm_uint32_t ehsm_crypto_object_submit_cmd (
 ehsm_cmd_req_st * cmd)
```

submit a command to crypto object

##### Returns

0 for success, negtive values for error.

#### 4.11.3.8 ehsm\_del\_cmd\_from\_priority\_queue()

```
ehsm_uint32_t ehsm_del_cmd_from_priority_queue (
 ehsm_cmd_req_st * cmd)
```

delete cmd from priority queue of crypto object

Definition at line 185 of file eHSM\_Dspt\_CryObj\_Ip.c.



## 4.11.3.9 ehsm\_del\_cmd\_from\_sent\_queue()

```
ehsm_uint32_t ehsm_del_cmd_from_sent_queue (
 ehsm_cmd_req_st * cmd)
```

Definition at line 355 of file eHSM\_Dspt\_CryObj\_Ip.c.

## 4.11.3.10 ehsm\_fetch\_cmd\_from\_crypto\_object()

```
ehsm_uint32_t ehsm_fetch_cmd_from_crypto_object (
 ehsm_cmd_req_st ** cmd)
```

fetch a command from priority queue of crypto object

Definition at line 261 of file eHSM\_Dspt\_CryObj\_Ip.c.

## 4.12 eHSM\_Dspt\_Ip.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Dspt_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Debug_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Compt_List.h"
#include "eHSM_Mailbox_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
#include "eHSM_Dspt_CryObj_Ip.h"
#include "eHSM_Exclusive_Area.h"
```

## Functions

- void [hw\\_interrupt\\_disable](#) ()
- void [hw\\_interrupt\\_enable](#) ()
- [ehsm\\_uint32\\_t ptest\\_time\\_counting\\_get\\_state](#) (ehsm\_uint32\_t time\_id)
- void [ptest\\_time\\_counting\\_end](#) (ehsm\_uint32\_t time\_id)
- [ehsm\\_uint32\\_t ehsm\\_submit\\_cmd\\_req](#) (ehsm\_cmd\_req\_st \*cmd)
- [ehsm\\_uint32\\_t ehsm\\_remove\\_cmd\\_from\\_queue](#) (ehsm\_cmd\_req\_st \*cmd)
- [ehsm\\_uint32\\_t ehsm\\_dispatcher\\_init](#) (void)
- void [ehsm\\_mbox\\_polling](#) ()
- void [Crypto\\_MainFunction](#) (void)

## 4.12.1 Function Documentation

### 4.12.1.1 Crypto\_MainFunction()

```
void Crypto_MainFunction (
 void)
```

Definition at line 195 of file eHSM\_Dspt\_Ip.c.

### 4.12.1.2 ehsm\_dispatcher\_init()

```
ehsm_uint32_t ehsm_dispatcher_init (
 void)
```

init ehsm dispatcher

#### Returns

0 for success, negtive values for error.

Definition at line 181 of file eHSM\_Dspt\_Ip.c.

### 4.12.1.3 ehsm\_mbox\_polling()

```
void ehsm_mbox_polling ()
```

Definition at line 309 of file eHSM\_Mailbox\_Ip.c.

### 4.12.1.4 ehsm\_remove\_cmd\_from\_queue()

```
ehsm_uint32_t ehsm_remove_cmd_from_queue (
 ehsm_cmd_req_st * cmd)
```

Definition at line 146 of file eHSM\_Dspt\_Ip.c.

#### 4.12.1.5 ehsm\_submit\_cmd\_req()

```
ehsm_uint32_t ehsm_submit_cmd_req (
 ehsm_cmd_req_st * cmd)
```

submit a command to ehsm dispatcher [in] object of command.

##### Returns

0 for success, negtive values for error.

Definition at line 84 of file eHSM\_Dspt\_Ip.c.

#### 4.12.1.6 hw\_interrupt\_disable()

```
void hw_interrupt_disable ()
```

#### 4.12.1.7 hw\_interrupt\_enable()

```
void hw_interrupt_enable ()
```

#### 4.12.1.8 ptest\_time\_counting\_end()

```
void ptest_time_counting_end (
 ehsm_uint32_t time_id)
```

#### 4.12.1.9 ptest\_time\_counting\_get\_state()

```
ehsm_uint32_t ptest_time_counting_get_state (
 ehsm_uint32_t time_id)
```

## 4.13 eHSM\_Dspt\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
```

## Functions

- `ehsm_uint32_t ehsm_dispatcher_init (void)`
- `ehsm_uint32_t ehsm_submit_cmd_req (ehsm_cmd_req_st *cmd)`
- `void Crypto_MainFunction (void)`
- `ehsm_uint32_t ehsm_remove_cmd_from_queue (ehsm_cmd_req_st *cmd)`

### 4.13.1 Function Documentation

#### 4.13.1.1 Crypto\_MainFunction()

```
void Crypto_MainFunction (
 void)
```

Definition at line 195 of file eHSM\_Dspt\_Ip.c.

#### 4.13.1.2 ehsm\_dispatcher\_init()

```
ehsm_uint32_t ehsm_dispatcher_init (
 void)
```

init ehsm dispatcher

##### Returns

0 for success, negative values for error.

Definition at line 181 of file eHSM\_Dspt\_Ip.c.

#### 4.13.1.3 ehsm\_remove\_cmd\_from\_queue()

```
ehsm_uint32_t ehsm_remove_cmd_from_queue (
 ehsm_cmd_req_st * cmd)
```

Definition at line 146 of file eHSM\_Dspt\_Ip.c.

#### 4.13.1.4 ehsm\_submit\_cmd\_req()

```
ehsm_uint32_t ehsm_submit_cmd_req (
 ehsm_cmd_req_st * cmd)
```

submit a command to ehsm dispatcher [in] object of command.

##### Returns

0 for success, negative values for error.

Definition at line 84 of file eHSM\_Dspt\_Ip.c.

## 4.14 eHSM\_Err\_Code\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
```

### Macros

- #define [EHSM\\_ERR\\_MAILBOX\\_SUCCESS](#) 0xA55AU  
*Error code definitions.*
- #define [EHSM\\_ERR\\_SW\\_SUCCESS](#) 0x0U  
*No error.*
- #define [EHSM\\_ERR\\_GENERAL\\_ERROR](#) 0x01U  
*Error not covered by the following.*
- #define [EHSM\\_ERR\\_IPCORE\\_TRNG\\_BUFFER\\_NULL](#) 0x1U  
*TRNG general error.*
- #define [EHSM\\_ERR\\_IPCORE\\_TRNG\\_INVALID\\_INPUT](#) 0x2U
- #define [EHSM\\_ERR\\_IPCORE\\_TRNG\\_INVALID\\_CONFIG](#) 0x3U
- #define [EHSM\\_ERR\\_IPCORE\\_TRNG\\_HT\\_ERROR](#) 0x4U
- #define [EHSM\\_ERR\\_IPCORE\\_TRNG\\_TIMEOUT\\_ERROR](#) 0x5U
- #define [EHSM\\_ERR\\_IPCORE\\_TRNG\\_ERROR](#) 0x6U
- #define [EHSM\\_ERR\\_IPCORE\\_SKE\\_BUFFER\\_NULL](#) 0x1U  
*SKE general error.*
- #define [EHSM\\_ERR\\_IPCORE\\_SKE\\_CONFIG\\_INVALID](#) 0x2U
- #define [EHSM\\_ERR\\_IPCORE\\_SKE\\_INPUT\\_INVALID](#) 0x3U
- #define [EHSM\\_ERR\\_IPCORE\\_SKE\\_ATTACK\\_ALARM](#) 0x4U
- #define [EHSM\\_ERR\\_IPCORE\\_SKE\\_PADDING\\_ERROR](#) 0x5U
- #define [EHSM\\_ERR\\_IPCORE\\_SKE\\_ERROR](#) 0x6U
- #define [EHSM\\_ERR\\_IPCORE\\_HASH\\_BUFFER\\_NULL](#) 0x1U  
*HASH general error.*
- #define [EHSM\\_ERR\\_IPCORE\\_HASH\\_CONFIG\\_INVALID](#) 0x2U
- #define [EHSM\\_ERR\\_IPCORE\\_HASH\\_INPUT\\_INVALID](#) 0x3U
- #define [EHSM\\_ERR\\_IPCORE\\_HASH\\_LEN\\_OVERFLOW](#) 0x4U
- #define [EHSM\\_ERR\\_IPCORE\\_HASH\\_OUTPUT\\_ZERO\\_ALL](#) 0x5U
- #define [EHSM\\_ERR\\_IPCORE\\_HASH\\_ERROR](#) 0x6U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_NO\\_MODINV](#) 0x1U  
*PKE general error.*
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_NOT\\_ON\\_CURVE](#) 0x2U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_INFINITY\\_POINT](#) 0x3U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_ZERO\\_ALL](#) 0x4U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_INTEGER\\_TOO\\_BIG](#) 0x5U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_INVALID\\_INPUT](#) 0x6U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_FINISHED](#) 0x7U
- #define [EHSM\\_ERR\\_IPCORE\\_PKE\\_ERROR](#) 0x8U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_BUFFER\\_NULL](#) 0x40U  
*PKE specific algorithm error.*
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_NOT\\_ON\\_CURVE](#) 0x41U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_EXCHANGE\\_ROLE\\_INVALID](#) 0x42U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_INPUT\\_INVALID](#) 0x43U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_ZERO\\_ALL](#) 0x44U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_INTEGER\\_TOO\\_BIG](#) 0x45U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_VERIFY\\_FAILED](#) 0x46U
- #define [EHSM\\_ERR\\_IPCORE\\_SM2\\_DECRYPT\\_VERIFY\\_FAILED](#) 0x47U
- #define [EHSM\\_ERR\\_IPCORE\\_ECDSA\\_POINTOR\\_NULL](#) 0x50U

- #define `EHSM_ERR_IPCORE_ECDSA_INVALID_INPUT` 0x51U
- #define `EHSM_ERR_IPCORE_ECDSA_ZERO_ALL` 0x52U
- #define `EHSM_ERR_IPCORE_ECDSA_INTEGER_TOO_BIG` 0x53U
- #define `EHSM_ERR_IPCORE_ECDSA_VERIFY_FAILED` 0x54U
- #define `EHSM_ERR_IPCORE_ECDH_POINTOR_NULL` 0x60U
- #define `EHSM_ERR_IPCORE_ECDH_INVALID_INPUT` 0x61U
- #define `EHSM_ERR_IPCORE_ECDH_ZERO_ALL` 0x62U
- #define `EHSM_ERR_IPCORE_ECDH_INTEGER_TOO_BIG` 0x63U
- #define `EHSM_ERR_IPCORE_DH_POINTER_NULL` 0xA0U
- #define `EHSM_ERR_IPCORE_DH_INVALID_INPUT` 0xA1U
- #define `EHSM_ERR_IPCORE_DH_ZERO_ALL` 0xA2U
- #define `EHSM_ERR_IPCORE_DH_VALUE_ONE` 0xA3U
- #define `EHSM_ERR_IPCORE_DH_INTEGER_TOO_BIG` 0xA4U
- #define `EHSM_ERR_IPCORE_ECIES_POINTOR_NULL` 0x100U
- #define `EHSM_ERR_IPCORE_ECIES_INVALID_INPUT` 0x101U
- #define `EHSM_ERR_IPCORE_ECIES_ZERO_ALL` 0x102U
- #define `EHSM_ERR_IPCORE_ECIES_INTEGER_TOO_BIG` 0x103U
- #define `EHSM_ERR_IPCORE_ECIES_ERROR` 0x104U
- #define `EHSM_ERR_NOT_NIT` 0x0F00U  
*The crypto driver is not initialized.*
- #define `EHSM_ERR_OUT_OF_MEM` 0x0F01U  
*No free space to handle the request command.*
- #define `EHSM_ERR_EHSM_BUSY` 0x0F02U  
*The service request failed because the service is still busy. Value is 0x02U.*
- #define `EHSM_ERR_QUEUE_FULL` 0x0F03U  
*The service request failed because the queue is full. Value is 0x05U.*
- #define `EHSM_ERR_JOB_CANCELED` 0x0F04U  
*The service request failed because the Job has been canceled. Value is 0x0BU.*
- #define `EHSM_ERR_SMALL_BUFFER` 0x0F05U  
*The service request failed because the provided buffer is too small to store the result. Value is 0x03U.*
- #define `EHSM_ERR_PARAM_ERROR` 0x0F06U  
*General parameter wrong.*
- #define `EHSM_ERR_NOT_SUPPORT` 0x0F07U  
*The alg, mode, padding or required feature is not supported.*
- #define `EHSM_ERR_WRONG_MODULE_TYPE` 0x0F08U  
*Wrong module type for module\_status\_cmd.*
- #define `EHSM_ERR_WRONG_JOB_ID` 0x0F09U  
*Wrong job id for cancel\_cmd.*
- #define `EHSM_ERR_CMD_CANCELED` 0x0F0AU  
*The service request failed because the synchronous Job has been canceled.*
- #define `EHSM_ERR_CODE_MOVE_ERROR` 0x0F0BU  
*move code into iram error*
- #define `EHSM_ERR_WRONG_KEY_USAGE` 0x1E00U  
*Wrong key usage.*
- #define `EHSM_ERR_WRONG_KEY_HANDLE` 0x1E01U /\*Given key handle is wrong.\*/
- #define `EHSM_ERR_AUTH_FAILED` 0x1E02U /\*Given authorization value was wrong.\*/
- #define `EHSM_ERR_WRONG_KEY_LIFE_LIMIT` 0x1E03U /\*Wrong key life limitation.\*/
- #define `EHSM_ERR_WRONG_KEY_TYPE` 0x1E04U /\*Wrong key type.\*/
- #define `EHSM_ERR_WRONG_KEY_SIZE` 0x1E05U /\*Wrong key size.\*/
- #define `EHSM_ERR_WRONG_KEY_DERIVE_FUNC` 0x1E06U /\*Wrong key derivation function.\*/
- #define `EHSM_ERR_WRONG_SALT_SIZE` 0x1E07U /\*Wrong salt size for key derivation.\*/
- #define `EHSM_ERR_KEY_BUFF_SMALLER` 0x1E08U /\*Buffer for key storage is too small.\*/
- #define `EHSM_ERR_KEY_STORE_FULL` 0x1E09U /\*Key storage in eHSM is full.\*/
- #define `EHSM_ERR_WRONG_KEY_LEVEL` 0x1E0AU /\*Wrong key level for bootrom.\*/
- #define `EHSM_ERR_WRONG_PUB_KEY` 0x1E0BU /\*Wrong public key for debug authentication.\*/

- #define `EHSM_ERR_TRANSPORT_IMPOSSIBLE` 0x1E0CU /\*Give transport key is not a capable transport key.\*/
- #define `EHSM_ERR_ALL_KEY_SPACE_OCCUPIED` 0x1E0DU /\*Evita error: no resources left to create/import an additional key\*/
- #define `EHSM_ERR_REMOVE_IMPOSSIBLE` 0x1E0EU /\*Evita error: key is not allowed to become removed\*/
- #define `EHSM_ERR_WRONG_CERT_KEY_HANDLE` 0x1E0FU /\*Evita error: given certification key handle is unknown wrong\*/
- #define `EHSM_ERR_ALGORITHM_ERROR` 0x1E10U /\*Evita error: Algorithm or mode not available\*/
- #define `EHSM_ERR_WRONG_AUTHORIZATION` 0x1E11U /\*Evita error: given athorization structure does not fit key flags\*/
- #define `EHSM_ERR_INVALID_KEY_FLAG` 0x1E12U /\*Evita error: given key flag is vinvalid (e.g., wrong combination)\*/
- #define `EHSM_ERR_WRONG_REMOTE_KEY_HANDLE` 0x1E13U /\*Evita error: Given remote key handle is unknown or wrong\*/
- #define `EHSM_ERR_WRONG_KEY_COMBINATION` 0x1E14U /\*Keys do not fit the algorithm (e.g., RSA key vs. ECDH)\*/
- #define `EHSM_ERR_KEY_EMPTY_ERROR` 0x1E18U /\*key data empty \*/
- #define `EHSM_ERR_WRITE_PROTECTED` 0x1E15U /\*She key has been write protect.\*/
- #define `EHSM_ERR_KEY_UPDATE_ERROR` 0x1E16U /\*She key update error\*/
- #define `EHSM_ERR_KEY_INVALID_ERROR` 0x1E17U /\*She key invalid\*/
- #define `EHSM_ERR_KEY_NOT_AVAILABLE_ERROR` 0x1E19U /\*She key not available \*/
- #define `EHSM_ERR_MEMORY_FAILURE` 0x1E1AU /\*She key memory failure \*/
- #define `EHSM_ERR_KMGR_READ_ERROR` 0x1E1BU  
*ehsm kmgr module read fail*
- #define `EHSM_ERR_SKE_WRONG_ALG` 0x2D00U  
*Wrong algorithm.*
- #define `EHSM_ERR_SKE_WRONG_MODE` 0x2D01U  
*Wrong mode for the algorithm.*
- #define `EHSM_ERR_SKE_WRONG_PADIDNG` 0x2D02U  
*Wrong padding for the algorithm.*
- #define `EHSM_ERR_SKE_WRONG_K_SIZE` 0x2D03U  
*Wrong key size for the algorithm.*
- #define `EHSM_ERR_SKE_WRONG_IV_SIZE` 0x2D04U  
*Wrong iv size for the algorithm.*
- #define `EHSM_ERR_SKE_IV_SHOULD_NOT_NULL` 0x2D05U  
*Iv should not be null, but it's null.*
- #define `EHSM_ERR_SKE_WRONG_TAG_SIZE` 0x2D06U  
*Wrong tag size.*
- #define `EHSM_ERR_SKE_WRONG_L_SIZE` 0x2D07U  
*Wrong L size (only for CCM).*
- #define `EHSM_ERR_SKE_WRONG_AAD_SIZE` 0x2D08U  
*Wrong aad size (only fro GCM and CCM)*
- #define `EHSM_ERR_SKE_MAC_VRY_FAILED` 0x2D09U  
*MAC verify failed.*
- #define `EHSM_ERR_PKE_WRONG_CURVE_ID` 0x2D20U  
*Wrong curve\_id for ECIES or ECCP. Reserved some values for ske.*
- #define `EHSM_ERR_PKE_WRONG_RSA_CRT_MODE` 0x2D21U  
*Wrong CRT mode for RSA.*
- #define `EHSM_ERR_PKE_WRONG_E_SIZE` 0x2D22U  
*Wrong e value size for RSA key generation.*
- #define `EHSM_ERR_PKE_WRONG_N_SIZE` 0x2D23U  
*Wrong n value size for RSA key generation.*
- #define `EHSM_ERR_PKE_SIGN_FAILED` 0x2D24U  
*PKE sign failed.*
- #define `EHSM_ERR_PKE_VERIFY_FAILED` 0x2D25U

- PKE verify failed.*
- #define [EHSM\\_ERR\\_PKE\\_ED25519\\_MSG\\_FLOW](#) 0x2D26U  
*Message size is overflow for ed25519.*
- #define [EHSM\\_ERR\\_PKE\\_WRONG\\_KDF\\_ALG](#) 0x2D27U  
*Wrong KDF algorithm for ECIES.*
- #define [EHSM\\_ERR\\_HMAC\\_VERIFY\\_FAILED](#) 0x2D40U  
*HMAC verify failed.*
- #define [EHSM\\_ERR\\_HASH\\_BUFFER\\_NULL](#) 0x2D41U  
*HASH buffer is NULL.*
- #define [EHSM\\_ERR\\_HASH\\_CONFIG\\_INVALID](#) 0x2D42U  
*HASH configuration invalid.*
- #define [EHSM\\_ERR\\_HASH\\_INPUT\\_INVALID](#) 0x2D43U  
*HASH input invalid.*
- #define [EHSM\\_ERR\\_HASH\\_LEN\\_OVER\\_FLOW](#) 0x2D44U  
*HASH input overflow.*
- #define [EHSM\\_ERR\\_HASH\\_OUTPUT\\_ZERO\\_ALL](#) 0x2D45U  
*HASH output data all zero.*
- #define [EHSM\\_ERR\\_TRNG\\_BUFFER\\_NULL](#) 0x2D60U  
*TRNG buffer is NULL.*
- #define [EHSM\\_ERR\\_TRNG\\_INVALID\\_INPUT](#) 0x2D61U  
*TRNG input is invalid.*
- #define [EHSM\\_ERR\\_TRNG\\_INVALID\\_CONFIG](#) 0x2D62U  
*TRNG with invalid configuration.*
- #define [EHSM\\_ERR\\_TRNG\\_HT\\_ERROR](#) 0x2D63U  
*TRNG HT error.*
- #define [EHSM\\_ERR\\_TRNG\\_TIMEOUT\\_ERROR](#) 0x2D64U  
*TRNG timeout while working.*
- #define [EHSM\\_ERR\\_DRBG\\_RESEED\\_FAILED](#) 0x2D65U  
*CTR\_DRBG reseed fail.*
- #define [EHSM\\_ERR\\_DRBG\\_BUFFER\\_NULL](#) 0x2D66U  
*CTR\_DRBG buffer is NULL.*
- #define [EHSM\\_ERR\\_DRBG\\_LENGTH\\_INVALID](#) 0x2D67U  
*CTR\_DRBG input length is invalid.*
- #define [EHSM\\_ERR\\_DRBG\\_LENGTH\\_NOT\\_MUL\\_8](#) 0x2D68U  
*CTR\_DRBG input length is not multiple to 8.*
- #define [EHSM\\_ERR\\_DRBG\\_DF\\_OVERFLOW](#) 0x2D69U  
*CTR\_DRBG DF overflow.*
- #define [EHSM\\_ERR\\_CRYPT\\_CERT\\_PARSE\\_FAILED](#) 0x2D80U  
*Certificate parse fail.*
- #define [EHSM\\_ERR\\_CRYPT\\_CERT\\_VERIFY\\_FAILED](#) 0x2D81U  
*Certificate verify failed.*
- #define [EHSM\\_ERR\\_CRYPT\\_SM9\\_WRONG\\_ID\\_SIZE](#) 0x2D90U  
*Wrong id size for SM9.*
- #define [EHSM\\_ERR\\_CRYPT\\_SM9\\_WRONG\\_K2\\_SIZE](#) 0x2D91U  
*Wrong key2 size for SM9.*
- #define [EHSM\\_ERR\\_TRNG\\_WORK\\_ERROR](#) 0x2DFCU  
*TRNG IP work general error by crypto software.*
- #define [EHSM\\_ERR\\_SKE\\_WORK\\_ERROR](#) 0x2DFDU  
*SKE IP work err by crypto software.*
- #define [EHSM\\_ERR\\_PKE\\_WORK\\_ERROR](#) 0x2DFEU  
*PKE IP work err by crypto software.*
- #define [EHSM\\_ERR\\_HASH\\_WORK\\_ERROR](#) 0x2DFFU  
*HASH IP work err by crypto software.*



- #define [EHSM\\_ERR\\_WRONG\\_PROC\\_MODE](#) 0x3C00U  
*Given process mode is wrong.*
- #define [EHSM\\_ERR\\_WRONG\\_CONTEXT](#) 0x3C01U  
*Given context was wrong.*
- #define [EHSM\\_ERR\\_WRONG\\_ALGORITHM](#) 0x3C02U  
*Given algorithm or algorithm mode not available.*
- #define [EHSM\\_ERR\\_WRONG\\_DIRECT](#) 0x3C03U  
*Direction is wrong for cipher/hash/signature/verification.*
- #define [EHSM\\_ERR\\_WRONG\\_PADDING\\_TYPE](#) 0x3C04U  
*Wrong padding type for cipher/mac/signature/verification.*
- #define [EHSM\\_ERR\\_WRONG\\_EHSM\\_ADDR](#) 0x3C05U  
*Wrong eHSM address for otp writing or reading.*
- #define [EHSM\\_ERR\\_WRONG\\_DATA\\_LENGTH](#) 0x3C06U  
*Wrong data length.*
- #define [EHSM\\_ERR\\_EHSM\\_LIFECYCLE\\_LIMIT](#) 0x3C07U  
*function is limited by lifecycle*
- #define [EHSM\\_ERR\\_WRONG\\_VERSION\\_COUNTER](#) 0x3C08U  
*Wrong version counter.*
- #define [EHSM\\_ERR\\_EQUAL\\_VERSION\\_COUNTER](#) 0x3C09U  
*Wrong version counter.*
- #define [EHSM\\_ERR\\_DATA\\_CHECK\\_ERROR](#) 0x3C0AU  
*two parts of data is not the same*
- #define [EHSM\\_ERR\\_INVALID\\_CODE\\_FLAG](#) 0x3C0BU  
*code flag is invalid*
- #define [EHSM\\_ERR\\_REG\\_TWICE\\_NOT\\_MATCH](#) 0x3C0CU  
*The twice operation result does not match.*
- #define [EHSM\\_ERR\\_ERC\\_NO\\_SECURE\\_BOOT](#) 0x3C0DU  
*No secure boot is done.*
- #define [EHSM\\_ERR\\_IMAGE\\_VERIFY\\_FAILED](#) 0x3C0EU  
*Image verify failed for secure boot.*
- #define [EHSM\\_ERR\\_MIDDLE\\_SW](#) 0x3C0FU  
*This is special value for success in secure boot, image upgrade module.*
- #define [EHSM\\_ERR\\_WRONG\\_CHALLENGE\\_TYPE](#) 0x3C20U  
*Wrong challenge type for getting challenge.*
- #define [EHSM\\_ERR\\_CHALLENGE\\_FAILED](#) 0x3C21U  
*Failed for challenge.*
- #define [EHSM\\_ERR\\_CHALLENGE\\_EXPIRED](#) 0x3C22U  
*The challenge has been expired.*
- #define [EHSM\\_ERR\\_NO\\_CHALLENGE\\_AVAILABLE](#) 0x3C23U  
*No challenge available for the authentication.*
- #define [EHSM\\_ERR\\_DEBUG\\_AUTH\\_FAILED](#) 0x3C24U  
*Debug authentication was failed.*
- #define [EHSM\\_ERR\\_COUNTER\\_AUTH\\_FAILED](#) 0x5A00U  
*Wrong counter access authorization value.*
- #define [EHSM\\_ERR\\_COUNTER\\_NOT\\_INIT](#) 0x5A01U  
*Counter not initialization.*
- #define [EHSM\\_ERR\\_COUNTER\\_WRONG\\_ID](#) 0x5A02U  
*Wrong id index.*
- #define [EHSM\\_ERR\\_ALL\\_COUNTER\\_BUSY](#) 0x5A03U  
*All counters are busy.*
- #define [EHSM\\_ERR\\_INVALID\\_COUNTER\\_INCREMENTATION](#) 0x5A04U  
*Invalid incrementaion of counter.*
- #define [EHSM\\_ERR\\_CHECK\\_TIME\\_FAILED](#) 0x5A20U

*Failed for timestamp checking.*

- #define [EHSM\\_ERR\\_WRONG\\_UTC\\_TIME](#) 0x5A21U  
*Wrong UTC time.*
- #define [EHSM\\_ERR\\_UTC\\_TIMER\\_NOT\\_SYNC](#) 0x5A22U  
*UTC timer is not synchronized.*
- #define [EHSM\\_ERR\\_UTC\\_TIMER\\_INVALID\\_INDEX](#) 0x5A23U  
*UTC timer index is invalid.*
- #define [EHSM\\_ERR\\_TIME\\_STAMP\\_VERIFY\\_FAILED](#) 0x5A24U  
*Time stamp verify failed.*
- #define [EHSM\\_ERR\\_TIME\\_STAMP\\_EXPIRED](#) 0x5A25U  
*Time stamp is expired.*
- #define [EHSM\\_ERR\\_CHECK\\_TIME\\_STAMP\\_ERROR](#) 0x5A26U  
*Verify time stamp failed.*
- #define [EHSM\\_ERR\\_UTC\\_SYNCHRONIZATION\\_FAILED](#) 0x5A27U  
*UTC time synchronization failed.*
- #define [EHSM\\_ERR\\_TIME\\_CHALLENGE\\_EXPIRED](#) 0x5A28U  
*Time challenge expired.*
- #define [EHSM\\_ERR\\_DATA\\_EMPTY](#) 0x5A29U  
*Item data is empty.*

#### 4.14.1 Macro Definition Documentation

##### 4.14.1.1 EHSM\_ERR\_ALGORITHM\_ERROR

```
#define EHSM_ERR_ALGORITHM_ERROR 0x1E10U /*Evita error: Algorithm or mode not available*/
```

Definition at line 233 of file eHSM\_Err\_Code\_Ip.h.

##### 4.14.1.2 EHSM\_ERR\_ALL\_COUNTER\_BUSY

```
#define EHSM_ERR_ALL_COUNTER_BUSY 0x5A03U
```

All counters are busy.

Definition at line 592 of file eHSM\_Err\_Code\_Ip.h.

##### 4.14.1.3 EHSM\_ERR\_ALL\_KEY\_SPACE\_OCCUPIED

```
#define EHSM_ERR_ALL_KEY_SPACE_OCCUPIED 0x1E0DU /*Evita error: no resources left to create/import
an additional key*/
```

Definition at line 230 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.4 EHSM\_ERR\_AUTH\_FAILED

```
#define EHSM_ERR_AUTH_FAILED 0x1E02U /*Given authorization value was wrong.*/
```

Definition at line 219 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.5 EHSM\_ERR\_CHALLENGE\_EXPIRED

```
#define EHSM_ERR_CHALLENGE_EXPIRED 0x3C22U
```

The challenge has been expired.

Definition at line 562 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.6 EHSM\_ERR\_CHALLENGE\_FAILED

```
#define EHSM_ERR_CHALLENGE_FAILED 0x3C21U
```

Failed for challenge.

Definition at line 557 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.7 EHSM\_ERR\_CHECK\_TIME\_FAILED

```
#define EHSM_ERR_CHECK_TIME_FAILED 0x5A20U
```

Failed for timestamp checking.

Definition at line 602 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.8 EHSM\_ERR\_CHECK\_TIME\_STAMP\_ERROR

```
#define EHSM_ERR_CHECK_TIME_STAMP_ERROR 0x5A26U
```

Verify time stamp failed.

Definition at line 632 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.9 EHSM\_ERR\_CMD\_CANCELED

```
#define EHSM_ERR_CMD_CANCELED 0x0F0AU
```

The service request failed because the synchronous Job has been canceled.

Definition at line 207 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.10 EHSM\_ERR\_CODE\_MOVE\_ERROR

```
#define EHSM_ERR_CODE_MOVE_ERROR 0x0F0BU
```

move code into iram error

Definition at line 212 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.11 EHSM\_ERR\_COUNTER\_AUTH\_FAILED

```
#define EHSM_ERR_COUNTER_AUTH_FAILED 0x5A00U
```

Wrong counter access authorization value.

Definition at line 577 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.12 EHSM\_ERR\_COUNTER\_NOT\_INIT

```
#define EHSM_ERR_COUNTER_NOT_INIT 0x5A01U
```

Counter not initialization.

Definition at line 582 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.13 EHSM\_ERR\_COUNTER\_WRONG\_ID

```
#define EHSM_ERR_COUNTER_WRONG_ID 0x5A02U
```

Wrong id index.

Definition at line 587 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.14 EHSM\_ERR\_CRYPTOCERT\_PARSE\_FAILED**

```
#define EHSM_ERR_CRYPTOCERT_PARSE_FAILED 0x2D80U
```

Certificate parse fail.

Definition at line 424 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.15 EHSM\_ERR\_CRYPTOCERT\_VERIFY\_FAILED**

```
#define EHSM_ERR_CRYPTOCERT_VERIFY_FAILED 0x2D81U
```

Certificate verify failed.

Definition at line 429 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.16 EHSM\_ERR\_CRYPTOSM9\_WRONG\_ID\_SIZE**

```
#define EHSM_ERR_CRYPTOSM9_WRONG_ID_SIZE 0x2D90U
```

Wrong id size for SM9.

Definition at line 434 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.17 EHSM\_ERR\_CRYPTOSM9\_WRONG\_K2\_SIZE**

```
#define EHSM_ERR_CRYPTOSM9_WRONG_K2_SIZE 0x2D91U
```

Wrong key2 size for SM9.

Definition at line 439 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.18 EHSM\_ERR\_DATA\_CHECK\_ERROR**

```
#define EHSM_ERR_DATA_CHECK_ERROR 0x3C0AU
```

two parts of data is not the same

Definition at line 522 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.19 EHSM\_ERR\_DATA\_EMPTY**

```
#define EHSM_ERR_DATA_EMPTY 0x5A29U
```

Item data is empty.

Definition at line 647 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.20 EHSM\_ERR\_DEBUG\_AUTH\_FAILED**

```
#define EHSM_ERR_DEBUG_AUTH_FAILED 0x3C24U
```

Debug authentication was failed.

Definition at line 572 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.21 EHSM\_ERR\_DRBG\_BUFFER\_NULL**

```
#define EHSM_ERR_DRBG_BUFFER_NULL 0x2D66U
```

CTR\_DRBG buffer is NULL.

Definition at line 404 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.22 EHSM\_ERR\_DRBG\_DF\_OVERFLOW**

```
#define EHSM_ERR_DRBG_DF_OVERFLOW 0x2D69U
```

CTR\_DRBG DF overflow.

Definition at line 419 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.23 EHSM\_ERR\_DRBG\_LENGTH\_INVALID**

```
#define EHSM_ERR_DRBG_LENGTH_INVALID 0x2D67U
```

CTR\_DRBG input length is invalid.

Definition at line 409 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.24 EHSM\_ERR\_DRBG\_LENGTH\_NOT\_MUL\_8

```
#define EHSM_ERR_DRBG_LENGTH_NOT_MUL_8 0x2D68U
```

CTR\_DRBG input length is not multiple to 8.

Definition at line 414 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.25 EHSM\_ERR\_DRBG\_RESEED\_FAILED

```
#define EHSM_ERR_DRBG_RESEED_FAILED 0x2D65U
```

CTR\_DRBG reseed fail.

Definition at line 399 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.26 EHSM\_ERR\_EHSM\_BUSY

```
#define EHSM_ERR_EHSM_BUSY 0x0F02U
```

The service request failed because the service is still busy. Value is 0x02U.

Definition at line 164 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.27 EHSM\_ERR\_EHSM\_LIFECYCLE\_LIMIT

```
#define EHSM_ERR_EHSM_LIFECYCLE_LIMIT 0x3C07U
```

function is limited by lifecycle

Definition at line 507 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.28 EHSM\_ERR\_EQUAL\_VERSION\_COUNTER

```
#define EHSM_ERR_EQUAL_VERSION_COUNTER 0x3C09U
```

Wrong version counter.

Definition at line 517 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.29 EHSM\_ERR\_ERC\_NO\_SECURE\_BOOT

```
#define EHSM_ERR_ERC_NO_SECURE_BOOT 0x3C0DU
```

No secure boot is done.

Definition at line 537 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.30 EHSM\_ERR\_GENERAL\_ERROR

```
#define EHSM_ERR_GENERAL_ERROR 0x01U
```

Error not covered by the following.

Definition at line 64 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.31 EHSM\_ERR\_HASH\_BUFFER\_NULL

```
#define EHSM_ERR_HASH_BUFFER_NULL 0x2D41U
```

HASH buffer is NULL.

Definition at line 349 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.32 EHSM\_ERR\_HASH\_CONFIG\_INVALID

```
#define EHSM_ERR_HASH_CONFIG_INVALID 0x2D42U
```

HASH configuration invalid.

Definition at line 354 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.33 EHSM\_ERR\_HASH\_INPUT\_INVALID

```
#define EHSM_ERR_HASH_INPUT_INVALID 0x2D43U
```

HASH input invalid.

Definition at line 359 of file eHSM\_Err\_Code\_Ip.h.



**4.14.1.34 EHSM\_ERR\_HASH\_LEN\_OVER\_FLOW**

```
#define EHSM_ERR_HASH_LEN_OVER_FLOW 0x2D44U
```

HASH input overflow.

Definition at line 364 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.35 EHSM\_ERR\_HASH\_OUTPUT\_ZERO\_ALL**

```
#define EHSM_ERR_HASH_OUTPUT_ZERO_ALL 0x2D45U
```

HASH output data all zero.

Definition at line 369 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.36 EHSM\_ERR\_HASH\_WORK\_ERROR**

```
#define EHSM_ERR_HASH_WORK_ERROR 0x2DFFU
```

HASH IP work err by crypto software.

**Note**

Not recommend this error code since we don't know what happens.

Definition at line 467 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.37 EHSM\_ERR\_HMAC\_VERIFY\_FAILED**

```
#define EHSM_ERR_HMAC_VERIFY_FAILED 0x2D40U
```

HMAC verify failed.

Definition at line 344 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.38 EHSM\_ERR\_IMAGE\_VERIFY\_FAILED**

```
#define EHSM_ERR_IMAGE_VERIFY_FAILED 0x3C0EU
```

Image verify failed for secure boot.

Definition at line 542 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.39 EHSM\_ERR\_INVALID\_CODE\_FLAG

```
#define EHSM_ERR_INVALID_CODE_FLAG 0x3C0BU
```

code flag is invalid

Definition at line 527 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.40 EHSM\_ERR\_INVALID\_COUNTER\_INCREMENTATION

```
#define EHSM_ERR_INVALID_COUNTER_INCREMENTATION 0x5A04U
```

Invalid incrementaion of counter.

Definition at line 597 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.41 EHSM\_ERR\_INVALID\_KEY\_FLAG

```
#define EHSM_ERR_INVALID_KEY_FLAG 0x1E12U /*Evita error: given key flag is vinvalid (e.g., wrong
combination)*/
```

Definition at line 235 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.42 EHSM\_ERR\_IPCORE\_DH\_INTEGER\_TOO\_BIG

```
#define EHSM_ERR_IPCORE_DH_INTEGER_TOO_BIG 0xA4U
```

Definition at line 142 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.43 EHSM\_ERR\_IPCORE\_DH\_INVALID\_INPUT

```
#define EHSM_ERR_IPCORE_DH_INVALID_INPUT 0xA1U
```

Definition at line 139 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.44 EHSM\_ERR\_IPCORE\_DH\_POINTER\_NULL

```
#define EHSM_ERR_IPCORE_DH_POINTER_NULL 0xA0U
```

Definition at line 138 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.45 EHSM\_ERR\_IPCORE\_DH\_VALUE\_ONE**

```
#define EHSM_ERR_IPCORE_DH_VALUE_ONE 0xA3U
```

Definition at line 141 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.46 EHSM\_ERR\_IPCORE\_DH\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_DH_ZERO_ALL 0xA2U
```

Definition at line 140 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.47 EHSM\_ERR\_IPCORE\_ECDH\_INTEGER\_TOO\_BIG**

```
#define EHSM_ERR_IPCORE_ECDH_INTEGER_TOO_BIG 0x63U
```

Definition at line 135 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.48 EHSM\_ERR\_IPCORE\_ECDH\_INVALID\_INPUT**

```
#define EHSM_ERR_IPCORE_ECDH_INVALID_INPUT 0x61U
```

Definition at line 133 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.49 EHSM\_ERR\_IPCORE\_ECDH\_POINTOR\_NULL**

```
#define EHSM_ERR_IPCORE_ECDH_POINTOR_NULL 0x60U
```

Definition at line 132 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.50 EHSM\_ERR\_IPCORE\_ECDH\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_ECDH_ZERO_ALL 0x62U
```

Definition at line 134 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.51 EHSM\_ERR\_IPCORE\_ECDSA\_INTEGER\_TOO\_BIG**

```
#define EHSM_ERR_IPCORE_ECDSA_INTEGER_TOO_BIG 0x53U
```

Definition at line 129 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.52 EHSM\_ERR\_IPCORE\_ECDSA\_INVALID\_INPUT**

```
#define EHSM_ERR_IPCORE_ECDSA_INVALID_INPUT 0x51U
```

Definition at line 127 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.53 EHSM\_ERR\_IPCORE\_ECDSA\_POINTOR\_NULL**

```
#define EHSM_ERR_IPCORE_ECDSA_POINTOR_NULL 0x50U
```

Definition at line 126 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.54 EHSM\_ERR\_IPCORE\_ECDSA\_VERIFY\_FAILED**

```
#define EHSM_ERR_IPCORE_ECDSA_VERIFY_FAILED 0x54U
```

Definition at line 130 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.55 EHSM\_ERR\_IPCORE\_ECDSA\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_ECDSA_ZERO_ALL 0x52U
```

Definition at line 128 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.56 EHSM\_ERR\_IPCORE\_ECIES\_ERROR**

```
#define EHSM_ERR_IPCORE_ECIES_ERROR 0x104U
```

Definition at line 148 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.57 EHSM\_ERR\_IPCORE\_ECIES\_INTEGER\_TOO\_BIG**

```
#define EHSM_ERR_IPCORE_ECIES_INTEGER_TOO_BIG 0x103U
```

Definition at line 147 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.58 EHSM\_ERR\_IPCORE\_ECIES\_INVALID\_INPUT**

```
#define EHSM_ERR_IPCORE_ECIES_INVALID_INPUT 0x101U
```

Definition at line 145 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.59 EHSM\_ERR\_IPCORE\_ECIES\_POINTOR\_NULL**

```
#define EHSM_ERR_IPCORE_ECIES_POINTOR_NULL 0x100U
```

Definition at line 144 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.60 EHSM\_ERR\_IPCORE\_ECIES\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_ECIES_ZERO_ALL 0x102U
```

Definition at line 146 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.61 EHSM\_ERR\_IPCORE\_HASH\_BUFFER\_NULL**

```
#define EHSM_ERR_IPCORE_HASH_BUFFER_NULL 0x1U
```

HASH general error.

Definition at line 89 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.62 EHSM\_ERR\_IPCORE\_HASH\_CONFIG\_INVALID**

```
#define EHSM_ERR_IPCORE_HASH_CONFIG_INVALID 0x2U
```

Definition at line 90 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.63 EHSM\_ERR\_IPCORE\_HASH\_ERROR**

```
#define EHSM_ERR_IPCORE_HASH_ERROR 0x6U
```

Definition at line 94 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.64 EHSM\_ERR\_IPCORE\_HASH\_INPUT\_INVALID**

```
#define EHSM_ERR_IPCORE_HASH_INPUT_INVALID 0x3U
```

Definition at line 91 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.65 EHSM\_ERR\_IPCORE\_HASH\_LEN\_OVERFLOW**

```
#define EHSM_ERR_IPCORE_HASH_LEN_OVERFLOW 0x4U
```

Definition at line 92 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.66 EHSM\_ERR\_IPCORE\_HASH\_OUTPUT\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_HASH_OUTPUT_ZERO_ALL 0x5U
```

Definition at line 93 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.67 EHSM\_ERR\_IPCORE\_PKE\_ERROR**

```
#define EHSM_ERR_IPCORE_PKE_ERROR 0x8U
```

Definition at line 106 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.68 EHSM\_ERR\_IPCORE\_PKE\_FINISHED**

```
#define EHSM_ERR_IPCORE_PKE_FINISHED 0x7U
```

Definition at line 105 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.69 EHSM\_ERR\_IPCORE\_PKE\_INFINITY\_POINT**

```
#define EHSM_ERR_IPCORE_PKE_INFINITY_POINT 0x3U
```

Definition at line 101 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.70 EHSM\_ERR\_IPCORE\_PKE\_INTEGER\_TOO\_BIG**

```
#define EHSM_ERR_IPCORE_PKE_INTEGER_TOO_BIG 0x5U
```

Definition at line 103 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.71 EHSM\_ERR\_IPCORE\_PKE\_INVALID\_INPUT**

```
#define EHSM_ERR_IPCORE_PKE_INVALID_INPUT 0x6U
```

Definition at line 104 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.72 EHSM\_ERR\_IPCORE\_PKE\_NO\_MODINV**

```
#define EHSM_ERR_IPCORE_PKE_NO_MODINV 0x1U
```

PKE general error.

Definition at line 99 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.73 EHSM\_ERR\_IPCORE\_PKE\_NOT\_ON\_CURVE**

```
#define EHSM_ERR_IPCORE_PKE_NOT_ON_CURVE 0x2U
```

Definition at line 100 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.74 EHSM\_ERR\_IPCORE\_PKE\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_PKE_ZERO_ALL 0x4U
```

Definition at line 102 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.75 EHSM\_ERR\_IPCORE\_SKE\_ATTACK\_ALARM**

```
#define EHSM_ERR_IPCORE_SKE_ATTACK_ALARM 0x4U
```

Definition at line 82 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.76 EHSM\_ERR\_IPCORE\_SKE\_BUFFER\_NULL**

```
#define EHSM_ERR_IPCORE_SKE_BUFFER_NULL 0x1U
```

SKE general error.

Definition at line 79 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.77 EHSM\_ERR\_IPCORE\_SKE\_CONFIG\_INVALID**

```
#define EHSM_ERR_IPCORE_SKE_CONFIG_INVALID 0x2U
```

Definition at line 80 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.78 EHSM\_ERR\_IPCORE\_SKE\_ERROR**

```
#define EHSM_ERR_IPCORE_SKE_ERROR 0x6U
```

Definition at line 84 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.79 EHSM\_ERR\_IPCORE\_SKE\_INPUT\_INVALID**

```
#define EHSM_ERR_IPCORE_SKE_INPUT_INVALID 0x3U
```

Definition at line 81 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.80 EHSM\_ERR\_IPCORE\_SKE\_PADDING\_ERROR**

```
#define EHSM_ERR_IPCORE_SKE_PADDING_ERROR 0x5U
```

Definition at line 83 of file eHSM\_Err\_Code\_Ip.h.



**4.14.1.81 EHSM\_ERR\_IPCORE\_SM2\_BUFFER\_NULL**

```
#define EHSM_ERR_IPCORE_SM2_BUFFER_NULL 0x40U
```

PKE specific algorithm error.

Definition at line 117 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.82 EHSM\_ERR\_IPCORE\_SM2\_DECRYPT\_VERIFY\_FAILED**

```
#define EHSM_ERR_IPCORE_SM2_DECRYPT_VERIFY_FAILED 0x47U
```

Definition at line 124 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.83 EHSM\_ERR\_IPCORE\_SM2\_EXCHANGE\_ROLE\_INVALID**

```
#define EHSM_ERR_IPCORE_SM2_EXCHANGE_ROLE_INVALID 0x42U
```

Definition at line 119 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.84 EHSM\_ERR\_IPCORE\_SM2\_INPUT\_INVALID**

```
#define EHSM_ERR_IPCORE_SM2_INPUT_INVALID 0x43U
```

Definition at line 120 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.85 EHSM\_ERR\_IPCORE\_SM2\_INTEGER\_TOO\_BIG**

```
#define EHSM_ERR_IPCORE_SM2_INTEGER_TOO_BIG 0x45U
```

Definition at line 122 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.86 EHSM\_ERR\_IPCORE\_SM2\_NOT\_ON\_CURVE**

```
#define EHSM_ERR_IPCORE_SM2_NOT_ON_CURVE 0x41U
```

Definition at line 118 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.87 EHSM\_ERR\_IPCORE\_SM2\_VERIFY\_FAILED**

```
#define EHSM_ERR_IPCORE_SM2_VERIFY_FAILED 0x46U
```

Definition at line 123 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.88 EHSM\_ERR\_IPCORE\_SM2\_ZERO\_ALL**

```
#define EHSM_ERR_IPCORE_SM2_ZERO_ALL 0x44U
```

Definition at line 121 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.89 EHSM\_ERR\_IPCORE\_TRNG\_BUFFER\_NULL**

```
#define EHSM_ERR_IPCORE_TRNG_BUFFER_NULL 0x1U
```

TRNG general error.

Definition at line 69 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.90 EHSM\_ERR\_IPCORE\_TRNG\_ERROR**

```
#define EHSM_ERR_IPCORE_TRNG_ERROR 0x6U
```

Definition at line 74 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.91 EHSM\_ERR\_IPCORE\_TRNG\_HT\_ERROR**

```
#define EHSM_ERR_IPCORE_TRNG_HT_ERROR 0x4U
```

Definition at line 72 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.92 EHSM\_ERR\_IPCORE\_TRNG\_INVALID\_CONFIG**

```
#define EHSM_ERR_IPCORE_TRNG_INVALID_CONFIG 0x3U
```

Definition at line 71 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.93 EHSM\_ERR\_IPCORE\_TRNG\_INVALID\_INPUT**

```
#define EHSM_ERR_IPCORE_TRNG_INVALID_INPUT 0x2U
```

Definition at line 70 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.94 EHSM\_ERR\_IPCORE\_TRNG\_TIMEOUT\_ERROR**

```
#define EHSM_ERR_IPCORE_TRNG_TIMEOUT_ERROR 0x5U
```

Definition at line 73 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.95 EHSM\_ERR\_JOB\_CANCELED**

```
#define EHSM_ERR_JOB_CANCELED 0x0F04U
```

The service request failed because the Job has been canceled. Value is 0x0BU.

Definition at line 176 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.96 EHSM\_ERR\_KEY\_BUFF\_SMALLER**

```
#define EHSM_ERR_KEY_BUFF_SMALLER 0x1E08U /*Buffer for key storage is too small.*/
```

Definition at line 225 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.97 EHSM\_ERR\_KEY\_EMPTY\_ERROR**

```
#define EHSM_ERR_KEY_EMPTY_ERROR 0x1E18U /*key data empty */
```

Definition at line 238 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.98 EHSM\_ERR\_KEY\_INVALID\_ERROR**

```
#define EHSM_ERR_KEY_INVALID_ERROR 0x1E17U /*She key invalid*/
```

Definition at line 241 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.99 EHSM\_ERR\_KEY\_NOT\_AVAILABLE\_ERROR

```
#define EHSM_ERR_KEY_NOT_AVAILABLE_ERROR 0x1E19U /*She key not available */
```

Definition at line 242 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.100 EHSM\_ERR\_KEY\_STORE\_FULL

```
#define EHSM_ERR_KEY_STORE_FULL 0x1E09U /*Key storage in eHSM is full.*/
```

Definition at line 226 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.101 EHSM\_ERR\_KEY\_UPDATE\_ERROR

```
#define EHSM_ERR_KEY_UPDATE_ERROR 0x1E16U /*She key update error*/
```

Definition at line 240 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.102 EHSM\_ERR\_KMGR\_READ\_ERROR

```
#define EHSM_ERR_KMGR_READ_ERROR 0x1E1BU
```

ehsm kmgr module read fail

Definition at line 248 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.103 EHSM\_ERR\_MAILBOX\_SUCCESS

```
#define EHSM_ERR_MAILBOX_SUCCESS 0xA55AU
```

Error code definitions.

Except for values in [0x00U, 0xFFU], others values have two bytes. The first byte is used to distingwish modules. It has 16 values, where the higher 4-bit and low 4-bit are just in negation. The next byte is from [0, 255] as a specific error number.

0x00U is used for success. [0x0001U - 0x01FFU] for crypto lib basic error [0x0200U - 0x0EFFU] are reserved to be compatible with spec error code. [0x0F00U - 0x0FFFU] are used for some common errors. [0x1E00U - 0x1EFFU] are used for key management errors. [0x2D00U - 0x2DFFU] are used for crypto primitive errors. [0x2D00U - 0x2D1FU] SKE errors [0x2D20U - 0x2D3FU] PKE errors [0x2D40U - 0x2D5FU] HASH errors [0x2D60U - 0x2D7FU] RNG errors [0x2D80U - 0x2D8FU] CERT errors [0x2D90U - 0x2D9FU] SM9 errors [0x2DA0U - 0x2DFFU] Reserved or general error. [0x3C00U - 0x3CFFU] are used for secure boot, upgrade, debug auth errors. [0x3C00U - 0x3C1FU] Secure boot, upgrade errors [0x3C20U - 0x3C3FU] debug auth errors [0x3C40U - 0x3CFFU] Reserved or general error. [0x4B00U - 0x4BFFU] are used for system management, self test, .etc errors. [0x5A00U - 0x5AFFU] are used for counter, timer errors. [0x5A00U - 0x5A1FU] Counter errors [0x5A20U - 0x5A3FU] Timer errors [0x6900U - 0xF0FFU] are reserved.

#### Note

- 1) 0xA55AU is a specail value for mailbox success. 2) The value of an error number can be changed to keep a better order, if some errors are added or deleted. This is a specail value just for mailbox success.

Definition at line 54 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.104 EHSM\_ERR\_MEMORY\_FAILURE**

```
#define EHSM_ERR_MEMORY_FAILURE 0x1E1AU /*She key memory failure */
```

Definition at line 243 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.105 EHSM\_ERR\_MIDDLE\_SW**

```
#define EHSM_ERR_MIDDLE_SW 0x3C0FU
```

This is special value for success in secure boot, image upgrade module.

Definition at line 547 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.106 EHSM\_ERR\_NO\_CHALLENGE\_AVAILABLE**

```
#define EHSM_ERR_NO_CHALLENGE_AVAILABLE 0x3C23U
```

No challenge available for the authentication.

Definition at line 567 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.107 EHSM\_ERR\_NOT\_NIT**

```
#define EHSM_ERR_NOT_NIT 0x0F00U
```

The crypto driver is not initialized.

Definition at line 153 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.108 EHSM\_ERR\_NOT\_SUPPORT**

```
#define EHSM_ERR_NOT_SUPPORT 0x0F07U
```

The alg, mode, padding or required feature is not supported.

Definition at line 192 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.109 EHSM\_ERR\_OUT\_OF\_MEM**

```
#define EHSM_ERR_OUT_OF_MEM 0x0F01U
```

No free space to handle the request command.

Definition at line 158 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.110 EHSM\_ERR\_PARAM\_ERROR**

```
#define EHSM_ERR_PARAM_ERROR 0x0F06U
```

General parameter wrong.

Definition at line 187 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.111 EHSM\_ERR\_PKE\_ED25519\_MSG\_FLOW**

```
#define EHSM_ERR_PKE_ED25519_MSG_FLOW 0x2D26U
```

Message size is overflow for ed25519.

Definition at line 334 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.112 EHSM\_ERR\_PKE\_SIGN\_FAILED**

```
#define EHSM_ERR_PKE_SIGN_FAILED 0x2D24U
```

PKE sign failed.

Definition at line 324 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.113 EHSM\_ERR\_PKE\_VERIFY\_FAILED**

```
#define EHSM_ERR_PKE_VERIFY_FAILED 0x2D25U
```

PKE verify failed.

Definition at line 329 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.114 EHSM\_ERR\_PKE\_WORK\_ERROR**

```
#define EHSM_ERR_PKE_WORK_ERROR 0x2DFEU
```

PKE IP work err by crypto software.

**Note**

Not recommend this error code since we don't know what happens.

Definition at line 460 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.115 EHSM\_ERR\_PKE\_WRONG\_CURVE\_ID**

```
#define EHSM_ERR_PKE_WRONG_CURVE_ID 0x2D20U
```

Wrong curve\_id for ECIES or ECCP. Reserved some values for ske.

Definition at line 304 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.116 EHSM\_ERR\_PKE\_WRONG\_E\_SIZE**

```
#define EHSM_ERR_PKE_WRONG_E_SIZE 0x2D22U
```

Wrong e value size for RSA key generation.

Definition at line 314 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.117 EHSM\_ERR\_PKE\_WRONG\_KDF\_ALG**

```
#define EHSM_ERR_PKE_WRONG_KDF_ALG 0x2D27U
```

Wrong KDF algorithm for ECIES.

Definition at line 339 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.118 EHSM\_ERR\_PKE\_WRONG\_N\_SIZE**

```
#define EHSM_ERR_PKE_WRONG_N_SIZE 0x2D23U
```

Wrong n value size for RSA key generation.

Definition at line 319 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.119 EHSM\_ERR\_PKE\_WRONG\_RSA\_CERT\_MODE**

```
#define EHSM_ERR_PKE_WRONG_RSA_CERT_MODE 0x2D21U
```

Wrong CRT mode for RSA.

Definition at line 309 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.120 EHSM\_ERR\_QUEUE\_FULL**

```
#define EHSM_ERR_QUEUE_FULL 0x0F03U
```

The service request failed because the queue is full. Value is 0x05U.

Definition at line 170 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.121 EHSM\_ERR\_REG\_TWICE\_NOT\_MATCH**

```
#define EHSM_ERR_REG_TWICE_NOT_MATCH 0x3C0CU
```

The twice operation result does not match.

Definition at line 532 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.122 EHSM\_ERR\_REMOVE\_IMPOSSIBLE**

```
#define EHSM_ERR_REMOVE_IMPOSSIBLE 0x1E0EU /*Evita error: key is not allowed to become removed*/
```

Definition at line 231 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.123 EHSM\_ERR\_SKE\_IV\_SHOULD\_NOT\_NULL**

```
#define EHSM_ERR_SKE_IV_SHOULD_NOT_NULL 0x2D05U
```

Iv should not be null, but it's null.

Definition at line 278 of file eHSM\_Err\_Code\_Ip.h.



**4.14.1.124 EHSM\_ERR\_SKE\_MAC\_VRY\_FAILED**

```
#define EHSM_ERR_SKE_MAC_VRY_FAILED 0x2D09U
```

MAC verify failed.

Definition at line 298 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.125 EHSM\_ERR\_SKE\_WORK\_ERROR**

```
#define EHSM_ERR_SKE_WORK_ERROR 0x2DFDU
```

SKE IP work err by crypto software.

**Note**

Not recommend this error code since we don't know what happens.

Definition at line 453 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.126 EHSM\_ERR\_SKE\_WRONG\_AAD\_SIZE**

```
#define EHSM_ERR_SKE_WRONG_AAD_SIZE 0x2D08U
```

Wrong aad size (only fro GCM and CCM)

Definition at line 293 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.127 EHSM\_ERR\_SKE\_WRONG\_ALG**

```
#define EHSM_ERR_SKE_WRONG_ALG 0x2D00U
```

Wrong algorithm.

Definition at line 253 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.128 EHSM\_ERR\_SKE\_WRONG\_IV\_SIZE**

```
#define EHSM_ERR_SKE_WRONG_IV_SIZE 0x2D04U
```

Wrong iv size for the algorithm.

Definition at line 273 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.129 EHSM\_ERR\_SKE\_WRONG\_K\_SIZE**

```
#define EHSM_ERR_SKE_WRONG_K_SIZE 0x2D03U
```

Wrong key size for the algorithm.

Definition at line 268 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.130 EHSM\_ERR\_SKE\_WRONG\_L\_SIZE**

```
#define EHSM_ERR_SKE_WRONG_L_SIZE 0x2D07U
```

Wrong L size (only for CCM).

Definition at line 288 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.131 EHSM\_ERR\_SKE\_WRONG\_MODE**

```
#define EHSM_ERR_SKE_WRONG_MODE 0x2D01U
```

Wrong mode for the algorithm.

Definition at line 258 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.132 EHSM\_ERR\_SKE\_WRONG\_PADIDNG**

```
#define EHSM_ERR_SKE_WRONG_PADIDNG 0x2D02U
```

Wrong padding for the algorithm.

Definition at line 263 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.133 EHSM\_ERR\_SKE\_WRONG\_TAG\_SIZE**

```
#define EHSM_ERR_SKE_WRONG_TAG_SIZE 0x2D06U
```

Wrong tag size.

Definition at line 283 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.134 EHSM\_ERR\_SMALL\_BUFFER**

```
#define EHSM_ERR_SMALL_BUFFER 0x0F05U
```

The service request failed because the provided buffer is too small to store the result. Value is 0x03U.

Definition at line 182 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.135 EHSM\_ERR\_SW\_SUCCESS**

```
#define EHSM_ERR_SW_SUCCESS 0x00U
```

No error.

Definition at line 59 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.136 EHSM\_ERR\_TIME\_CHALLENGE\_EXPIRED**

```
#define EHSM_ERR_TIME_CHALLENGE_EXPIRED 0x5A28U
```

Time challenge expired.

Definition at line 642 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.137 EHSM\_ERR\_TIME\_STAMP\_EXPIRED**

```
#define EHSM_ERR_TIME_STAMP_EXPIRED 0x5A25U
```

Time stamp is expired.

Definition at line 627 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.138 EHSM\_ERR\_TIME\_STAMP\_VERIFY\_FAILED**

```
#define EHSM_ERR_TIME_STAMP_VERIFY_FAILED 0x5A24U
```

Time stamp verify failed.

Definition at line 622 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.139 EHSM\_ERR\_TRANSPORT\_IMPOSSIBLE**

```
#define EHSM_ERR_TRANSPORT_IMPOSSIBLE 0x1E0CU /*Give tansport key is not a capable transport key.*/
```

Definition at line 229 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.140 EHSM\_ERR\_TRNG\_BUFFER\_NULL**

```
#define EHSM_ERR_TRNG_BUFFER_NULL 0x2D60U
```

TRNG buffer is NULL.

Definition at line 374 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.141 EHSM\_ERR\_TRNG\_HT\_ERROR**

```
#define EHSM_ERR_TRNG_HT_ERROR 0x2D63U
```

TRNG HT error.

Definition at line 389 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.142 EHSM\_ERR\_TRNG\_INVALID\_CONFIG**

```
#define EHSM_ERR_TRNG_INVALID_CONFIG 0x2D62U
```

TRNG with invalid configuration.

Definition at line 384 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.143 EHSM\_ERR\_TRNG\_INVALID\_INPUT**

```
#define EHSM_ERR_TRNG_INVALID_INPUT 0x2D61U
```

TRNG input is invalid.

Definition at line 379 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.144 EHSM\_ERR\_TRNG\_TIMEOUT\_ERROR**

```
#define EHSM_ERR_TRNG_TIMEOUT_ERROR 0x2D64U
```

TRNG timeout while working.

Definition at line 394 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.145 EHSM\_ERR\_TRNG\_WORK\_ERROR**

```
#define EHSM_ERR_TRNG_WORK_ERROR 0x2DFCU
```

TRNG IP work general error by crypto software.

**Note**

Not recommend this error code since we don't know what happens.

Definition at line 446 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.146 EHSM\_ERR\_UTC\_SYNCHRONIZATION\_FAILED**

```
#define EHSM_ERR_UTC_SYNCHRONIZATION_FAILED 0x5A27U
```

UTC time synchronization failed.

Definition at line 637 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.147 EHSM\_ERR\_UTC\_TIMER\_INVALID\_INDEX**

```
#define EHSM_ERR_UTC_TIMER_INVALID_INDEX 0x5A23U
```

UTC timer index is invalid.

Definition at line 617 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.148 EHSM\_ERR\_UTC\_TIMER\_NOT\_SYNC**

```
#define EHSM_ERR_UTC_TIMER_NOT_SYNC 0x5A22U
```

UTC timer is not synchronized.

Definition at line 612 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.149 EHSM\_ERR\_WRITE\_PROTECTED**

```
#define EHSM_ERR_WRITE_PROTECTED 0x1E15U /*She key has been write protect.*/
```

Definition at line 239 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.150 EHSM\_ERR\_WRONG\_ALGORITHM**

```
#define EHSM_ERR_WRONG_ALGORITHM 0x3C02U
```

Given algorithm or algorithm mode not available.

Definition at line 482 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.151 EHSM\_ERR\_WRONG\_AUTHORIZATION**

```
#define EHSM_ERR_WRONG_AUTHORIZATION 0x1E11U /*Evita error: given athorization structure does not
fit key flags*/
```

Definition at line 234 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.152 EHSM\_ERR\_WRONG\_CERT\_KEY\_HANDLE**

```
#define EHSM_ERR_WRONG_CERT_KEY_HANDLE 0x1E0FU /*Evita error: given certification key handle is
unkown wrong*/
```

Definition at line 232 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.153 EHSM\_ERR\_WRONG\_CHALLENGE\_TYPE**

```
#define EHSM_ERR_WRONG_CHALLENGE_TYPE 0x3C20U
```

Wrong challenge type for getting challenge.

Definition at line 552 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.154 EHSM\_ERR\_WRONG\_CONTEXT**

```
#define EHSM_ERR_WRONG_CONTEXT 0x3C01U
```

Given context was wrong.

Definition at line 477 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.155 EHSM\_ERR\_WRONG\_DATA\_LENGTH**

```
#define EHSM_ERR_WRONG_DATA_LENGTH 0x3C06U
```

Wrong data length.

Definition at line 502 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.156 EHSM\_ERR\_WRONG\_DIRECT**

```
#define EHSM_ERR_WRONG_DIRECT 0x3C03U
```

Direction is wrong for cipher/hash/signature/verification.

Definition at line 487 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.157 EHSM\_ERR\_WRONG\_EHSM\_ADDR**

```
#define EHSM_ERR_WRONG_EHSM_ADDR 0x3C05U
```

Wrong eHSM address for otp writing or reading.

Definition at line 497 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.158 EHSM\_ERR\_WRONG\_JOB\_ID**

```
#define EHSM_ERR_WRONG_JOB_ID 0x0F09U
```

Wrong job id for cancel\_cmd.

Definition at line 202 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.159 EHSM\_ERR\_WRONG\_KEY\_COMBINATION**

```
#define EHSM_ERR_WRONG_KEY_COMBINATION 0x1E14U /*Keys do not fit the algorithm (e.g., RSA key vs.
ECDH)*/
```

Definition at line 237 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.160 EHSM\_ERR\_WRONG\_KEY\_DERIVE\_FUNC**

```
#define EHSM_ERR_WRONG_KEY_DERIVE_FUNC 0x1E06U /*Wrong key derivation function.*/
```

Definition at line 223 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.161 EHSM\_ERR\_WRONG\_KEY\_HANDLE**

```
#define EHSM_ERR_WRONG_KEY_HANDLE 0x1E01U /*Given key handle is wrong.*/
```

Definition at line 218 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.162 EHSM\_ERR\_WRONG\_KEY\_LEVEL**

```
#define EHSM_ERR_WRONG_KEY_LEVEL 0x1E0AU /*Wrong key level for bootrom.*/
```

Definition at line 227 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.163 EHSM\_ERR\_WRONG\_KEY\_LIFE\_LIMIT**

```
#define EHSM_ERR_WRONG_KEY_LIFE_LIMIT 0x1E03U /*Wrong key life limitation.*/
```

Definition at line 220 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.164 EHSM\_ERR\_WRONG\_KEY\_SIZE**

```
#define EHSM_ERR_WRONG_KEY_SIZE 0x1E05U /*Wrong key size.*/
```

Definition at line 222 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.165 EHSM\_ERR\_WRONG\_KEY\_TYPE**

```
#define EHSM_ERR_WRONG_KEY_TYPE 0x1E04U /*Wrong key type.*/
```

Definition at line 221 of file eHSM\_Err\_Code\_Ip.h.



**4.14.1.166 EHSM\_ERR\_WRONG\_KEY\_USAGE**

```
#define EHSM_ERR_WRONG_KEY_USAGE 0x1E00U
```

Wrong key usage.

Definition at line 217 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.167 EHSM\_ERR\_WRONG\_MODULE\_TYPE**

```
#define EHSM_ERR_WRONG_MODULE_TYPE 0x0F08U
```

Wrong module type for module\_status\_cmd.

Definition at line 197 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.168 EHSM\_ERR\_WRONG\_PADDING\_TYPE**

```
#define EHSM_ERR_WRONG_PADDING_TYPE 0x3C04U
```

Wrong padding type for cipher/mac/signature/verification.

Definition at line 492 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.169 EHSM\_ERR\_WRONG\_PROC\_MODE**

```
#define EHSM_ERR_WRONG_PROC_MODE 0x3C00U
```

Given process mode is wrong.

Definition at line 472 of file eHSM\_Err\_Code\_Ip.h.

**4.14.1.170 EHSM\_ERR\_WRONG\_PUB\_KEY**

```
#define EHSM_ERR_WRONG_PUB_KEY 0x1E0BU /*Wrong public key for debug authentication.*/
```

Definition at line 228 of file eHSM\_Err\_Code\_Ip.h.

#### 4.14.1.171 EHSM\_ERR\_WRONG\_REMOTE\_KEY\_HANDLE

```
#define EHSM_ERR_WRONG_REMOTE_KEY_HANDLE 0x1E13U /*Evita error: Given remote key handle is unknow
or wrong*/
```

Definition at line 236 of file eHSM\_Err\_Code\_lp.h.

#### 4.14.1.172 EHSM\_ERR\_WRONG\_SALT\_SIZE

```
#define EHSM_ERR_WRONG_SALT_SIZE 0x1E07U /*Wrong salt size for key derivation.*/
```

Definition at line 224 of file eHSM\_Err\_Code\_lp.h.

#### 4.14.1.173 EHSM\_ERR\_WRONG\_UTC\_TIME

```
#define EHSM_ERR_WRONG_UTC_TIME 0x5A21U
```

Wrong UTC time.

Definition at line 607 of file eHSM\_Err\_Code\_lp.h.

#### 4.14.1.174 EHSM\_ERR\_WRONG\_VERSION\_COUNTER

```
#define EHSM_ERR_WRONG_VERSION_COUNTER 0x3C08U
```

Wrong version counter.

Definition at line 512 of file eHSM\_Err\_Code\_lp.h.

## 4.15 eHSM\_Exclusive\_Area.h File Reference

### Functions

- void [Exclusive\\_area\\_enter](#) (void)  
*Fcuntion for critical area protection.*
- void [Exclusive\\_area\\_exit](#) (void)

### 4.15.1 Function Documentation

## 4.15.1.1 Exclusive\_area\_enter()

```
void Exclusive_area_enter (
 void)
```

Function for critical area protection.

## 4.15.1.2 Exclusive\_area\_exit()

```
void Exclusive_area_exit (
 void)
```

## 4.16 eHSM\_If\_Asr\_Cipher\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
#include "eHSM_If_Asr_ErrCode_Ip.h"
#include "eHSM_If_Asr_Job_Ip.h"
#include "eHSM_If_Asr_Key_Ip.h"
#include "eHSM_If_Asr_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_Mgr_Ctx_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Asr_Cipher_Ip.h"
```

## Macros

- `#define IV_BUFF_SIZE (32U)`

## Functions

- `Std_HsmReturnType ehsm_random_generate (Crypto_JobType *job)`  
*Fills the random generator service descriptor.*
- `Std_HsmReturnType ehsm_hash_hmac (Crypto_JobType *job)`  
*Fills the HASH and HMAC generation/verification service descriptor.*
- `Std_HsmReturnType ehsm_signature_gen_vry (Crypto_JobType *job)`  
*Fills the signature generation/verification service descriptor.*
- `Std_HsmReturnType ehsm_mac_request (Crypto_JobType *job)`  
*MAC service.*
- `Std_HsmReturnType ehsm_encrypt_request (Crypto_JobType *job)`  
*Encryption/Decryption service.*
- `Std_HsmReturnType ehsm_aead_request (Crypto_JobType *job)`  
*Aead Encryption/Decryption service.*

## 4.16.1 Macro Definition Documentation

### 4.16.1.1 IV\_BUFF\_SIZE

```
#define IV_BUFF_SIZE (32U)
```

Definition at line 31 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

## 4.16.2 Function Documentation

### 4.16.2.1 ehsm\_aead\_request()

```
Std_HsmReturnType ehsm_aead_request (
 Crypto_JobType * job)
```

Aead Encryption/Decryption service.

#### Parameters

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

#### Returns

Std\_HsmReturnType

#### Note

Definition at line 2282 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

### 4.16.2.2 ehsm\_encrypt\_request()

```
Std_HsmReturnType ehsm_encrypt_request (
 Crypto_JobType * job)
```

Encryption/Decryption service.

#### Parameters

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 2248 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.16.2.3 ehsm\_hash\_hmac()**

```
Std_HsmReturnType ehsm_hash_hmac (
 Crypto_JobType * job)
```

Fills the HASH and HMAC generation/verification service descriptor.

**Parameters**

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 1856 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.16.2.4 ehsm\_mac\_request()**

```
Std_HsmReturnType ehsm_mac_request (
 Crypto_JobType * job)
```

MAC service.

**Parameters**

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 2220 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.16.2.5 ehsm\_random\_generate()**

```
Std_HsmReturnType ehsm_random_generate (
 Crypto_JobType * job)
```

Fills the random generator service descriptor.

**Parameters**

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 1785 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.16.2.6 ehsm\_signature\_gen\_vry()**

```
Std_HsmReturnType ehsm_signature_gen_vry (
 Crypto_JobType * job)
```

Fills the signature genreation/verification service descriptor.

**Parameters**

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 2006 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

## 4.17 eHSM\_If\_Asr\_Cipher\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
```

### Functions

- [Std\\_HsmReturnType ehsm\\_random\\_generate \(Crypto\\_JobType \\*job\)](#)  
*Fills the random generator service descriptor.*
- [Std\\_HsmReturnType ehsm\\_hash\\_hmac \(Crypto\\_JobType \\*job\)](#)  
*Fills the HASH and HMAC generation/verification service descriptor.*
- [Std\\_HsmReturnType ehsm\\_mac\\_request \(Crypto\\_JobType \\*job\)](#)  
*MAC service.*
- [Std\\_HsmReturnType ehsm\\_encrypt\\_request \(Crypto\\_JobType \\*job\)](#)  
*Encryption/Decryption service.*
- [Std\\_HsmReturnType ehsm\\_aead\\_request \(Crypto\\_JobType \\*job\)](#)  
*Aead Encryption/Decryption service.*
- [Std\\_HsmReturnType ehsm\\_signature\\_gen\\_vry \(Crypto\\_JobType \\*job\)](#)  
*Fills the signature generation/verification service descriptor.*

### 4.17.1 Function Documentation

#### 4.17.1.1 ehsm\_aead\_request()

```
Std_HsmReturnType ehsm_aead_request (
 Crypto_JobType * job)
```

Aead Encryption/Decryption service.

#### Parameters

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

#### Returns

Std\_HsmReturnType

#### Note

Definition at line 2282 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

#### 4.17.1.2 ehsm\_encrypt\_request()

```
Std_HsmReturnType ehsm_encrypt_request (
 Crypto_JobType * job)
```

Encryption/Decryption service.

##### Parameters

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

##### Returns

Std\_HsmReturnType

##### Note

Definition at line 2248 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

#### 4.17.1.3 ehsm\_hash\_hmac()

```
Std_HsmReturnType ehsm_hash_hmac (
 Crypto_JobType * job)
```

Fills the HASH and HMAC generation/verification service descriptor.

##### Parameters

|    |     |  |
|----|-----|--|
| in | job |  |
|----|-----|--|

##### Returns

Std\_HsmReturnType

##### Note

Definition at line 1856 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

#### 4.17.1.4 ehsm\_mac\_request()

```
Std_HsmReturnType ehsm_mac_request (
 Crypto_JobType * job)
```

MAC service.



**Parameters**

|           |            |  |
|-----------|------------|--|
| <i>in</i> | <i>job</i> |  |
|-----------|------------|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 2220 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.17.1.5 ehsm\_random\_generate()**

```
Std_HsmReturnType ehsm_random_generate (
 Crypto_JobType * job)
```

Fills the random generator service descriptor.

**Parameters**

|           |            |  |
|-----------|------------|--|
| <i>in</i> | <i>job</i> |  |
|-----------|------------|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 1785 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.17.1.6 ehsm\_signature\_gen\_vry()**

```
Std_HsmReturnType ehsm_signature_gen_vry (
 Crypto_JobType * job)
```

Fills the signature genreation/verification service descriptor.

**Parameters**

|           |            |  |
|-----------|------------|--|
| <i>in</i> | <i>job</i> |  |
|-----------|------------|--|

**Returns**

Std\_HsmReturnType

**Note**

Definition at line 2006 of file eHSM\_If\_Asr\_Cipher\_Ip.c.

**4.18 eHSM\_If\_Asr\_ErrCode\_Ip.c File Reference**

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Asr_ErrCode_Ip.h"
#include "eHSM_Debug_Ip.h"
```

**Functions**

- [Std\\_HsmReturnType ehsm\\_autosar\\_convert\\_ret\\_code](#) (ehsm\_int32\_t ret)  
*translate a eHSM error code into AutoSAR error code*

**4.18.1 Function Documentation****4.18.1.1 ehsm\_autosar\_convert\_ret\_code()**

```
Std_HsmReturnType ehsm_autosar_convert_ret_code (
 ehsm_int32_t ret)
```

translate a eHSM error code into AutoSAR error code

**Parameters**

|           |            |                                                                                |
|-----------|------------|--------------------------------------------------------------------------------|
| <i>in</i> | <i>ret</i> | error code of eHSM, refers to error code in <a href="#">eHSM_Err_Code_Ip.h</a> |
|-----------|------------|--------------------------------------------------------------------------------|

**Returns**

AutoSAR error code, refers to error code in [eHSM\\_If\\_Asr\\_Types\\_Ip.h](#)

Definition at line 52 of file eHSM\_If\_Asr\_ErrCode\_Ip.c.

## 4.19 eHSM\_If\_Asr\_ErrCode\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
```

### Functions

- [Std\\_HsmReturnType ehsm\\_autosar\\_convert\\_ret\\_code](#) ([ehsm\\_int32\\_t](#) ret)  
*translate a eHSM error code into AutoSAR error code*

### 4.19.1 Function Documentation

#### 4.19.1.1 ehsm\_autosar\_convert\_ret\_code()

```
Std_HsmReturnType ehsm_autosar_convert_ret_code (
 ehsm_int32_t ret)
```

translate a eHSM error code into AutoSAR error code

#### Parameters

|                    |                     |                                                                                |
|--------------------|---------------------|--------------------------------------------------------------------------------|
| <a href="#">in</a> | <a href="#">ret</a> | error code of eHSM, refers to error code in <a href="#">eHSM_Err_Code_Ip.h</a> |
|--------------------|---------------------|--------------------------------------------------------------------------------|

#### Returns

AutoSAR error code, refers to error code in [eHSM\\_If\\_Asr\\_Types\\_Ip.h](#)

Definition at line 52 of file eHSM\_If\_Asr\_ErrCode\_Ip.c.

## 4.20 eHSM\_If\_Asr\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
```

### Functions

- [Std\\_HsmReturnType ehsm\\_init](#) (void)  
*Initializes the Crypto Driver.*
- [Std\\_HsmReturnType ehsm\\_get\\_version](#) ([Std\\_VersionInfoType](#) \*versioninfo)

Returns the version information of this module.

- `Std_HsmReturnType ehsm_job_submit (ehsm_uint32_t object_id, Crypto_JobType *job)`

Performs the crypto primitive, that is configured in the job parameter.

- `Std_HsmReturnType ehsm_job_cancel (ehsm_uint32_t object_id, Crypto_JobType *job)`

This interface removes the provided job from the queue and cancels the processing of the job if possible.

- `Std_HsmReturnType ehsm_key_element_set (ehsm_uint32_t crypto_key_id, ehsm_uint32_t key_element_id, const ehsm_uint8_t *key_ptr, ehsm_uint32_t key_length)`

Sets the given key element bytes to the key identified by cryptoKeyld.

- `Std_HsmReturnType ehsm_key_set_valid (ehsm_uint32_t crypto_key_id)`

Sets the key state of the key identified by cryptoKeyld to valid.

- `Std_HsmReturnType ehsm_key_element_get (ehsm_uint32_t crypto_key_id, ehsm_uint32_t key_element_id, ehsm_uint8_t *result_ptr, ehsm_uint32_t *result_length_ptr)`

This interface shall be used to get a key element of the key identified by the cryptoKeyld and store the key element in the memory location pointed by the result pointer.

- `Std_HsmReturnType ehsm_key_element_copy (ehsm_uint32_t crypto_key_id, ehsm_uint32_t key_element_id, ehsm_uint32_t target_crypto_key_id, ehsm_uint32_t target_key_element_id)`

Copies a key element to another key element in the same crypto driver. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation)

- `Std_HsmReturnType ehsm_key_element_copy_partial (ehsm_uint32_t crypto_key_id, ehsm_uint32_t key_element_id, ehsm_uint32_t key_element_source_offset, ehsm_uint32_t key_element_target_offset, ehsm_uint32_t key_element_copy_length, ehsm_uint32_t target_crypto_key_id, ehsm_uint32_t target_key_element_id)`

Copies a key element to another key element in the same crypto driver. The keyElementSourceOffset and keyElementCopyLength allows to copy just a part of the source key element into the destination. The offset of the target key is also specified with this function.

- `Std_HsmReturnType ehsm_key_copy (ehsm_uint32_t crypto_key_id, ehsm_uint32_t target_crypto_key_id)`

Copies a key with all its elements to another key in the same crypto driver. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation)

- `Std_HsmReturnType ehsm_key_element_ids_get (ehsm_uint32_t crypto_key_id, ehsm_uint32_t *key_element_ids_ptr, ehsm_uint32_t *key_element_ids_length)`

Used to retrieve information which key elements are available in a given key.

- `Std_HsmReturnType ehsm_key_exchange_calcsecret (ehsm_uint32_t crypto_key_id, ehsm_uint8_t *partner_public_value, ehsm_uint32_t partner_public_size, ehsm_uint8_t *shared_secret)`

Calculates the shared secret key for the key exchange with the key material of the key identified by the cryptoKeyld and the partner public key. The shared secret key is stored as a key element in the same key.

- `Std_HsmReturnType ehsm_key_generate (ehsm_uint32_t crypto_key_id)`

Generates new key material store it in the key identified by cryptoKeyld.

- `Std_HsmReturnType ehsm_key_derive (ehsm_uint32_t crypto_key_id, ehsm_uint32_t target_crypto_key_id)`

Derives a new key by using the key elements in the given key identified by the cryptoKeyld. The given key contains the key elements for the password, salt. The derived key is stored in the key element with the id 1 of the key identified by targetCryptoKeyld. The number of iterations is given in the key element CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS.

- `Std_HsmReturnType ehsm_key_exchange_calcpubval (ehsm_uint32_t crypto_key_id, ehsm_uint8_t *partner_public_value, ehsm_uint32_t *partner_public_size)`

Calculates the public value for the key exchange and stores the public key in the memory location pointed by the public value pointer.

- `Std_HsmReturnType ehsm_certificate_parse (ehsm_uint32_t cryptoKeyld)`

Parses the certificate data stored in the key element CRYPTO\_KE\_CERT\_DATA and fills the key elements CRYPTO\_KE\_CERT\_SIGNEDDATA, CRYPTO\_KE\_CERT\_PARSEDPUBLICKEY and CRYPTO\_KE\_CERT\_SIGNATURE.

- `Std_HsmReturnType ehsm_certificate_verify (ehsm_uint32_t cryptoKeyld, ehsm_uint32_t verifyCryptoKeyld, ehsm_uint32_t *verifyPtr)`

Verifies the certificate stored in the key referenced by cryptoValidateKeyld with the certificate stored in the key referenced by cryptoKeyld.

- `Std_HsmReturnType ehsm_remove_key_extend (ehsm_uint32_t crypto_key_id)`

Remove one key.

## 4.20.1 Function Documentation

## 4.20.1.1 ehsm\_certificate\_parse()

```
Std_HsmReturnType ehsm_certificate_parse (
 ehsm_uint32_t cryptoKeyId)
```

Parses the certificate data stored in the key element CRYPTO\_KE\_CERT\_DATA and fills the key elements CRYPTO\_KEY\_CERT\_SIGNEDDATA, CRYPTO\_KE\_CERT\_PARSEDPUBLICKEY and CRYPTO\_KE\_CERT\_SIGNATURE.

## Parameters

|    |                    |                                                        |
|----|--------------------|--------------------------------------------------------|
| in | <i>cryptoKeyld</i> | Holds the identifier of the key which shall be parsed. |
|----|--------------------|--------------------------------------------------------|

## Returns

Std\_HsmReturnType

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>E_OK</i>               | Request successful                                         |
| <i>E_NOT_OK</i>           | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>      | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_EMPTY</i> | Request failed because of uninitialized source key element |

## Note

## 4.20.1.2 ehsm\_certificate\_verify()

```
Std_HsmReturnType ehsm_certificate_verify (
 ehsm_uint32_t cryptoKeyId,
 ehsm_uint32_t verifyCryptoKeyId,
 ehsm_uint32_t * verifyPtr)
```

Verifies the certificate stored in the key referenced by cryptoValidateKeyId with the certificate stored in the key referenced by cryptoKeyId.

## Parameters

|     |                          |                                                                                                       |
|-----|--------------------------|-------------------------------------------------------------------------------------------------------|
| in  | <i>cryptoKeyId</i>       | Holds the identifier of the key which shall be used to validate the certificate.                      |
| in  | <i>verifyCryptoKeyld</i> | Holds the identifier of the key contain                                                               |
| out | <i>verifyPtr</i>         | Holds a pointer to the memory location which will contain the result of the certificate verification. |

## Returns

Std\_HsmReturnType

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>E_OK</i>               | Request successful                                         |
| <i>E_NOT_OK</i>           | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>      | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_EMPTY</i> | Request failed because of uninitialized source key element |

## Note

## 4.20.1.3 ehsm\_get\_version()

```
Std_HsmReturnType ehsm_get_version (
 Std_VersionInfoType * versioninfo)
```

Returns the version information of this module.

## Parameters

|    |                    |                                                                   |
|----|--------------------|-------------------------------------------------------------------|
| in | <i>versioninfo</i> | Pointer to where to store the version information of this module. |
|----|--------------------|-------------------------------------------------------------------|

## Returns

Std\_HsmReturnType

## Return values

|                 |                    |
|-----------------|--------------------|
| <i>E_OK</i>     | Request successful |
| <i>E_NOT_OK</i> | Request failed     |

## Note

## 4.20.1.4 ehsm\_init()

```
Std_HsmReturnType ehsm_init (
 void)
```

Initializes the Crypto Driver.

## Returns

Std\_HsmReturnType

## Return values

|                 |                    |
|-----------------|--------------------|
| <i>E_OK</i>     | Request successful |
| <i>E_NOT_OK</i> | Request failed     |

Definition at line 313 of file eHSM\_If\_Asr\_Job\_Ip.c.

## 4.20.1.5 ehsm\_job\_cancel()

```
Std_HsmReturnType ehsm_job_cancel (
 ehsm_uint32_t object_id,
 Crypto_JobType * job)
```

This interface removes the provided job from the queue and cancels the processing of the job if possible.

## Parameters

|    |                  |                                                                                                           |
|----|------------------|-----------------------------------------------------------------------------------------------------------|
| in | <i>object_id</i> | Holds the identifier of the Crypto Driver Object.                                                         |
| in | <i>job</i>       | Pointer to the configuration of the job. Contains structures with job and primitive relevant information. |

## Returns

Std\_HsmReturnType

## Return values

|                              |                                                                                                    |
|------------------------------|----------------------------------------------------------------------------------------------------|
| <i>E_OK</i>                  | Request successful                                                                                 |
| <i>E_NOT_OK</i>              | Request failed                                                                                     |
| <i>CRYPTO_E_JOB_CANCELED</i> | The job has been cancelled but is still processed. No results will be returned to the application. |

## Note

Definition at line 296 of file eHSM\_If\_Asr\_Job\_Ip.c.

## 4.20.1.6 ehsm\_job\_submit()

```
Std_HsmReturnType ehsm_job_submit (
 ehsm_uint32_t object_id,
 Crypto_JobType * job)
```

Performs the crypto primitive, that is configured in the job parameter.

## Parameters

|    |                        |                                                                                                                                              |
|----|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>object↔<br/>_id</i> | refers to the crypto object to do this job.                                                                                                  |
| in | <i>job</i>             | Pointer to the configuration of the job. Contains structures with job and primitive relevant information but also pointer to result buffers. |

## Returns

Std\_HsmReturnType

## Return values

|                                    |                                                                                                                                                            |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>E_OK</i>                        | Request successful                                                                                                                                         |
| <i>E_NOT_OK</i>                    | Request failed                                                                                                                                             |
| <i>CRYPTO_E_BUSY</i>               | Request failed, Crypro Driver Object is busy                                                                                                               |
| <i>CRYPTO_E_KEY_NOT_VALID</i>      | Request failed, the key is not valid                                                                                                                       |
| <i>CRYPTO_E_KEY_SIZE_MISMATCH</i>  | Request failed, a key element has the wrong size                                                                                                           |
| <i>CRYPTO_E_QUEUE_FULL</i>         | Request failed, the queue is full                                                                                                                          |
| <i>CRYPTO_E_KEY_READ_FAIL</i>      | The service request failed, because key element extraction is not allowed                                                                                  |
| <i>CRYPTO_E_KEY_WRITE_FAIL</i>     | The service request failed because the writing access failed                                                                                               |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i>  | The service request failed because the key is not available                                                                                                |
| <i>CRYPTO_E_ENTROPY_EXHAUSTION</i> | Request failed, the entropy is exhausted                                                                                                                   |
| <i>CRYPTO_E_SMALL_BUFFER</i>       | The provided buffer is too small to store the result                                                                                                       |
| <i>CRYPTO_E_JOB_CANCELED</i>       | The service request failed because the synchronous Job has been canceled<br>CRYPTO_E_KEY_EMPTY: Request failed because of uninitialized source key element |

## Note

Definition at line 170 of file eHSM\_If\_Asr\_Job\_Ip.c.

## 4.20.1.7 ehsm\_key\_copy()

```
Std_HsmReturnType ehsm_key_copy (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t target_crypto_key_id)
```

Copies a key with all its elements to another key in the same crypto driver. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation)

## Parameters

|    |                                   |                                                                                     |
|----|-----------------------------------|-------------------------------------------------------------------------------------|
| in | <i>crypto_key_id</i>              | Holds the identifier of the key whose key element shall be the source element.      |
| in | <i>target_crypto_key↔<br/>_id</i> | Holds the identifier of the key whose key element shall be the destination element. |



## Returns

Std\_HsmReturnType

## Return values

|                                   |                                                            |
|-----------------------------------|------------------------------------------------------------|
| <i>E_OK</i>                       | Request successful                                         |
| <i>E_NOT_OK</i>                   | Request failed                                             |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i> | Request failed, the requested key element is not available |
| <i>CRYPTO_E_KEY_READ_FAIL</i>     | Request failed, not allowed to extract key element         |
| <i>CRYPTO_E_KEY_WRITE_FAIL</i>    | Request failed, not allowed to write key element           |
| <i>CRYPTO_E_KEY_SIZE_MISMATCH</i> | Request failed, key element sizes are not compatible       |
| <i>CRYPTO_E_KEY_EMPTY</i>         | Request failed because of uninitialized source key element |

## Note

## 4.20.1.8 ehsm\_key\_derive()

```
Std_HsmReturnType ehsm_key_derive (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t target_crypto_key_id)
```

Derives a new key by using the key elements in the given key identified by the cryptoKeyld. The given key contains the key elements for the password, salt. The derived key is stored in the key element with the id 1 of the key identified by targetCryptoKeyld. The number of iterations is given in the key element CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS.

## Parameters

|    |                             |                                                                         |
|----|-----------------------------|-------------------------------------------------------------------------|
| in | <i>crypto_key_id</i>        | Holds the identifier of the key which is used for key derivation.       |
| in | <i>target_crypto_key_id</i> | Holds the identifier of the key which is used to store the derived key. |

## Returns

Std\_HsmReturnType

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>E_OK</i>               | Request successful                                         |
| <i>E_NOT_OK</i>           | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>      | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_EMPTY</i> | Request failed because of uninitialized source key element |

## Note

## 4.20.1.9 ehsm\_key\_element\_copy()

```
Std_HsmReturnType ehsm_key_element_copy (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t key_element_id,
 ehsm_uint32_t target_crypto_key_id,
 ehsm_uint32_t target_key_element_id)
```

Copies a key element to another key element in the same crypto driver. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation)

## Parameters

|    |                               |                                                                                                |
|----|-------------------------------|------------------------------------------------------------------------------------------------|
| in | <i>crypto_key_id</i>          | Holds the identifier of the key whose key element shall be the source element.                 |
| in | <i>key_element_id</i>         | Holds the identifier of the key element which shall be the source for the copy operation.      |
| in | <i>target_crypto_key_id</i>   | Holds the identifier of the key whose key element shall be the destination element.            |
| in | <i>target_key_element↵_id</i> | Holds the identifier of the key element which shall be the destination for the copy operation. |

## Returns

Std\_HsmReturnType

## Return values

|                                   |                                                            |
|-----------------------------------|------------------------------------------------------------|
| <i>E_OK</i>                       | Request successful                                         |
| <i>E_NOT_OK</i>                   | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>              | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i> | Request failed, the requested key element is not available |
| <i>CRYPTO_E_KEY_READ_FAIL</i>     | Request failed, not allowed to extract key element         |
| <i>CRYPTO_E_KEY_WRITE_FAIL</i>    | Request failed, not allowed to write key element           |
| <i>CRYPTO_E_KEY_SIZE_MISMATCH</i> | Request failed, key element sizes are not compatible       |
| <i>CRYPTO_E_KEY_EMPTY</i>         | Request failed because of uninitialized source key element |

## Note

## 4.20.1.10 ehsm\_key\_element\_copy\_partial()

```
Std_HsmReturnType ehsm_key_element_copy_partial (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t key_element_id,
 ehsm_uint32_t key_element_source_offset,
 ehsm_uint32_t key_element_target_offset,
 ehsm_uint32_t key_element_copy_length,
 ehsm_uint32_t target_crypto_key_id,
 ehsm_uint32_t target_key_element_id)
```

Copies a key element to another key element in the same crypto driver. The keyElementSourceOffset and keyElement↵CopyLength allows to copy just a part of the source key element into the destination. The offset of the target key is also specified with this function.

## Parameters

|    |                                  |                                                                                                       |
|----|----------------------------------|-------------------------------------------------------------------------------------------------------|
| in | <i>crypto_key_id</i>             | Holds the identifier of the key whose key element shall be the source element.                        |
| in | <i>key_element_id</i>            | Holds the identifier of the key element which shall be the source for the copy operation.             |
| in | <i>key_element_source_offset</i> | This is the offset of the of the source key element indicating the start index of the copy operation. |
| in | <i>key_element_target_offset</i> | This is the offset of the of the target key element indicating the start index of the copy operation. |
| in | <i>key_element_copy_length</i>   | Specifies the number of bytes that shall be copied.                                                   |
| in | <i>target_crypto_key_id</i>      | Holds the identifier of the key whose key element shall be the destination element.                   |
| in | <i>target_key_element_id</i>     | Holds the identifier of the key element which shall be the destination for the copy operation.        |

## Returns

Std\_HsmReturnType

## Return values

|                                   |                                                            |
|-----------------------------------|------------------------------------------------------------|
| <i>E_OK</i>                       | Request successful                                         |
| <i>E_NOT_OK</i>                   | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>              | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i> | Request failed, the requested key element is not available |
| <i>CRYPTO_E_KEY_READ_FAIL</i>     | Request failed, not allowed to extract key element         |
| <i>CRYPTO_E_KEY_WRITE_FAIL</i>    | Request failed, not allowed to write key element           |
| <i>CRYPTO_E_KEY_SIZE_MISMATCH</i> | Request failed, key element sizes are not compatible       |
| <i>CRYPTO_E_KEY_EMPTY</i>         | Request failed because of uninitialized source key element |

## Note

## 4.20.1.11 ehsm\_key\_element\_get()

```
Std_HsmReturnType ehsm_key_element_get (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t key_element_id,
 ehsm_uint8_t * result_ptr,
 ehsm_uint32_t * result_length_ptr)
```

This interface shall be used to get a key element of the key identified by the cryptoKeyId and store the key element in the memory location pointed by the result pointer.

## Parameters

|     |                       |                                                                      |
|-----|-----------------------|----------------------------------------------------------------------|
| in  | <i>crypto_key_id</i>  | Holds the identifier of the key whose key element shall be returned. |
| in  | <i>key_element_id</i> | Holds the identifier of the key element which shall be returned.     |
| out | <i>result_ptr</i>     | Holds the pointer of the buffer for the returned key element         |

## Parameters

|         |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in, out | <i>result_length_ptr</i> | Holds a pointer to a memory location in which the length information is stored. On calling this function this parameter shall contain the size of the buffer provided by resultPtr. If the key element is configured to allow partial access, this parameter contains the amount of data which should be read from the key element. The size may not be equal to the size of the provided buffer anymore. When the request has finished, the amount of data that has been stored shall be stored. |
|---------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Returns

Std\_HsmReturnType

## Return values

|                                   |                                                            |
|-----------------------------------|------------------------------------------------------------|
| <i>E_OK</i>                       | Request successful                                         |
| <i>E_NOT_OK</i>                   | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>              | Request failed, Crypro Driver Object is busy               |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i> | Request failed, the requested key element is not available |
| <i>CRYPTO_E_KEY_READ_FAIL</i>     | Request failed because read access was denied              |
| <i>CRYPTO_E_SMALL_BUFFER</i>      | The provided buffer is too small to store the result       |
| <i>CRYPTO_E_KEY_EMPTY</i>         | Request failed because of uninitialized source key element |

## Note

## 4.20.1.12 ehsm\_key\_element\_ids\_get()

```
Std_HsmReturnType ehsm_key_element_ids_get (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t * key_elementids_ptr,
 ehsm_uint32_t * key_elementids_length)
```

Used to retrieve information which key elements are available in a given key.

## Parameters

|     |                              |                                                                                                                                                                                                                                                                                                      |
|-----|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>crypto_key_id</i>         | Holds the identifier of the key whose available element ids shall be exported.                                                                                                                                                                                                                       |
| out | <i>key_elementids_ptr</i>    | Contains the pointer to the array where the ids of the key elements shall be stored.                                                                                                                                                                                                                 |
| in  | <i>key_elementids_length</i> | Holds a pointer to the memory location in which the number of key elements in the given key is stored. On calling this function, this parameter shall contain the size of the buffer provided by keyElementIdsPtr. When the request has finished, the actual number of key elements shall be stored. |

## Returns

Std\_HsmReturnType

## Return values

|                              |                                                      |
|------------------------------|------------------------------------------------------|
| <i>E_OK</i>                  | Request successful                                   |
| <i>E_NOT_OK</i>              | Request failed                                       |
| <i>CRYPTO_E_SMALL_BUFFER</i> | The provided buffer is too small to store the result |

## Note

## 4.20.1.13 ehsm\_key\_element\_set()

```
Std_HsmReturnType ehsm_key_element_set (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t key_element_id,
 const ehsm_uint8_t * key_ptr,
 ehsm_uint32_t key_length)
```

Sets the given key element bytes to the key identified by cryptoKeyld.

## Parameters

|    |                       |                                                                      |
|----|-----------------------|----------------------------------------------------------------------|
| in | <i>crypto_key_id</i>  | Holds the identifier of the key whose key element shall be set.      |
| in | <i>key_element_id</i> | Holds the identifier of the key element which shall be set.          |
| in | <i>key_ptr</i>        | Holds the pointer to the key data which shall be set as key element. |
| in | <i>key_length</i>     | Contains the length of the key element in bytes.                     |

## Returns

Std\_HsmReturnType

## Return values

|                                   |                                                                       |
|-----------------------------------|-----------------------------------------------------------------------|
| <i>E_OK</i>                       | Request successful                                                    |
| <i>E_NOT_OK</i>                   | Request failed                                                        |
| <i>CRYPTO_E_KEY_WRITE_FAIL</i>    | The service request failed because the writing access failed          |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i> | The service request failed because the key is not available           |
| <i>CRYPTO_E_KEY_SIZE_MISMATCH</i> | Request failed, key element size does not match size of provided data |

## Note

## 4.20.1.14 ehsm\_key\_exchange\_caclpubval()

```
Std_HsmReturnType ehsm_key_exchange_caclpubval (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint8_t * partner_pub_value,
 ehsm_uint32_t * partner_pub_size)
```

Calculates the public value for the key exchange and stores the public key in the memory location pointed by the public value pointer.

## Parameters

|         |                          |                                                                                                                                                                                                                                                                                                  |
|---------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in      | <i>crypto_key_id</i>     | Holds the identifier of the key which shall be used for the key exchange protocol.                                                                                                                                                                                                               |
| out     | <i>partner_pub_value</i> |                                                                                                                                                                                                                                                                                                  |
| in, out | <i>partner_pub_size</i>  | Holds a pointer to the memory location in which the public value length information is stored. On calling this function, this parameter shall contain the size of the buffer provided by publicValuePtr. When the request has finished, the actual length of the returned value shall be stored. |

## Returns

Std\_HsmReturnType

## Return values

|                              |                                                            |
|------------------------------|------------------------------------------------------------|
| <i>E_OK</i>                  | Request successful                                         |
| <i>E_NOT_OK</i>              | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>         | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_SMALL_BUFFER</i> | The provided buffer is too small to store the result       |
| <i>CRYPTO_E_KEY_EMPTY</i>    | Request failed because of uninitialized source key element |

## Note

## 4.20.1.15 ehsm\_key\_exchange\_calcsecret()

```
Std_HsmReturnType ehsm_key_exchange_calcsecret (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint8_t * partner_pub_value,
 ehsm_uint32_t partner_pub_size,
 ehsm_uint8_t * shared_secret)
```

Calculates the shared secret key for the key exchange with the key material of the key identified by the cryptoKeyId and the partner public key. The shared secret key is stored as a key element in the same key.

## Parameters

|    |                          |                                                                                     |
|----|--------------------------|-------------------------------------------------------------------------------------|
| in | <i>crypto_key_id</i>     | Holds the identifier of the key which shall be used for the key exchange protocol.  |
| in | <i>partner_pub_value</i> | Holds the pointer to the memory location which contains the partner's public value. |
| in | <i>partner_pub_size</i>  | Contains the length of the partner's public value in bytes.                         |

## Returns

Std\_HsmReturnType

## Return values

|                              |                                                            |
|------------------------------|------------------------------------------------------------|
| <i>E_OK</i>                  | Request successful                                         |
| <i>E_NOT_OK</i>              | Request failed                                             |
| <i>CRYPTO_E_SMALL_BUFFER</i> | The provided buffer is too small to store the result       |
| <i>CRYPTO_E_KEY_EMPTY</i>    | Request failed because of uninitialized source key element |

## Note

## 4.20.1.16 ehsm\_key\_generate()

```
Std_HsmReturnType ehsm_key_generate (
 ehsm_uint32_t crypto_key_id)
```

Generates new key material store it in the key identified by cryptoKeyId.

## Parameters

|    |                      |                                                                                  |
|----|----------------------|----------------------------------------------------------------------------------|
| in | <i>crypto_key_id</i> | Holds the identifier of the key which is to be updated with the generated value. |
|----|----------------------|----------------------------------------------------------------------------------|

## Returns

Std\_HsmReturnType

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>E_OK</i>               | Request successful                                         |
| <i>E_NOT_OK</i>           | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>      | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_EMPTY</i> | Request failed because of uninitialized source key element |

## Note

## 4.20.1.17 ehsm\_key\_set\_valid()

```
Std_HsmReturnType ehsm_key_set_valid (
 ehsm_uint32_t crypto_key_id)
```

Sets the key state of the key identified by cryptoKeyId to valid.

## Parameters

|    |                            |                                                              |
|----|----------------------------|--------------------------------------------------------------|
| in | <i>crypto_key↔<br/>_id</i> | Holds the identifier of the key which shall be set to valid. |
|----|----------------------------|--------------------------------------------------------------|

## Returns

Std\_HsmReturnType

## Return values

|                      |                                              |
|----------------------|----------------------------------------------|
| <i>E_OK</i>          | Request successful                           |
| <i>E_NOT_OK</i>      | Request failed                               |
| <i>CRYPTO_E_BUSY</i> | Request failed, Crypro Driver Object is busy |

## Note

## 4.20.1.18 ehsm\_remove\_key\_extend()

```
Std_HsmReturnType ehsm_remove_key_extend (
 ehsm_uint32_t crypto_key_id)
```

Remove one key.

## Parameters

|    |                            |                                                         |
|----|----------------------------|---------------------------------------------------------|
| in | <i>crypto_key↔<br/>_id</i> | Holds the identifier of the key which is to be removed. |
|----|----------------------------|---------------------------------------------------------|

## Returns

Std\_HsmReturnType

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>E_OK</i>               | Request successful                                         |
| <i>E_NOT_OK</i>           | Request failed                                             |
| <i>CRYPTO_E_BUSY</i>      | Request failed, Crypto Driver Object is busy               |
| <i>CRYPTO_E_KEY_EMPTY</i> | Request failed because of uninitialized source key element |

## Note



## 4.21 eHSM\_If\_Asr\_Job\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Asr_Job_Ip.h"
#include "eHSM_If_Asr_Ip.h"
#include "eHSM_If_Asr_Key_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
#include "eHSM_If_Asr_Cipher_Ip.h"
#include "eHSM_Debug_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Asr_ErrCode_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
```

### Functions

- [Std\\_HsmReturnType ehsm\\_job\\_submit](#) ([ehsm\\_uint32\\_t](#) object\_id, [Crypto\\_JobType](#) \*job)  
*Performs the crypto primitive, that is configured in the job parameter.*
- [Std\\_HsmReturnType ehsm\\_job\\_cancel](#) ([ehsm\\_uint32\\_t](#) object\_id, [Crypto\\_JobType](#) \*job)  
*This interface removes the provided job from the queue and cancels the processing of the job if possible.*
- [ehsm\\_uint32\\_t ehsm\\_init](#) (void)  
*Initializes the Crypto Driver.*

### 4.21.1 Function Documentation

#### 4.21.1.1 ehsm\_init()

```
ehsm_uint32_t ehsm_init (
 void)
```

Initializes the Crypto Driver.

#### Returns

[Std\\_HsmReturnType](#)

#### Return values

|                          |                    |
|--------------------------|--------------------|
| <a href="#">E_OK</a>     | Request successful |
| <a href="#">E_NOT_OK</a> | Request failed     |

Definition at line 313 of file eHSM\_If\_Asr\_Job\_Ip.c.

## 4.21.1.2 ehsm\_job\_cancel()

```
Std_HsmReturnType ehsm_job_cancel (
 ehsm_uint32_t object_id,
 Crypto_JobType * job)
```

This interface removes the provided job from the queue and cancels the processing of the job if possible.

## Parameters

|    |                  |                                                                                                           |
|----|------------------|-----------------------------------------------------------------------------------------------------------|
| in | <i>object_id</i> | Holds the identifier of the Crypto Driver Object.                                                         |
| in | <i>job</i>       | Pointer to the configuration of the job. Contains structures with job and primitive relevant information. |

## Returns

Std\_HsmReturnType

## Return values

|                              |                                                                                                    |
|------------------------------|----------------------------------------------------------------------------------------------------|
| <i>E_OK</i>                  | Request successful                                                                                 |
| <i>E_NOT_OK</i>              | Request failed                                                                                     |
| <i>CRYPTO_E_JOB_CANCELED</i> | The job has been cancelled but is still processed. No results will be returned to the application. |

## Note

Definition at line 296 of file eHSM\_If\_Asr\_Job\_Ip.c.

## 4.21.1.3 ehsm\_job\_submit()

```
Std_HsmReturnType ehsm_job_submit (
 ehsm_uint32_t object_id,
 Crypto_JobType * job)
```

Performs the crypto primitive, that is configured in the job parameter.

## Parameters

|    |                  |                                                                                                                                              |
|----|------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>object_id</i> | refers to the crypto object to do this job.                                                                                                  |
| in | <i>job</i>       | Pointer to the configuration of the job. Contains structures with job and primitive relevant information but also pointer to result buffers. |

## Returns

Std\_HsmReturnType

## Return values

|                                    |                                                                                                                                                            |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>E_OK</i>                        | Request successful                                                                                                                                         |
| <i>E_NOT_OK</i>                    | Request failed                                                                                                                                             |
| <i>CRYPTO_E_BUSY</i>               | Request failed, Crypro Driver Object is busy                                                                                                               |
| <i>CRYPTO_E_KEY_NOT_VALID</i>      | Request failed, the key is not valid                                                                                                                       |
| <i>CRYPTO_E_KEY_SIZE_MISMATCH</i>  | Request failed, a key element has the wrong size                                                                                                           |
| <i>CRYPTO_E_QUEUE_FULL</i>         | Request failed, the queue is full                                                                                                                          |
| <i>CRYPTO_E_KEY_READ_FAIL</i>      | The service request failed, because key element extraction is not allowed                                                                                  |
| <i>CRYPTO_E_KEY_WRITE_FAIL</i>     | The service request failed because the writing access failed                                                                                               |
| <i>CRYPTO_E_KEY_NOT_AVAILABLE</i>  | The service request failed because the key is not available                                                                                                |
| <i>CRYPTO_E_ENTROPY_EXHAUSTION</i> | Request failed, the entropy is exhausted                                                                                                                   |
| <i>CRYPTO_E_SMALL_BUFFER</i>       | The provided buffer is too small to store the result                                                                                                       |
| <i>CRYPTO_E_JOB_CANCELED</i>       | The service request failed because the synchronous Job has been canceled<br>CRYPTO_E_KEY_EMPTY: Request failed because of uninitialized source key element |

## Note

Definition at line 170 of file eHSM\_If\_Asr\_Job\_Ip.c.

## 4.22 eHSM\_If\_Asr\_Job\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
```

### Macros

- #define [Crypto\\_GetJobKey](#)(pJob) (pJob->cryptoKeyId)
- #define [Crypto\\_GetKeyIdx](#)(keyId) (((keyId) < CRYPTO\_MAX\_KEY\_ID) ? (keyId) : CRYPTO\_INVALID\_KEY\_ID)

### 4.22.1 Macro Definition Documentation

#### 4.22.1.1 [Crypto\\_GetJobKey](#)

```
#define Crypto_GetJobKey(
 pJob) (pJob->cryptoKeyId)
```

Definition at line 24 of file eHSM\_If\_Asr\_Job\_Ip.h.

## 4.22.1.2 Crypto\_GetKeyIdx

```
#define Crypto_GetKeyIdx(
 keyId) (((keyId) < CRYPTO_MAX_KEY_ID) ? (keyId) : CRYPTO_INVALID_KEY_ID)
```

Definition at line 25 of file eHSM\_If\_Asr\_Job\_Ip.h.

## 4.23 eHSM\_If\_Asr\_Key\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include <stdio.h>
#include <string.h>
```

## 4.24 eHSM\_If\_Asr\_Key\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
```

## Functions

- Std\_HsmReturnType ehsm\_keygen\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_keyderi\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_calcpubval\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_calcsecret\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_keysetvalid\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_keyremove\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_keyimport\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_keyexport\_with\_job (Crypto\_JobType \*job)
- Std\_HsmReturnType ehsm\_key\_is\_valid (ehsm\_uint32\_t crypto\_key\_id)
- ehsm\_int32\_t ehsm\_key\_element\_type\_get\_ex (ehsm\_uint32\_t crypto\_key\_id, ehsm\_uint32\_t key\_element\_id, CryptoKeyElementType \*\*element)  
*get the key element of a crypto key [in] crypto key id [in] crypto element id [out] pointer of \*element*
- void ehsm\_key\_mgr\_init (void)
- ehsm\_key\_type\_e ehsm\_get\_ehsm\_key\_type (ehsm\_uint32\_t crypto\_key\_id)

## 4.24.1 Function Documentation

## 4.24.1.1 ehsm\_calcpubval\_with\_job()

```
Std_HsmReturnType ehsm_calcpubval_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.2 ehsm\_calcsecret\_with\_job()

```
Std_HsmReturnType ehsm_calcsecret_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.3 ehsm\_get\_ehsm\_key\_type()

```
ehsm_key_type_e ehsm_get_ehsm_key_type (
 ehsm_uint32_t crypto_key_id)
```

#### 4.24.1.4 ehsm\_key\_element\_type\_get\_ex()

```
ehsm_int32_t ehsm_key_element_type_get_ex (
 ehsm_uint32_t crypto_key_id,
 ehsm_uint32_t key_element_id,
 CryptoKeyElementType ** element)
```

get the key element of a crypto key [in] crypto key id [in] crypto element id [out] pointer of \*element

#### Returns

0 for success, negative values for error.

#### 4.24.1.5 ehsm\_key\_is\_valid()

```
Std_HsmReturnType ehsm_key_is_valid (
 ehsm_uint32_t crypto_key_id)
```

#### 4.24.1.6 ehsm\_key\_mgr\_init()

```
void ehsm_key_mgr_init (
 void)
```

#### 4.24.1.7 ehsm\_keyderi\_with\_job()

```
Std_HsmReturnType ehsm_keyderi_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.8 ehsm\_keyexport\_with\_job()

```
Std_HsmReturnType ehsm_keyexport_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.9 ehsm\_keygen\_with\_job()

```
Std_HsmReturnType ehsm_keygen_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.10 ehsm\_keyimport\_with\_job()

```
Std_HsmReturnType ehsm_keyimport_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.11 ehsm\_keyremove\_with\_job()

```
Std_HsmReturnType ehsm_keyremove_with_job (
 Crypto_JobType * job)
```

#### 4.24.1.12 ehsm\_keysetvalid\_with\_job()

```
Std_HsmReturnType ehsm_keysetvalid_with_job (
 Crypto_JobType * job)
```

## 4.25 eHSM\_If\_Asr\_KeyCfg\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
#include "eHSM_If_Asr_Ip.h"
#include "eHSM_Debug_Ip.h"
#include "Asr_Standard_Types.h"
```

### Functions

- `crypto_key_element_type_info_st * ehsm_get_crypto_ke_info (ehsm_uint32_t crypto_key_type, ehsm_uint32_t *size)`
- `void ehsm_printf_ke_size (void)`
- `CryptoDriverObject * ehsm_get_crypto_driver_object (void)`

## Variables

- [crypto\\_key\\_element\\_type\\_info\\_st she\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st she\\_plain\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st mac\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st cipher\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st exchange\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st signature\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st derive\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st generate\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st import\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st export\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st key\\_status\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st key\\_remove\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st copy\\_key\\_elements \[\]](#)
- [crypto\\_key\\_element\\_type\\_info\\_st certificate\\_key\\_elements \[\]](#)
- [CryptoKey crypto\\_keys \[CRYPTO\\_MAX\\_KEY\\_NUM\]](#)
- *configuration for crypto key type*
- [CryptoPrimitive Ske\\_CryptoPrimitives \[\]](#)
- [CryptoPrimitive Pke\\_CryptoPrimitives \[\]](#)
- [CryptoPrimitive Trng\\_CryptoPrimitives \[\]](#)
- [CryptoPrimitive Hash\\_CryptoPrimitives \[\]](#)
- [CryptoPrimitive Key\\_CryptoPrimitives \[\]](#)
- [CryptoDriverObject CryptoDriverObjects \[CRYPTO\\_OBJECT\\_TYPE\\_MAX\]](#)

## 4.25.1 Function Documentation

### 4.25.1.1 ehsm\_get\_crypto\_driver\_object()

```
CryptoDriverObject* ehsm_get_crypto_driver_object (
 void) [inline]
```

Definition at line 1805 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

### 4.25.1.2 ehsm\_get\_crypto\_ke\_info()

```
crypto_key_element_type_info_st* ehsm_get_crypto_ke_info (
 ehsm_uint32_t crypto_key_type,
 ehsm_uint32_t * size)
```

Definition at line 707 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

### 4.25.1.3 ehsm\_printf\_ke\_size()

```
void ehsm_printf_ke_size (
 void)
```

Definition at line 773 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2 Variable Documentation

### 4.25.2.1 certificate\_key\_elements

`crypto_key_element_type_info_st` certificate\_key\_elements[ ]

Definition at line 606 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

### 4.25.2.2 cipher\_key\_elements

`crypto_key_element_type_info_st` cipher\_key\_elements[ ]

Definition at line 152 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

### 4.25.2.3 copy\_key\_elements

`crypto_key_element_type_info_st` copy\_key\_elements[ ]

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KE_COPY_KEY_PARENT_KEY,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_PERSIST,
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_COPY_KEY_PARENT_KEY_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_PARENT_K_WRITE_ACCESS
 },
 {
 .CryptoKeyElementId = CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_PARTIAL_ACCESS
 ,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_PERSIST,
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_READ_ACCESS
 ,
 .CryptoKeyElementMaxSize = CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE_SIZE
 ,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_COPY_K_TARGET_K_HANDLE_WRITE_ACCESS
 }
}
```

Definition at line 584 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.



## 4.25.2.4 crypto\_keys

```
CryptoKey crypto_keys[CRYPTO_MAX_KEY_NUM]
```

configuration for crypto key type

Definition at line 649 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.5 CryptoDriverObjects

```
CryptoDriverObject CryptoDriverObjects[CRYPTO_OBJECT_TYPE_MAX]
```

Definition at line 1753 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.6 derive\_key\_elements

```
crypto_key_element_type_info_st derive_key_elements[]
```

Definition at line 398 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.7 exchange\_key\_elements

```
crypto_key_element_type_info_st exchange_key_elements[]
```

Definition at line 246 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.8 export\_key\_elements

```
crypto_key_element_type_info_st export_key_elements[]
```

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KE_EXPORT_KEY,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_EXPORT_KEY_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_WRITE_ACCESS
 },
 {
 .CryptoKeyElementId = CRYPTO_KE_EXPORT_KEY_BLOB,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_EXPORT_KEY_BLOB_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_EXPORT_K_BLOB_WRITE_ACCESS
 }
}
```

Definition at line 536 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.9 generate\_key\_elements

```
crypto_key_element_type_info_st generate_key_elements[]
```

Definition at line 465 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.10 Hash\_CryptoPrimitives

```
CryptoPrimitive Hash_CryptoPrimitives[]
```

Definition at line 1341 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.11 import\_key\_elements

```
crypto_key_element_type_info_st import_key_elements[]
```

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KE_IMPORT_KEY,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_IMPORT_KEY_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_IMPORT_K_WRITE_ACCESS
 },
 {
 .CryptoKeyElementId = CRYPTO_KE_IMPORTED_KEY_KEYHANDLE,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_PARTIAL_ACCESS
 },
 {
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist =
 CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_PERSIST,
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_IMPORTED_KEY_KEYHANDLE_SIZE
 },
 {
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_IMPORTED_K_KHANDLE_WRITE_ACCESS
 }
}
```

Definition at line 514 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.12 Key\_CryptoPrimitives

```
CryptoPrimitive Key_CryptoPrimitives[]
```

Definition at line 1688 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

#### 4.25.2.13 key\_remove\_elements

`crypto_key_element_type_info_st` key\_remove\_elements[ ]

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KEY_REMOVE_KEY,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KEY_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_PERSIST
 },
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KEY_REMOVE_KEY_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_REMOVE_K_WRITE_ACCESS
}
```

Definition at line 571 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

#### 4.25.2.14 key\_status\_elements

`crypto_key_element_type_info_st` key\_status\_elements[ ]

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KEY_KEY_STATUS,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_K_STATUS_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KEY_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_K_STATUS_PERSIST
 },
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_K_STATUS_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KEY_KEY_STATUS_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_K_STATUS_WRITE_ACCESS
}
```

Definition at line 558 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.



## 4.25.2.18 she\_plain\_key\_elements

```
crypto_key_element_type_info_st she_plain_key_elements[]
```

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KE_KEY_MATERIAL,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_SHEKEYS,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_KEY_MATERIAL_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_K_MATERIAL_WRITE_ACCESS
 },
 {
 .CryptoKeyElementId = CRYPTO_KE_CIPHER_IV,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_CIPHER_IV_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_WRITE_ACCESS
 },
 {
 .CryptoKeyElementId = CRYPTO_KE_KEYGENERATE_SEED,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_KEYGENERATE_SEED_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_KGENERATE_SEED_WRITE_ACCESS
 }
}
```

Definition at line 99 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.25.2.19 signature\_key\_elements

```
crypto_key_element_type_info_st signature_key_elements[]
```

**Initial value:**

```
=
{
 {
 .CryptoKeyElementId = CRYPTO_KE_SIGNATURE_KEY,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PARTIAL_ACCESS,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist = CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_PERSIST
 },
 {
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_READ_ACCESS,
 .CryptoKeyElementMaxSize = CRYPTO_KE_SIGNATURE_KEY_SIZE,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_K_WRITE_ACCESS
 },
 {
 .CryptoKeyElementId = CRYPTO_KE_SIGNATURE_TIMESTAMPED,
```

```

 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PARTIAL_ACCESS
 ,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_PERSIST,
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_READ_ACCESS
 ,
 .CryptoKeyElementMaxSize = CRYPTO_KE_SIGNATURE_TIMESTAMPED_SIZE
 ,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_TIMESTAMPED_WRITE_ACCESS
},
{
 .CryptoKeyElementId = CRYPTO_KE_SIGNATURE_RSA_CRT_MODE,
 .CryptoKeyElementAllowPartialAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_PARTIAL_ACCESS
 ,
 .CryptoKeyFormat = CRYPTO_KE_FORMAT_BIN_OCTET,
 .CryptoKeyElementPersist =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_PERSIST,
 .CryptoKeyElementReadAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_READ_ACCESS
 ,
 .CryptoKeyElementMaxSize = CRYPTO_KE_SIGNATURE_RSA_CRT_MODE_SIZE
 ,
 .CryptoKeyElementWriteAccess =
 CONFIG_EHSM_KMGR_V_ASR_SIGNATURE_RSA_CRT_MODE_WRITE_ACCESS
}
}

```

Definition at line 367 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

#### 4.25.2.20 Ske\_CryptoPrimitives

```
CryptoPrimitive Ske_CryptoPrimitives[]
```

Definition at line 804 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

#### 4.25.2.21 Trng\_CryptoPrimitives

```
CryptoPrimitive Trng_CryptoPrimitives[]
```

**Initial value:**

```

=
{
 {
 .CryptoPrimitiveAlgorithmFamily = CRYPTO_ALGOFAM_RNG,
 .CryptoPrimitiveAlgorithmMode = CRYPTO_ALGOMODE_CTRDRBG,
 .CryptoPrimitiveAlgorithmSecondaryFamily = CRYPTO_ALGOFAM_NOT_SET,
 .CryptoPrimitiveService = CRYPTO_RANDOMGENERATE,
 },
 {
 .CryptoPrimitiveAlgorithmFamily = CRYPTO_ALGOFAM_AES,
 .CryptoPrimitiveAlgorithmMode = CRYPTO_ALGOMODE_CTRDRBG,
 .CryptoPrimitiveAlgorithmSecondaryFamily = CRYPTO_ALGOFAM_NOT_SET,
 .CryptoPrimitiveService = CRYPTO_RANDOMGENERATE,
 },
}

```

Definition at line 1323 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.26 eHSM\_If\_Asr\_KeyCfg\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Asr_Types_Ip.h"
```

### Macros

- #define CRYPTO\_MAX\_KEY\_NUM 18
- #define CRYPTO\_KEY1\_SHE\_KEY 0
- #define CRYPTO\_KEY2\_SHE\_PLAIN\_KEY 1
- #define CRYPTO\_KEY3\_EVITA\_SIGNATURE\_KEY 2
- #define CRYPTO\_KEY4\_EVITA\_CIPHER\_KEY 3
- #define CRYPTO\_KEY5\_EVITA\_EXCHANGE\_KEY 4
- #define CRYPTO\_KEY6\_EVITA\_EXCHANGE\_KEY 5
- #define CRYPTO\_KEY7\_EVITA\_DERIVE\_KEY 6
- #define CRYPTO\_KEY8\_EVITA\_DERIVE\_KEY 7
- #define CRYPTO\_KEY9\_EVITA\_GENERATE\_KEY 8
- #define CRYPTO\_KEY10\_EVITA\_IMPORT\_KEY 9
- #define CRYPTO\_KEY11\_EVITA\_EXPORT\_KEY 10
- #define CRYPTO\_KEY12\_EVITA\_REMOVE\_KEY 11
- #define CRYPTO\_KEY13\_EVITA\_KEY\_STATUS 12
- #define CRYPTO\_KEY14\_EVITA\_KEY\_COPY\_KEY 13
- #define CRYPTO\_KEY15\_EVITA\_KEY\_COPY\_KEY 14
- #define CRYPTO\_KEY16\_CERTIFICATE\_KEY 15
- #define CRYPTO\_KEY17\_CERTIFICATE\_KEY 16
- #define CRYPTO\_KEY18\_MAC\_KEY 17
- #define CRYPTO\_INVALID\_KEY\_ID 0xFFFFFFFFU
- #define CRYPTO\_SHE\_KEY\_NUM 1U
- #define CRYPTO\_SHE\_PLAIN\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_SIGNATURE\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_CIPHER\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_EXCHANGE\_KEY\_NUM 2U
- #define CRYPTO\_EVITA\_DERIVE\_KEY\_NUM 2U
- #define CRYPTO\_EVITA\_GENERATE\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_IMPORT\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_EXPORT\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_REMOVE\_KEY\_NUM 1U
- #define CRYPTO\_EVITA\_KEY\_STATUS\_NUM 1U
- #define CRYPTO\_EVITA\_KEY\_COPY\_KEY\_NUM 2U
- #define CRYPTO\_CERTIFICATE\_KEY\_NUM 2U
- #define CRYPTO\_MAC\_KEY\_NUM 1U
- #define CRYPTO\_KE\_MAC\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_MAC\_PROOF\_SIZE CRYPTO\_M4\_SIZE\_U32
- #define CRYPTO\_KE\_SIGNATURE\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_RANDOM\_SEED\_STATE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_RANDOM\_ALGORITHM\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_CIPHER\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_CIPHER\_IV\_SIZE ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_KE\_CIPHER\_PROOF\_SIZE CRYPTO\_M5\_SIZE\_U32
- #define CRYPTO\_KE\_CIPHER\_CURVE\_ID\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_CIPHER\_CIPHER\_ALG\_SIZE ((ehsm\_uint32\_t)4U)

- #define CRYPTO\_KE\_CIPHER\_KDF\_ALG\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_CIPHER\_MAC\_ALG\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_CIPHER\_MAC\_SIZE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_AEAD\_TAG\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_CIPHER\_2NDKEY\_SIZE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_BASE\_SIZE ((ehsm\_uint32\_t)8U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_KEYEXCHANGE\_OWNPUKEY\_SIZE ((ehsm\_uint32\_t)64U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_PEERPUKEY\_SIZE ((ehsm\_uint32\_t)512U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_create\_evita\_key\_info\_↵  
st))
- #define CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINFO\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_evita\_↵  
key\_info\_st))
- #define CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK\_SIZE ((ehsm\_uint32\_t)sizeof(ehsm\_export\_pub\_↵  
key\_st))
- #define CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE\_SIZE ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE\_SIZE ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYDERIVATION\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_create\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_KEYDERIVATION\_PASSWD\_SIZE ((ehsm\_uint32\_t)64U)
- #define CRYPTO\_KE\_KEYDERIVATION\_SALT\_SIZE ((ehsm\_uint32\_t)64U)
- #define CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYDERIVATION\_TYPE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYGENERATE\_KEYINFO\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_create\_evita\_key\_info\_↵  
st))
- #define CRYPTO\_KE\_KEYGENERATE\_SEED\_SIZE ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_KE\_KEYGENERATE\_ALGORITHM\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYGENERATE\_KEY\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_copy\_key\_dh\_key\_↵  
info\_st))
- #define CRYPTO\_KE\_IMPORT\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_import\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_EXPORT\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_key\_export\_info\_st))
- #define CRYPTO\_KE\_EXPORT\_KEY\_BLOB\_SIZE ((ehsm\_uint32\_t)1024U)
- #define CRYPTO\_KE\_SHE\_PLAIN\_KEY\_SIZE ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_KE\_KEY\_MATERIAL\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_EXT\_SHE\_KEY\_SIZE ((ehsm\_uint32\_t)4U)
- #define CRYPTO\_KE\_REMOVE\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_evita\_key\_info\_st))
- #define CRYPTO\_KE\_KEY\_STATUS\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_key\_status\_info\_st))
- #define CRYPTO\_KE\_KEY\_STATUS\_BLOB\_SIZE ((ehsm\_uint32\_t)1024U)
- #define CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY\_SIZE ((ehsm\_uint32\_t)sizeof(crypto\_copy\_key\_info\_st))
- #define CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE\_SIZE ((ehsm\_uint32\_t)4)
- #define CRYPTO\_SHE\_KEY\_ELEMENT\_SIZE
- #define CRYPTO\_SHE\_PLAIN\_KEY\_ELEMENT\_SIZE
- #define CRYPTO\_EVITA\_CIPHER\_KEY\_ELEMENT\_SIZE
- #define CRYPTO\_EVITA\_EXCHANGE\_KEY\_ELEMENT\_SIZE
- #define CRYPTO\_EVITA\_DERIVE\_KEY\_ELEMENT\_SIZE
- #define CRYPTO\_EVITA\_GENERATE\_KEY\_ELEMENT\_SIZE
- #define CRYPTO\_EVITA\_IMPORT\_KEY\_ELEMENT\_SIZE (CRYPTO\_KE\_IMPORT\_KEY\_SIZE + CRYPTO\_KE\_↵  
IMPORTED\_KEY\_KEYHANDLE\_SIZE)
- #define CRYPTO\_EVITA\_EXPORT\_KEY\_ELEMENT\_SIZE (CRYPTO\_KE\_EXPORT\_KEY\_SIZE + CRYPTO\_K\_↵  
E\_EXPORT\_KEY\_BLOB\_SIZE)



- `#define CRYPTO_EVITA_REMOVE_KEY_ELEMENT_SIZE CRYPTO_KE_REMOVE_KEY_SIZE`
- `#define CRYPTO_EVITA_KEY_STATUS_ELEMENT_SIZE (CRYPTO_KE_KEY_STATUS_SIZE + CRYPTO_KE_↵_KEY_STATUS_BLOB_SIZE)`
- `#define CRYPTO_EVITA_SIGNATURE_KEY_ELEMENT_SIZE (CRYPTO_KE_SIGNATURE_KEY_SIZE + CRY↵PTO_KE_SIGNATURE_TIMESTAMPED_SIZE + CRYPTO_KE_SIGNATURE_RSA_CERT_MODE_SIZE)`
- `#define CRYPTO_EVITA_COPY_KEY_ELEMENT_SIZE (CRYPTO_KE_COPY_KEY_PARENT_KEY_SIZE + C↵RYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE_SIZE)`
- `#define CRYPTO_EVITA_MAC_KEY_ELEMENT_SIZE (CRYPTO_KE_MAC_KEY_SIZE + CRYPTO_KE_KEYG↵ENERATE_SEED_SIZE)`
- `#define CRYPTO_EVITA_CERTIFICATE_KEY_ELEMENT_SIZE (2048U)`
- `#define KEY_ELEMENT_RESERVER 8`
- `#define KEY_ELEMENT_NUM`
- `#define KEY_ELEMENT_VALUE_BUFFER_RESERVER 256`
- `#define KEY_ELEMENT_VALUE_BUFFER_SIZE`

## Enumerations

- `enum crypto_object_type_e {`  
`CRYPTO_OBJECT_TYPE_SKE = 0, CRYPTO_OBJECT_TYPE_PKE, CRYPTO_OBJECT_TYPE_TRNG, CRY↵`  
`PTO_OBJECT_TYPE_HASH,`  
`CRYPTO_OBJECT_TYPE_KEY, CRYPTO_OBJECT_TYPE_SYSMGR, CRYPTO_OBJECT_TYPE_MAX }`

## Functions

- `crypto_key_element_type_info_st * ehsm_get_crypto_ke_info (ehsm_uint32_t crypto_key_type, ehsm_uint32_↵t *size)`
- `CryptoDriverObject * ehsm_get_crypto_driver_object (void)`
- `void ehsm_printf_ke_size (void)`

### 4.26.1 Macro Definition Documentation

#### 4.26.1.1 CRYPTO\_CERTIFICATE\_KEY\_NUM

```
#define CRYPTO_CERTIFICATE_KEY_NUM 2U
```

Definition at line 105 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.2 CRYPTO\_EVITA\_CERTIFICATE\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_CERTIFICATE_KEY_ELEMENT_SIZE (2048U)
```

Definition at line 244 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.3 CRYPTO\_EVITA\_CIPHER\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_CIPHER_KEY_ELEMENT_SIZE
```

**Value:**

```
(CRYPTO_KE_CIPHER_KEY_SIZE + CRYPTO_KE_CIPHER_IV_SIZE + \
 CRYPTO_KE_CIPHER_CURVE_ID_SIZE + \
 CRYPTO_KE_CIPHER_KDF_ALG_SIZE + \
 CRYPTO_KE_CIPHER_2NDKEY_SIZE_SIZE + \
 CRYPTO_KE_AEAD_TAG_SIZE)
 CRYPTO_KE_CIPHER_PROOF_SIZE +
 CRYPTO_KE_CIPHER_CIPHER_ALG_SIZE +
 CRYPTO_KE_CIPHER_MAC_ALG_SIZE +
 CRYPTO_KE_CIPHER_MAC_SIZE_SIZE +
```

Definition at line 208 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.4 CRYPTO\_EVITA\_CIPHER\_KEY\_NUM

```
#define CRYPTO_EVITA_CIPHER_KEY_NUM 1U
```

Definition at line 78 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.5 CRYPTO\_EVITA\_COPY\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_COPY_KEY_ELEMENT_SIZE (CRYPTO_KE_COPY_KEY_PARENT_KEY_SIZE + CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE_SIZE)
```

Definition at line 239 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.6 CRYPTO\_EVITA\_DERIVE\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_DERIVE_KEY_ELEMENT_SIZE
```

**Value:**

```
(CRYPTO_KE_KEYDERIVATION_KEY_SIZE +
 CRYPTO_KE_KEYDERIVATION_PASSWD_SIZE + \
 CRYPTO_KE_KEYDERIVATION_ITERATIONS_SIZE + \
 CRYPTO_KE_KEYDERIVATION_KEYHANDLE_SIZE +
 CRYPTO_KE_KEYDERIVATION_TYPE_SIZE)
 CRYPTO_KE_KEYDERIVATION_SALT_SIZE +
 CRYPTO_KE_KEYDERIVATION_ALGORITHM_SIZE +
```

Definition at line 222 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.7 CRYPTO\_EVITA\_DERIVE\_KEY\_NUM

```
#define CRYPTO_EVITA_DERIVE_KEY_NUM 2U
```

Definition at line 84 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.8 CRYPTO\_EVITA\_EXCHANGE\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_EXCHANGE_KEY_ELEMENT_SIZE
```

**Value:**

```
(CRYPTO_KE_KEYEXCHANGE_SHAREDVALUE_SIZE +
 CRYPTO_KE_KEYEXCHANGE_BASE_SIZE + \
 CRYPTO_KE_KEYEXCHANGE_OWNPUKEY_SIZE + \
 CRYPTO_KE_KEYEXCHANGE_PEERPUKEY_SIZE + \
 CRYPTO_KE_KEYEXCHANGE_PUBTYPE_SIZE + \
 CRYPTO_KE_KEYEXCHANGE_SM2_PEERTMPUBK_SIZE + \
 CRYPTO_KE_KEYEXCHANGE_SM2_SA_SB_VALUE_SIZE + \
 CRYPTO_KE_KEYEXCHANGE_PRIVKEY_SIZE +
 CRYPTO_KE_KEYEXCHANGE_ALGORITHM_SIZE +
 CRYPTO_KE_KEYEXCHANGE_KEYINFO_SIZE +
 CRYPTO_KE_KEYEXCHANGE_SM2_LOCALTMPKINFO_SIZE +
 CRYPTO_KE_KEYEXCHANGE_SM2_S1_S2_VALUE_SIZE +
 CRYPTO_KE_KEYEXCHANGE_SM2_ROLE_SIZE)
```

Definition at line 214 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.9 CRYPTO\_EVITA\_EXCHANGE\_KEY\_NUM

```
#define CRYPTO_EVITA_EXCHANGE_KEY_NUM 2U
```

Definition at line 81 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.10 CRYPTO\_EVITA\_EXPORT\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_EXPORT_KEY_ELEMENT_SIZE (CRYPTO_KE_EXPORT_KEY_SIZE + CRYPTO_KE_EXPORT_KEY_BLOCK_SIZE)
```

Definition at line 231 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.11 CRYPTO\_EVITA\_EXPORT\_KEY\_NUM

```
#define CRYPTO_EVITA_EXPORT_KEY_NUM 1U
```

Definition at line 93 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.12 CRYPTO\_EVITA\_GENERATE\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_GENERATE_KEY_ELEMENT_SIZE
```

##### Value:

```
(CRYPTO_KE_KEYGENERATE_KEYINFO_SIZE +
 CRYPTO_KE_KEYGENERATE_SEED_SIZE + \
 CRYPTO_KE_KEYGENERATE_KEY_SIZE + \
 CRYPTO_KE_KEYGENERATE_ALGORITHM_SIZE +
 CRYPTO_KE_KEYGENERATE_DH_KEY_INFO_SIZE)
```

Definition at line 226 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.13 CRYPTO\_EVITA\_GENERATE\_KEY\_NUM

```
#define CRYPTO_EVITA_GENERATE_KEY_NUM 1U
```

Definition at line 87 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.14 CRYPTO\_EVITA\_IMPORT\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_IMPORT_KEY_ELEMENT_SIZE (CRYPTO_KE_IMPORT_KEY_SIZE + CRYPTO_KE_IMPORTED_KEY_K↔
EYHANDLE_SIZE)
```

Definition at line 230 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.15 CRYPTO\_EVITA\_IMPORT\_KEY\_NUM

```
#define CRYPTO_EVITA_IMPORT_KEY_NUM 1U
```

Definition at line 90 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.16 CRYPTO\_EVITA\_KEY\_COPY\_KEY\_NUM

```
#define CRYPTO_EVITA_KEY_COPY_KEY_NUM 2U
```

Definition at line 102 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.17 CRYPTO\_EVITA\_KEY\_STATUS\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_KEY_STATUS_ELEMENT_SIZE (CRYPTO_KEY_STATUS_SIZE + CRYPTO_KEY_STATUS_BLOCK_SIZE)
```

Definition at line 235 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.18 CRYPTO\_EVITA\_KEY\_STATUS\_NUM

```
#define CRYPTO_EVITA_KEY_STATUS_NUM 1U
```

Definition at line 99 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.19 CRYPTO\_EVITA\_MAC\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_MAC_KEY_ELEMENT_SIZE (CRYPTO_KEY_MAC_SIZE + CRYPTO_KEY_GENERATE_SEED_SIZE)
```

Definition at line 241 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.20 CRYPTO\_EVITA\_REMOVE\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_REMOVE_KEY_ELEMENT_SIZE CRYPTO_REMOVE_KEY_SIZE
```

Definition at line 233 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.21 CRYPTO\_EVITA\_REMOVE\_KEY\_NUM

```
#define CRYPTO_EVITA_REMOVE_KEY_NUM 1U
```

Definition at line 96 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.22 CRYPTO\_EVITA\_SIGNATURE\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_EVITA_SIGNATURE_KEY_ELEMENT_SIZE (CRYPTO_KEY_SIGNATURE_SIZE + CRYPTO_KEY_SIGNATURE_TIMESTAMPED_SIZE + CRYPTO_KEY_SIGNATURE_RSA_CERT_MODE_SIZE)
```

Definition at line 237 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.23 CRYPTO\_EVITA\_SIGNATURE\_KEY\_NUM

```
#define CRYPTO_EVITA_SIGNATURE_KEY_NUM 1U
```

Definition at line 75 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.24 CRYPTO\_INVALID\_KEY\_ID

```
#define CRYPTO_INVALID_KEY_ID 0xFFFFFFFFU
```

Definition at line 66 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.25 CRYPTO\_KE\_AEAD\_TAG\_SIZE

```
#define CRYPTO_KE_AEAD_TAG_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 134 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.26 CRYPTO\_KE\_CIPHER\_2NDKEY\_SIZE\_SIZE

```
#define CRYPTO_KE_CIPHER_2NDKEY_SIZE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 137 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.27 CRYPTO\_KE\_CIPHER\_CIPHER\_ALG\_SIZE

```
#define CRYPTO_KE_CIPHER_CIPHER_ALG_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 130 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.28 CRYPTO\_KE\_CIPHER\_CURVE\_ID\_SIZE

```
#define CRYPTO_KE_CIPHER_CURVE_ID_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 129 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.29 CRYPTO\_KE\_CIPHER\_IV\_SIZE

```
#define CRYPTO_KE_CIPHER_IV_SIZE ((ehsm_uint32_t)32U)
```

Definition at line 126 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.30 CRYPTO\_KE\_CIPHER\_KDF\_ALG\_SIZE

```
#define CRYPTO_KE_CIPHER_KDF_ALG_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 131 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.31 CRYPTO\_KE\_CIPHER\_KEY\_SIZE

```
#define CRYPTO_KE_CIPHER_KEY_SIZE ((ehsm_uint32_t)sizeof(crypto_evita_key_info_st))
```

Definition at line 125 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.32 CRYPTO\_KE\_CIPHER\_MAC\_ALG\_SIZE

```
#define CRYPTO_KE_CIPHER_MAC_ALG_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 132 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.33 CRYPTO\_KE\_CIPHER\_MAC\_SIZE\_SIZE

```
#define CRYPTO_KE_CIPHER_MAC_SIZE_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 133 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.34 CRYPTO\_KE\_CIPHER\_PROOF\_SIZE

```
#define CRYPTO_KE_CIPHER_PROOF_SIZE CRYPTO_M5_SIZE_U32
```

Definition at line 127 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.35 CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY\_SIZE

```
#define CRYPTO_KE_COPY_KEY_PARENT_KEY_SIZE ((ehsm_uint32_t)sizeof(crypto_copy_key_info_st))
```

Definition at line 199 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.36 CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE\_SIZE

```
#define CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE_SIZE ((ehsm_uint32_t)4)
```

Definition at line 200 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.37 CRYPTO\_KE\_EXPORT\_KEY\_BLOB\_SIZE

```
#define CRYPTO_KE_EXPORT_KEY_BLOB_SIZE ((ehsm_uint32_t)1024U)
```

Definition at line 181 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.38 CRYPTO\_KE\_EXPORT\_KEY\_SIZE

```
#define CRYPTO_KE_EXPORT_KEY_SIZE ((ehsm_uint32_t)sizeof(crypto_key_export_info_st))
```

Definition at line 180 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.39 CRYPTO\_KE\_EXT\_SHE\_KEY\_SIZE

```
#define CRYPTO_KE_EXT_SHE_KEY_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 189 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.40 CRYPTO\_KE\_IMPORT\_KEY\_SIZE

```
#define CRYPTO_KE_IMPORT_KEY_SIZE ((ehsm_uint32_t)sizeof(crypto_import_evita_key_info_st))
```

Definition at line 176 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.



#### 4.26.1.41 CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE\_SIZE

```
#define CRYPTO_KE_IMPORTED_KEY_KEYHANDLE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 177 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.42 CRYPTO\_KE\_KEY\_MATERIAL\_SIZE

```
#define CRYPTO_KE_KEY_MATERIAL_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 187 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.43 CRYPTO\_KE\_KEY\_STATUS\_BLOB\_SIZE

```
#define CRYPTO_KE_KEY_STATUS_BLOB_SIZE ((ehsm_uint32_t) 1024U)
```

Definition at line 196 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.44 CRYPTO\_KE\_KEY\_STATUS\_SIZE

```
#define CRYPTO_KE_KEY_STATUS_SIZE ((ehsm_uint32_t) sizeof(crypto_key_status_info_st))
```

Definition at line 195 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.45 CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM\_SIZE

```
#define CRYPTO_KE_KEYDERIVATION_ALGORITHM_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 164 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.46 CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS\_SIZE

```
#define CRYPTO_KE_KEYDERIVATION_ITERATIONS_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 163 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.47 CRYPTO\_KE\_KEYDERIVATION\_KEY\_SIZE**

```
#define CRYPTO_KE_KEYDERIVATION_KEY_SIZE ((ehsm_uint32_t) sizeof(crypto_create_evita_key_info_st))
```

Definition at line 160 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.48 CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE\_SIZE**

```
#define CRYPTO_KE_KEYDERIVATION_KEYHANDLE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 165 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.49 CRYPTO\_KE\_KEYDERIVATION\_PASSWD\_SIZE**

```
#define CRYPTO_KE_KEYDERIVATION_PASSWD_SIZE ((ehsm_uint32_t) 64U)
```

Definition at line 161 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.50 CRYPTO\_KE\_KEYDERIVATION\_SALT\_SIZE**

```
#define CRYPTO_KE_KEYDERIVATION_SALT_SIZE ((ehsm_uint32_t) 64U)
```

Definition at line 162 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.51 CRYPTO\_KE\_KEYDERIVATION\_TYPE\_SIZE**

```
#define CRYPTO_KE_KEYDERIVATION_TYPE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 166 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.52 CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM\_SIZE**

```
#define CRYPTO_KE_KEYEXCHANGE_ALGORITHM_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 148 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.53 CRYPTO\_KE\_KEYEXCHANGE\_BASE\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_BASE_SIZE ((ehsm_uint32_t) 8U)
```

Definition at line 143 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.54 CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_KEYINFO_SIZE ((ehsm_uint32_t) sizeof(crypto_create_evita_key_info_st))
```

Definition at line 151 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.55 CRYPTO\_KE\_KEYEXCHANGE\_OWNPUBKEY\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_OWNPUBKEY_SIZE ((ehsm_uint32_t) 64U)
```

Definition at line 147 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.56 CRYPTO\_KE\_KEYEXCHANGE\_PEERPUBKEY\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_PEERPUBKEY_SIZE ((ehsm_uint32_t) 512U)
```

Definition at line 150 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.57 CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_PRIVKEY_SIZE ((ehsm_uint32_t) sizeof(crypto_evita_key_info_st))
```

Definition at line 145 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.58 CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_PUBTYPE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 152 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.59 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINFO\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_LOCALTMPKINFO_SIZE ((ehsm_uint32_t)sizeof(crypto_evita_key_info_st))
```

Definition at line 153 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.60 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_PEERTMPPUBK_SIZE ((ehsm_uint32_t)sizeof(ehsm_export_pub_key_st))
```

Definition at line 154 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.61 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_ROLE_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 157 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.62 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_S1_S2_VALUE_SIZE ((ehsm_uint32_t)32U)
```

Definition at line 155 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.63 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE\_SIZE

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_SA_SB_VALUE_SIZE ((ehsm_uint32_t)32U)
```

Definition at line 156 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.64 CRYPTO\_KE\_KEYGENERATE\_ALGORITHM\_SIZE

```
#define CRYPTO_KE_KEYGENERATE_ALGORITHM_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 171 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.65 CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO\_SIZE

```
#define CRYPTO_KE_KEYGENERATE_DH_KEY_INFO_SIZE ((ehsm_uint32_t)sizeof(crypto_copy_key_dh_key_info_st))
```

Definition at line 173 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.66 CRYPTO\_KE\_KEYGENERATE\_KEY\_SIZE

```
#define CRYPTO_KE_KEYGENERATE_KEY_SIZE ((ehsm_uint32_t)4U)
```

Definition at line 172 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.67 CRYPTO\_KE\_KEYGENERATE\_KEYINFO\_SIZE

```
#define CRYPTO_KE_KEYGENERATE_KEYINFO_SIZE ((ehsm_uint32_t)sizeof(crypto_create_evita_key_info_st))
```

Definition at line 169 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.68 CRYPTO\_KE\_KEYGENERATE\_SEED\_SIZE

```
#define CRYPTO_KE_KEYGENERATE_SEED_SIZE ((ehsm_uint32_t)32U)
```

Definition at line 170 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.69 CRYPTO\_KE\_MAC\_KEY\_SIZE

```
#define CRYPTO_KE_MAC_KEY_SIZE ((ehsm_uint32_t)sizeof(crypto_evita_key_info_st))
```

Definition at line 112 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.70 CRYPTO\_KE\_MAC\_PROOF\_SIZE

```
#define CRYPTO_KE_MAC_PROOF_SIZE CRYPTO_M4_SIZE_U32
```

Definition at line 113 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.71 CRYPTO\_KE\_RANDOM\_ALGORITHM\_SIZE

```
#define CRYPTO_KE_RANDOM_ALGORITHM_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 122 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.72 CRYPTO\_KE\_RANDOM\_SEED\_STATE\_SIZE

```
#define CRYPTO_KE_RANDOM_SEED_STATE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 121 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.73 CRYPTO\_KE\_REMOVE\_KEY\_SIZE

```
#define CRYPTO_KE_REMOVE_KEY_SIZE ((ehsm_uint32_t) sizeof(crypto_evita_key_info_st))
```

Definition at line 192 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.74 CRYPTO\_KE\_SHE\_PLAIN\_KEY\_SIZE

```
#define CRYPTO_KE_SHE_PLAIN_KEY_SIZE ((ehsm_uint32_t) 32U)
```

Definition at line 184 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.75 CRYPTO\_KE\_SIGNATURE\_KEY\_SIZE

```
#define CRYPTO_KE_SIGNATURE_KEY_SIZE ((ehsm_uint32_t) sizeof(crypto_evita_key_info_st))
```

Definition at line 116 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

#### 4.26.1.76 CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE\_SIZE

```
#define CRYPTO_KE_SIGNATURE_RSA_CRT_MODE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 118 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.77 CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED\_SIZE**

```
#define CRYPTO_KE_SIGNATURE_TIMESTAMPED_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 117 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.78 CRYPTO\_KEY10\_EVITA\_IMPORT\_KEY**

```
#define CRYPTO_KEY10_EVITA_IMPORT_KEY 9
```

Definition at line 45 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.79 CRYPTO\_KEY11\_EVITA\_EXPORT\_KEY**

```
#define CRYPTO_KEY11_EVITA_EXPORT_KEY 10
```

Definition at line 48 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.80 CRYPTO\_KEY12\_EVITA\_REMOVE\_KEY**

```
#define CRYPTO_KEY12_EVITA_REMOVE_KEY 11
```

Definition at line 51 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.81 CRYPTO\_KEY13\_EVITA\_KEY\_STATUS**

```
#define CRYPTO_KEY13_EVITA_KEY_STATUS 12
```

Definition at line 54 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.82 CRYPTO\_KEY14\_EVITA\_KEY\_COPY\_KEY**

```
#define CRYPTO_KEY14_EVITA_KEY_COPY_KEY 13
```

Definition at line 57 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.83 CRYPTO\_KEY15\_EVITA\_KEY\_COPY\_KEY**

```
#define CRYPTO_KEY15_EVITA_KEY_COPY_KEY 14
```

Definition at line 58 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.84 CRYPTO\_KEY16\_CERTIFICATE\_KEY**

```
#define CRYPTO_KEY16_CERTIFICATE_KEY 15
```

Definition at line 61 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.85 CRYPTO\_KEY17\_CERTIFICATE\_KEY**

```
#define CRYPTO_KEY17_CERTIFICATE_KEY 16
```

Definition at line 62 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.86 CRYPTO\_KEY18\_MAC\_KEY**

```
#define CRYPTO_KEY18_MAC_KEY 17
```

Definition at line 64 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.87 CRYPTO\_KEY1\_SHE\_KEY**

```
#define CRYPTO_KEY1_SHE_KEY 0
```

Definition at line 22 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.88 CRYPTO\_KEY2\_SHE\_PLAIN\_KEY**

```
#define CRYPTO_KEY2_SHE_PLAIN_KEY 1
```

Definition at line 25 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.



**4.26.1.89 CRYPTO\_KEY3\_EVITA\_SIGNATURE\_KEY**

```
#define CRYPTO_KEY3_EVITA_SIGNATURE_KEY 2
```

Definition at line 28 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.90 CRYPTO\_KEY4\_EVITA\_CIPHER\_KEY**

```
#define CRYPTO_KEY4_EVITA_CIPHER_KEY 3
```

Definition at line 31 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.91 CRYPTO\_KEY5\_EVITA\_EXCHANGE\_KEY**

```
#define CRYPTO_KEY5_EVITA_EXCHANGE_KEY 4
```

Definition at line 34 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.92 CRYPTO\_KEY6\_EVITA\_EXCHANGE\_KEY**

```
#define CRYPTO_KEY6_EVITA_EXCHANGE_KEY 5
```

Definition at line 35 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.93 CRYPTO\_KEY7\_EVITA\_DERIVE\_KEY**

```
#define CRYPTO_KEY7_EVITA_DERIVE_KEY 6
```

Definition at line 38 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.94 CRYPTO\_KEY8\_EVITA\_DERIVE\_KEY**

```
#define CRYPTO_KEY8_EVITA_DERIVE_KEY 7
```

Definition at line 39 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.95 CRYPTO\_KEY9\_EVITA\_GENERATE\_KEY**

```
#define CRYPTO_KEY9_EVITA_GENERATE_KEY 8
```

Definition at line 42 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.96 CRYPTO\_MAC\_KEY\_NUM**

```
#define CRYPTO_MAC_KEY_NUM 1U
```

Definition at line 108 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.97 CRYPTO\_MAX\_KEY\_NUM**

```
#define CRYPTO_MAX_KEY_NUM 18
```

Definition at line 19 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.98 CRYPTO\_SHE\_KEY\_ELEMENT\_SIZE**

```
#define CRYPTO_SHE_KEY_ELEMENT_SIZE
```

**Value:**

```
(CRYPTO_KE_MAC_PROOF_SIZE + CRYPTO_KE_CIPHER_PROOF_SIZE
+ CRYPTO_KE_EXT_SHE_KEY_SIZE + \ CRYPTO_KE_CIPHER_IV_SIZE +
CRYPTO_KE_KEYGENERATE_SEED_SIZE +
CRYPTO_KE_KEY_MATERIAL_SIZE)
```

Definition at line 203 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.99 CRYPTO\_SHE\_KEY\_NUM**

```
#define CRYPTO_SHE_KEY_NUM 1U
```

Definition at line 69 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.100 CRYPTO\_SHE\_PLAIN\_KEY\_ELEMENT\_SIZE

```
#define CRYPTO_SHE_PLAIN_KEY_ELEMENT_SIZE
```

**Value:**

```
(CRYPTO_KE_KEY_MATERIAL_SIZE +
 CRYPTO_KE_CIPHER_IV_SIZE + \
```

CRYPTO\_KE\_KEYGENERATE\_SEED\_SIZE)

Definition at line 206 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.101 CRYPTO\_SHE\_PLAIN\_KEY\_NUM

```
#define CRYPTO_SHE_PLAIN_KEY_NUM 1U
```

Definition at line 72 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.102 CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE\_SIZE

```
#define CYRPTO_KE_KEYEXCHANGE_SHAREDVALUE_SIZE ((ehsm_uint32_t) 4U)
```

Definition at line 141 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

## 4.26.1.103 KEY\_ELEMENT\_NUM

```
#define KEY_ELEMENT_NUM
```

**Value:**

```
(CRYPTO_SHE_KEY_NUM * 6 + \
```

CRYPTO\_SHE\_PLAIN\_KEY\_NUM \* 3 +\  
CRYPTO\_EVITA\_SIGNATURE\_KEY\_NUM \* 3 +\  
CRYPTO\_EVITA\_CIPHER\_KEY\_NUM \* 8 +\  
CRYPTO\_EVITA\_EXCHANGE\_KEY\_NUM \* 15 +\  
CRYPTO\_EVITA\_DERIVE\_KEY\_NUM \* 7 +\  
CRYPTO\_EVITA\_GENERATE\_KEY\_NUM \* 4 +\  
CRYPTO\_EVITA\_IMPORT\_KEY\_NUM \* 2 +\  
CRYPTO\_EVITA\_KEY\_STATUS\_NUM \* 2 +\  
CRYPTO\_EVITA\_REMOVE\_KEY\_NUM \* 1 +\  
CRYPTO\_EVITA\_EXPORT\_KEY\_NUM \* 2 +\  
CRYPTO\_EVITA\_KEY\_COPY\_KEY\_NUM \* 2 +\  
CRYPTO\_CERTIFICATE\_KEY\_NUM \* 4 +\  
CRYPTO\_MAC\_KEY\_NUM \* 2 +\  
KEY\_ELEMENT\_RESERVER)

Definition at line 248 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.104 KEY\_ELEMENT\_RESERVER**

```
#define KEY_ELEMENT_RESERVER 8
```

Definition at line 246 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.105 KEY\_ELEMENT\_VALUE\_BUFFER\_RESERVER**

```
#define KEY_ELEMENT_VALUE_BUFFER_RESERVER 256
```

Definition at line 264 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.1.106 KEY\_ELEMENT\_VALUE\_BUFFER\_SIZE**

```
#define KEY_ELEMENT_VALUE_BUFFER_SIZE
```

**Value:**

```
(CRYPTO_SHE_KEY_ELEMENT_SIZE * CRYPTO_SHE_KEY_NUM + \
 CRYPTO_SHE_PLAIN_KEY_ELEMENT_SIZE *
 CRYPTO_SHE_PLAIN_KEY_NUM + \
 CRYPTO_EVITA_CIPHER_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_CIPHER_KEY_NUM + \
 CRYPTO_EVITA_EXCHANGE_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_EXCHANGE_KEY_NUM + \
 CRYPTO_EVITA_DERIVE_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_DERIVE_KEY_NUM + \
 CRYPTO_EVITA_GENERATE_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_GENERATE_KEY_NUM + \
 CRYPTO_EVITA_IMPORT_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_IMPORT_KEY_NUM + \
 CRYPTO_EVITA_EXPORT_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_EXPORT_KEY_NUM + \
 CRYPTO_EVITA_REMOVE_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_REMOVE_KEY_NUM + \
 CRYPTO_EVITA_SIGNATURE_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_SIGNATURE_KEY_NUM + \
 CRYPTO_EVITA_KEY_STATUS_ELEMENT_SIZE *
 CRYPTO_EVITA_KEY_STATUS_NUM + \
 CRYPTO_EVITA_COPY_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_COPY_KEY_NUM + \
 CRYPTO_EVITA_CERTIFICATE_KEY_ELEMENT_SIZE *
 CRYPTO_EVITA_CERTIFICATE_KEY_NUM + \
 CRYPTO_EVITA_MAC_KEY_ELEMENT_SIZE *
 CRYPTO_MAC_KEY_NUM + \
 KEY_ELEMENT_VALUE_BUFFER_RESERVER)
```

Definition at line 266 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

**4.26.2 Enumeration Type Documentation****4.26.2.1 crypto\_object\_type\_e**

```
enum crypto_object_type_e
```

## Enumerator

|                           |  |
|---------------------------|--|
| CRYPTO_OBJECT_TYPE_SKE    |  |
| CRYPTO_OBJECT_TYPE_PKE    |  |
| CRYPTO_OBJECT_TYPE_TRNG   |  |
| CRYPTO_OBJECT_TYPE_HASH   |  |
| CRYPTO_OBJECT_TYPE_KEY    |  |
| CRYPTO_OBJECT_TYPE_SYSMGR |  |
| CRYPTO_OBJECT_TYPE_MAX    |  |

Definition at line 284 of file eHSM\_If\_Asr\_KeyCfg\_Ip.h.

### 4.26.3 Function Documentation

#### 4.26.3.1 ehsm\_get\_crypto\_driver\_object()

```
CryptoDriverObject* ehsm_get_crypto_driver_object (
 void) [inline]
```

Definition at line 1805 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

#### 4.26.3.2 ehsm\_get\_crypto\_ke\_info()

```
crypto_key_element_type_info_st* ehsm_get_crypto_ke_info (
 ehsm_uint32_t crypto_key_type,
 ehsm_uint32_t * size)
```

Definition at line 707 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

#### 4.26.3.3 ehsm\_printf\_ke\_size()

```
void ehsm_printf_ke_size (
 void)
```

Definition at line 773 of file eHSM\_If\_Asr\_KeyCfg\_Ip.c.

## 4.27 eHSM\_If\_Asr\_Types\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "Asr_Standard_Types.h"
```

## Classes

- struct [crypto\\_key\\_element\\_type\\_info\\_st](#)
- struct [CryptoKeyType](#)
- struct [CryptoKey](#)
- struct [ehsm\\_crypto\\_key](#)
- struct [crypto\\_create\\_evita\\_key\\_info](#)
- struct [crypto\\_evita\\_key\\_info](#)
- struct [crypto\\_import\\_evita\\_key\\_info](#)
- struct [crypto\\_key\\_export\\_info](#)
- struct [crypto\\_exported\\_key](#)
- struct [crypto\\_key\\_status\\_info](#)
- struct [crypto\\_she\\_key](#)
- struct [crypto\\_copy\\_key\\_info](#)
- struct [crypto\\_copy\\_key\\_dh\\_key\\_info](#)
- struct [crypto\\_key\\_derive\\_info](#)

## Macros

- #define [CRYPTO\\_KE\\_SIGNATURE\\_TIMESTAMPED](#) ((ehsm\_uint32\_t)1101U)  
*Version information.*
- #define [CRYPTO\\_KE\\_SIGNATURE\\_RSA\\_CRT\\_MODE](#) ((ehsm\_uint32\_t)1102U)
- #define [CRYPTO\\_KE\\_CIPHER\\_CURVE\\_ID](#) ((ehsm\_uint32\_t)1110U)
- #define [CRYPTO\\_KE\\_CIPHER\\_CIPHER\\_ALG](#) ((ehsm\_uint32\_t)1111U)
- #define [CRYPTO\\_KE\\_CIPHER\\_KDF\\_ALG](#) ((ehsm\_uint32\_t)1112U)
- #define [CRYPTO\\_KE\\_CIPHER\\_MAC\\_ALG](#) ((ehsm\_uint32\_t)1113U)
- #define [CRYPTO\\_KE\\_CIPHER\\_MAC\\_SIZE](#) ((ehsm\_uint32\_t)1114U)
- #define [CRYPTO\\_KE\\_CIPHER\\_TAG\\_SIZE](#) ((ehsm\_uint32\_t)1115U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_PEERKEY](#) ((ehsm\_uint32\_t)1120U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_KEYINFO](#) ((ehsm\_uint32\_t)1001U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_PUBTYPE](#) ((ehsm\_uint32\_t)1121U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_SM2\\_LOCALTMPKINFO](#) ((ehsm\_uint32\_t)1122U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_SM2\\_PEERTMPKINFO](#) ((ehsm\_uint32\_t)1123U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_SM2\\_S1\\_S2\\_VALUE](#) ((ehsm\_uint32\_t)1124U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_SM2\\_SA\\_SB\\_VALUE](#) ((ehsm\_uint32\_t)1125U)
- #define [CRYPTO\\_KE\\_KEYEXCHANGE\\_SM2\\_ROLE](#) ((ehsm\_uint32\_t)1126U)
- #define [CRYPTO\\_KE\\_KEYDERIVATION\\_KEYHANDLE](#) ((ehsm\_uint32\_t)1000U)
- #define [CRYPTO\\_KE\\_KEYGENERATE\\_KEYINFO](#) ((ehsm\_uint32\_t)1001U)
- #define [CRYPTO\\_KE\\_KEYGENERATE\\_DH\\_KEY\\_INFO](#) ((ehsm\_uint32\_t)1130U)
- #define [CRYPTO\\_KE\\_CERTIFICATE\\_SIGNEDDATA](#) ((ehsm\_uint32\_t)1140U)
- #define [CRYPTO\\_KE\\_IMPORT\\_KEY](#) ((ehsm\_uint32\_t)1U)
- #define [CRYPTO\\_KE\\_IMPORTED\\_KEY\\_KEYHANDLE](#) ((ehsm\_uint32\_t)1000U)
- #define [CRYPTO\\_KE\\_EXPORT\\_KEY](#) ((ehsm\_uint32\_t)1U)
- #define [CRYPTO\\_KE\\_EXPORT\\_KEY\\_BLOB](#) ((ehsm\_uint32\_t)1150U)
- #define [CRYPTO\\_KE\\_KEY\\_MATERIAL](#) ((ehsm\_uint32\_t)1U)
- #define [CRYPTO\\_KE\\_KEY\\_HANDLE](#) ((ehsm\_uint32\_t)1160U)
- #define [CRYPTO\\_KE\\_EXT\\_SHE\\_KEY](#) ((ehsm\_uint32\_t)1161U)
- #define [CRYPTO\\_KE\\_REMOVE\\_KEY](#) ((ehsm\_uint32\_t)1U)
- #define [CRYPTO\\_KE\\_KEY\\_STATUS](#) ((ehsm\_uint32\_t)1U)
- #define [CRYPTO\\_KE\\_KEY\\_STATUS\\_BLOB](#) ((ehsm\_uint32\_t)1170U)
- #define [CRYPTO\\_KE\\_COPY\\_KEY\\_PARENT\\_KEY](#) ((ehsm\_uint32\_t)1U)
- #define [CRYPTO\\_KE\\_COPY\\_KEY\\_TARGET\\_KEY\\_HANDLE](#) ((ehsm\_uint32\_t)1000U)
- #define [CRYPTO\\_KE\\_KEYDERIVATION\\_KEYHANDLE](#) ((ehsm\_uint32\_t)1000U)
- #define [CRYPTO\\_KE\\_KEYDERIVATION\\_KEY](#) ((ehsm\_uint32\_t)1001U)
- #define [CRYPTO\\_KE\\_KEYDERIVATION\\_TYPE](#) ((ehsm\_uint32\_t)1190U)
- #define [CRYPTO\\_KE\\_ATTR\\_ALLOW\\_PARTIAL\\_ACCESS](#) ((ehsm\_uint32\_t)(1U << 0))

- #define CRYPTO\_KEY\_ATTR\_READ\_ACCESS ((ehsm\_uint32\_t)(1U << 1))
- #define CRYPTO\_KEY\_ATTR\_WRITE\_ACCESS ((ehsm\_uint32\_t)(1U << 2))
- #define CRYPTO\_M1\_SIZE\_U32 ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_M2\_SIZE\_U32 ((ehsm\_uint32\_t)32U)
- #define CRYPTO\_M3\_SIZE\_U32 ((ehsm\_uint32\_t)16U)
- #define CRYPTO\_M4\_SIZE\_U32 ((ehsm\_uint32\_t)48U)
- #define CRYPTO\_M5\_SIZE\_U32 ((ehsm\_uint32\_t)16U)
- #define CRYPTO\_INDEX\_M1\_U32 (0U)
- #define CRYPTO\_INDEX\_M2\_U32 ((ehsm\_uint32\_t)(CRYPTO\_M1\_SIZE\_U32))
- #define CRYPTO\_INDEX\_M3\_U32 ((ehsm\_uint32\_t)(CRYPTO\_M1\_SIZE\_U32 + CRYPTO\_M2\_SIZE\_U32))
- #define CRYPTO\_INDEX\_M4\_U32 ((ehsm\_uint32\_t)(CRYPTO\_M1\_SIZE\_U32 + CRYPTO\_M2\_SIZE\_U32 + CRYPTO\_M3\_SIZE\_U32))
- #define CRYPTO\_INDEX\_M5\_U32 ((ehsm\_uint32\_t)(CRYPTO\_M1\_SIZE\_U32 + CRYPTO\_M2\_SIZE\_U32 + CRYPTO\_M3\_SIZE\_U32 + CRYPTO\_M4\_SIZE\_U32))
- #define CRYPTO\_SHE\_SIZE\_OUT\_U32 ((ehsm\_uint32\_t)(CRYPTO\_M1\_SIZE\_U32 + CRYPTO\_M2\_SIZE\_U32 + CRYPTO\_M3\_SIZE\_U32 + CRYPTO\_M4\_SIZE\_U32 + CRYPTO\_M5\_SIZE\_U32))
- #define CRYPTO\_SHE\_SIZE\_IN\_U32 ((ehsm\_uint32\_t)(CRYPTO\_M1\_SIZE\_U32 + CRYPTO\_M2\_SIZE\_U32 + CRYPTO\_M3\_SIZE\_U32))
- #define EVITA\_KEY\_USE\_FLAG\_SIGN 0x1
- #define EVITA\_KEY\_USE\_FLAG\_VERIFY 0x2
- #define EVITA\_KEY\_USE\_FLAG\_ENCRYPT 0x4
- #define EVITA\_KEY\_USE\_FLAG\_DECRYPT 0x8
- #define EVITA\_KEY\_USE\_FLAG\_TIMESTAMP 0x10
- #define EVITA\_KEY\_USE\_FLAG\_SECUREBOOT 0x20
- #define EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE 0x40
- #define EVITA\_KEY\_USE\_FLAG\_DHKE 0x80
- #define EVITA\_KEY\_USE\_FLAG\_UTCSYNC 0x100
- #define EVITA\_KEY\_USE\_FLAG\_TRANSPORT 0x200
- #define EVITA\_KEY\_USE\_FLAG\_REMOVE 0x400
- #define CRYPTO\_SHE\_MAC\_KEY\_ID 1U
- #define CRYPTO\_SHE\_MAC\_KEY\_NUM 1U
- #define CRYPTO\_SHE\_MAC\_KEY\_ID 1U
- #define CRYPTO\_MAX\_AUTH\_VALUE\_SIZE 32
- #define CRYPTO\_MAX\_KEY\_FLAG\_ELEMENT 12
- #define CRYPTO\_MAX\_KEY\_ELEMENTS\_OF\_KEY\_TYPE 16

## Typedefs

- typedef struct ehsm\_crypto\_key ehsm\_crypto\_key\_st
- typedef struct crypto\_create\_evita\_key\_info crypto\_create\_evita\_key\_info\_st
- typedef struct crypto\_evita\_key\_info crypto\_evita\_key\_info\_st
- typedef struct crypto\_import\_evita\_key\_info crypto\_import\_evita\_key\_info\_st
- typedef struct crypto\_key\_export\_info crypto\_key\_export\_info\_st
- typedef struct crypto\_exported\_key crypto\_exported\_key\_st
- typedef struct crypto\_key\_status\_info crypto\_key\_status\_info\_st
- typedef struct crypto\_she\_key crypto\_she\_key\_st
- typedef struct crypto\_copy\_key\_info crypto\_copy\_key\_info\_st
- typedef struct crypto\_copy\_key\_dh\_key\_info crypto\_copy\_key\_dh\_key\_info\_st
- typedef struct crypto\_key\_derive\_info crypto\_key\_derive\_info\_st

## Enumerations

- enum `ehsm_key_type_e` { `EHSM_KEY_TYPE_SHE` = 0, `EHSM_KEY_TYPE_EVITA`, `EHSM_KEY_TYPE_INVALID` }
- enum `key_storage_type_e` { `KEY_STORAGE_TYPE_RAM` = 0, `KEY_STORAGE_TYPE_NVM`, `KEY_STORAGE_TYPE_END` }
- enum `ehsm_key_use_state_e` { `EHSM_KEY_USE_STATE_FREE` = 0, `EHSM_KEY_USE_STATE_IN_USE` }
- enum `crypto_key_type_e` { `CRYPTO_KEY_TYPE_SHE` = 1, `CRYPTO_KEY_TYPE_SHE_PLAIN`, `CRYPTO_KEY_TYPE_SIGNATURE`, `CRYPTO_KEY_TYPE_CIPHER`, `CRYPTO_KEY_TYPE_KEY_EXCHANGE`, `CRYPTO_KEY_TYPE_KEY_DERIVATION`, `CRYPTO_KEY_TYPE_KEY_GENERATE`, `CRYPTO_KEY_TYPE_KEY_IMPORT`, `CRYPTO_KEY_TYPE_KEY_EXPORT`, `CRYPTO_KEY_TYPE_KEY_REMOVE`, `CRYPTO_KEY_TYPE_KEY_STATUS`, `CRYPTO_KEY_TYPE_KEY_COPY`, `CRYPTO_KEY_TYPE_CERTIFICATE_KEY`, `CRYPTO_KEY_TYPE_MAC`, `CRYPTO_KEY_TYPE_INVALID` }

## Functions

- void `CryIf_CallbackNotification` (`Crypto_JobType` \*job, `Std_HsmReturnType` result)

### 4.27.1 Macro Definition Documentation

#### 4.27.1.1 CRYPTO\_INDEX\_M1\_U32

```
#define CRYPTO_INDEX_M1_U32 (0U)
```

Definition at line 105 of file `eHSM_If_Asr_Types_Ip.h`.

#### 4.27.1.2 CRYPTO\_INDEX\_M2\_U32

```
#define CRYPTO_INDEX_M2_U32 ((ehsm_uint32_t)(CRYPTO_M1_SIZE_U32))
```

Definition at line 106 of file `eHSM_If_Asr_Types_Ip.h`.

#### 4.27.1.3 CRYPTO\_INDEX\_M3\_U32

```
#define CRYPTO_INDEX_M3_U32 ((ehsm_uint32_t)(CRYPTO_M1_SIZE_U32 + CRYPTO_M2_SIZE_U32))
```

Definition at line 107 of file `eHSM_If_Asr_Types_Ip.h`.



#### 4.27.1.4 CRYPTO\_INDEX\_M4\_U32

```
#define CRYPTO_INDEX_M4_U32 ((ehsm_uint32_t)(CRYPTO_M1_SIZE_U32 + CRYPTO_M2_SIZE_U32 + CRYPTO_M3_SIZE_U32))
```

Definition at line 108 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.5 CRYPTO\_INDEX\_M5\_U32

```
#define CRYPTO_INDEX_M5_U32 ((ehsm_uint32_t)(CRYPTO_M1_SIZE_U32 + CRYPTO_M2_SIZE_U32 + CRYPTO_M3_SIZE_U32 + CRYPTO_M4_SIZE_U32))
```

Definition at line 109 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.6 CRYPTO\_KE\_CERTIFICATE\_SIGNEDDATA

```
#define CRYPTO_KE_CERTIFICATE_SIGNEDDATA ((ehsm_uint32_t)1140U)
```

Definition at line 59 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.7 CRYPTO\_KE\_CIPHER\_CIPHER\_ALG

```
#define CRYPTO_KE_CIPHER_CIPHER_ALG ((ehsm_uint32_t)1111U)
```

Definition at line 34 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.8 CRYPTO\_KE\_CIPHER\_CURVE\_ID

```
#define CRYPTO_KE_CIPHER_CURVE_ID ((ehsm_uint32_t)1110U)
```

Definition at line 33 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.9 CRYPTO\_KE\_CIPHER\_KDF\_ALG

```
#define CRYPTO_KE_CIPHER_KDF_ALG ((ehsm_uint32_t)1112U)
```

Definition at line 35 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.10 CRYPTO\_KE\_CIPHER\_MAC\_ALG

```
#define CRYPTO_KE_CIPHER_MAC_ALG ((ehsm_uint32_t)1113U)
```

Definition at line 36 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.11 CRYPTO\_KE\_CIPHER\_MAC\_SIZE

```
#define CRYPTO_KE_CIPHER_MAC_SIZE ((ehsm_uint32_t)1114U)
```

Definition at line 37 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.12 CRYPTO\_KE\_CIPHER\_TAG\_SIZE

```
#define CRYPTO_KE_CIPHER_TAG_SIZE ((ehsm_uint32_t)1115U)
```

Definition at line 38 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.13 CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY

```
#define CRYPTO_KE_COPY_KEY_PARENT_KEY ((ehsm_uint32_t)1U)
```

Definition at line 84 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.14 CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE

```
#define CRYPTO_KE_COPY_KEY_TARGET_KEY_HANDLE ((ehsm_uint32_t)1000U)
```

Definition at line 85 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.15 CRYPTO\_KE\_EXPORT\_KEY

```
#define CRYPTO_KE_EXPORT_KEY ((ehsm_uint32_t)1U)
```

Definition at line 66 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.16 CRYPTO\_KE\_EXPORT\_KEY\_BLOB

```
#define CRYPTO_KE_EXPORT_KEY_BLOB ((ehsm_uint32_t)1150U)
```

Definition at line 67 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.17 CRYPTO\_KE\_EXT\_SHE\_KEY

```
#define CRYPTO_KE_EXT_SHE_KEY ((ehsm_uint32_t)1161U)
```

Definition at line 74 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.18 CRYPTO\_KE\_IMPORT\_KEY

```
#define CRYPTO_KE_IMPORT_KEY ((ehsm_uint32_t)1U)
```

Definition at line 62 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.19 CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE

```
#define CRYPTO_KE_IMPORTED_KEY_KEYHANDLE ((ehsm_uint32_t)1000U)
```

Definition at line 63 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.20 CRYPTO\_KE\_KEY\_HANDLE

```
#define CRYPTO_KE_KEY_HANDLE ((ehsm_uint32_t)1160U)
```

Definition at line 72 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.21 CRYPTO\_KE\_KEY\_MATERIAL

```
#define CRYPTO_KE_KEY_MATERIAL ((ehsm_uint32_t)1U)
```

Definition at line 70 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.22 CRYPTO\_KE\_KEY\_STATUS

```
#define CRYPTO_KE_KEY_STATUS ((ehsm_uint32_t)1U)
```

Definition at line 80 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.23 CRYPTO\_KE\_KEY\_STATUS\_BLOB

```
#define CRYPTO_KE_KEY_STATUS_BLOB ((ehsm_uint32_t)1170U)
```

Definition at line 81 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.24 CRYPTO\_KE\_KEYDERIVATION\_KEY

```
#define CRYPTO_KE_KEYDERIVATION_KEY ((ehsm_uint32_t)1001U)
```

Definition at line 90 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.25 CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE [1/2]

```
#define CRYPTO_KE_KEYDERIVATION_KEYHANDLE ((ehsm_uint32_t)1000U)
```

Definition at line 88 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.26 CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE [2/2]

```
#define CRYPTO_KE_KEYDERIVATION_KEYHANDLE ((ehsm_uint32_t)1000U)
```

Definition at line 88 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.27 CRYPTO\_KE\_KEYDERIVATION\_TYPE

```
#define CRYPTO_KE_KEYDERIVATION_TYPE ((ehsm_uint32_t)1190U)
```

Definition at line 91 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.28 CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO**

```
#define CRYPTO_KE_KEYEXCHANGE_KEYINFO ((ehsm_uint32_t)1001U)
```

Definition at line 43 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.29 CRYPTO\_KE\_KEYEXCHANGE\_PEERPUBKEY**

```
#define CRYPTO_KE_KEYEXCHANGE_PEERPUBKEY ((ehsm_uint32_t)1120U)
```

Definition at line 42 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.30 CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE**

```
#define CRYPTO_KE_KEYEXCHANGE_PUBTYPE ((ehsm_uint32_t)1121U)
```

Definition at line 44 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.31 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINFO**

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_LOCALTMPKINFO ((ehsm_uint32_t)1122U)
```

Definition at line 45 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.32 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK**

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_PEERTMPPUBK ((ehsm_uint32_t)1123U)
```

Definition at line 46 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.33 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE**

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_ROLE ((ehsm_uint32_t)1126U)
```

Definition at line 49 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.34 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE**

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_S1_S2_VALUE ((ehsm_uint32_t)1124U)
```

Definition at line 47 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.35 CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE**

```
#define CRYPTO_KE_KEYEXCHANGE_SM2_SA_SB_VALUE ((ehsm_uint32_t)1125U)
```

Definition at line 48 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.36 CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO**

```
#define CRYPTO_KE_KEYGENERATE_DH_KEY_INFO ((ehsm_uint32_t)1130U)
```

Definition at line 56 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.37 CRYPTO\_KE\_KEYGENERATE\_KEYINFO**

```
#define CRYPTO_KE_KEYGENERATE_KEYINFO ((ehsm_uint32_t)1001U)
```

Definition at line 55 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.38 CRYPTO\_KE\_REMOVE\_KEY**

```
#define CRYPTO_KE_REMOVE_KEY ((ehsm_uint32_t)1U)
```

Definition at line 77 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.39 CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE**

```
#define CRYPTO_KE_SIGNATURE_RSA_CRT_MODE ((ehsm_uint32_t)1102U)
```

Definition at line 29 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.40 CRYPTO\_KEY\_SIGNATURE\_TIMESTAMPED

```
#define CRYPTO_KEY_SIGNATURE_TIMESTAMPED ((ehsm_uint32_t)1101U)
```

Version information.

Definition at line 28 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.41 CRYPTO\_KEY\_ATTR\_ALLOW\_PARTIAL\_ACCESS

```
#define CRYPTO_KEY_ATTR_ALLOW_PARTIAL_ACCESS ((ehsm_uint32_t)(1U << 0))
```

Definition at line 94 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.42 CRYPTO\_KEY\_ATTR\_READ\_ACCESS

```
#define CRYPTO_KEY_ATTR_READ_ACCESS ((ehsm_uint32_t)(1U << 1))
```

Definition at line 95 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.43 CRYPTO\_KEY\_ATTR\_WRITE\_ACCESS

```
#define CRYPTO_KEY_ATTR_WRITE_ACCESS ((ehsm_uint32_t)(1U << 2))
```

Definition at line 96 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.44 CRYPTO\_M1\_SIZE\_U32

```
#define CRYPTO_M1_SIZE_U32 ((ehsm_uint32_t)32U)
```

Definition at line 99 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.45 CRYPTO\_M2\_SIZE\_U32

```
#define CRYPTO_M2_SIZE_U32 ((ehsm_uint32_t)32U)
```

Definition at line 100 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.46 CRYPTO\_M3\_SIZE\_U32**

```
#define CRYPTO_M3_SIZE_U32 ((ehsm_uint32_t)16U)
```

Definition at line 101 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.47 CRYPTO\_M4\_SIZE\_U32**

```
#define CRYPTO_M4_SIZE_U32 ((ehsm_uint32_t)48U)
```

Definition at line 102 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.48 CRYPTO\_M5\_SIZE\_U32**

```
#define CRYPTO_M5_SIZE_U32 ((ehsm_uint32_t)16U)
```

Definition at line 103 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.49 CRYPTO\_MAX\_AUTH\_VALUE\_SIZE**

```
#define CRYPTO_MAX_AUTH_VALUE_SIZE 32
```

Definition at line 131 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.50 CRYPTO\_MAX\_KEY\_ELEMENTS\_OF\_KEY\_TYPE**

```
#define CRYPTO_MAX_KEY_ELEMENTS_OF_KEY_TYPE 16
```

Definition at line 135 of file eHSM\_If\_Asr\_Types\_Ip.h.

**4.27.1.51 CRYPTO\_MAX\_KEY\_FLAG\_ELEMENT**

```
#define CRYPTO_MAX_KEY_FLAG_ELEMENT 12
```

Definition at line 132 of file eHSM\_If\_Asr\_Types\_Ip.h.



**4.27.1.52 CRYPTO\_SHE\_MAC\_KEY\_ID** [1/2]

```
#define CRYPTO_SHE_MAC_KEY_ID 1U
```

Definition at line 129 of file eHSM\_If\_Asr\_Types\_lp.h.

**4.27.1.53 CRYPTO\_SHE\_MAC\_KEY\_ID** [2/2]

```
#define CRYPTO_SHE_MAC_KEY_ID 1U
```

Definition at line 129 of file eHSM\_If\_Asr\_Types\_lp.h.

**4.27.1.54 CRYPTO\_SHE\_MAC\_KEY\_NUM**

```
#define CRYPTO_SHE_MAC_KEY_NUM 1U
```

Definition at line 128 of file eHSM\_If\_Asr\_Types\_lp.h.

**4.27.1.55 CRYPTO\_SHE\_SIZE\_IN\_U32**

```
#define CRYPTO_SHE_SIZE_IN_U32 ((ehsm_uint32_t)(CRYPTO_M1_SIZE_U32 + CRYPTO_M2_SIZE_U32 + CRYPTO_M3_SIZE_U32))
```

Definition at line 112 of file eHSM\_If\_Asr\_Types\_lp.h.

**4.27.1.56 CRYPTO\_SHE\_SIZE\_OUT\_U32**

```
#define CRYPTO_SHE_SIZE_OUT_U32 ((ehsm_uint32_t)(CRYPTO_M1_SIZE_U32 + CRYPTO_M2_SIZE_U32 + CRYPTO_M3_SIZE_U32 + CRYPTO_M4_SIZE_U32 + CRYPTO_M5_SIZE_U32))
```

Definition at line 111 of file eHSM\_If\_Asr\_Types\_lp.h.

**4.27.1.57 EVITA\_KEY\_USE\_FLAG\_DECRYPT**

```
#define EVITA_KEY_USE_FLAG_DECRYPT 0x8
```

Definition at line 118 of file eHSM\_If\_Asr\_Types\_lp.h.

#### 4.27.1.58 EVITA\_KEY\_USE\_FLAG\_DHKE

```
#define EVITA_KEY_USE_FLAG_DHKE 0x80
```

Definition at line 122 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.59 EVITA\_KEY\_USE\_FLAG\_ENCRYPT

```
#define EVITA_KEY_USE_FLAG_ENCRYPT 0x4
```

Definition at line 117 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.60 EVITA\_KEY\_USE\_FLAG\_REMOVE

```
#define EVITA_KEY_USE_FLAG_REMOVE 0x400
```

Definition at line 125 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.61 EVITA\_KEY\_USE\_FLAG\_SECUREBOOT

```
#define EVITA_KEY_USE_FLAG_SECUREBOOT 0x20
```

Definition at line 120 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.62 EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE

```
#define EVITA_KEY_USE_FLAG_SECURESTORAGE 0x40
```

Definition at line 121 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.63 EVITA\_KEY\_USE\_FLAG\_SIGN

```
#define EVITA_KEY_USE_FLAG_SIGN 0x1
```

Definition at line 115 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.64 EVITA\_KEY\_USE\_FLAG\_TIMESTAMP

```
#define EVITA_KEY_USE_FLAG_TIMESTAMP 0x10
```

Definition at line 119 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.65 EVITA\_KEY\_USE\_FLAG\_TRANSPORT

```
#define EVITA_KEY_USE_FLAG_TRANSPORT 0x200
```

Definition at line 124 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.66 EVITA\_KEY\_USE\_FLAG\_UTCSYNC

```
#define EVITA_KEY_USE_FLAG_UTCSYNC 0x100
```

Definition at line 123 of file eHSM\_If\_Asr\_Types\_Ip.h.

#### 4.27.1.67 EVITA\_KEY\_USE\_FLAG\_VERIFY

```
#define EVITA_KEY_USE_FLAG_VERIFY 0x2
```

Definition at line 116 of file eHSM\_If\_Asr\_Types\_Ip.h.

### 4.27.2 Typedef Documentation

#### 4.27.2.1 crypto\_copy\_key\_dh\_key\_info\_st

```
typedef struct crypto_copy_key_dh_key_info crypto_copy_key_dh_key_info_st
```

#### 4.27.2.2 crypto\_copy\_key\_info\_st

```
typedef struct crypto_copy_key_info crypto_copy_key_info_st
```

#### 4.27.2.3 crypto\_create\_evita\_key\_info\_st

```
typedef struct crypto_create_evita_key_info crypto_create_evita_key_info_st
```

#### 4.27.2.4 crypto\_evita\_key\_info\_st

```
typedef struct crypto_evita_key_info crypto_evita_key_info_st
```

#### 4.27.2.5 crypto\_exported\_key\_st

```
typedef struct crypto_exported_key crypto_exported_key_st
```

#### 4.27.2.6 crypto\_import\_evita\_key\_info\_st

```
typedef struct crypto_import_evita_key_info crypto_import_evita_key_info_st
```

#### 4.27.2.7 crypto\_key\_derive\_info\_st

```
typedef struct crypto_key_derive_info crypto_key_derive_info_st
```

#### 4.27.2.8 crypto\_key\_export\_info\_st

```
typedef struct crypto_key_export_info crypto_key_export_info_st
```

#### 4.27.2.9 crypto\_key\_status\_info\_st

```
typedef struct crypto_key_status_info crypto_key_status_info_st
```

#### 4.27.2.10 crypto\_she\_key\_st

```
typedef struct crypto_she_key crypto_she_key_st
```

#### 4.27.2.11 ehsm\_crypto\_key\_st

```
typedef struct ehsm_crypto_key ehsm_crypto_key_st
```

### 4.27.3 Enumeration Type Documentation

#### 4.27.3.1 crypto\_key\_type\_e

```
enum crypto_key_type_e
```

## Enumerator

|                                 |  |
|---------------------------------|--|
| CRYPTO_KEY_TYPE_SHE             |  |
| CRYPTO_KEY_TYPE_SHE_PLAIN       |  |
| CRYPTO_KEY_TYPE_SIGNATURE       |  |
| CRYPTO_KEY_TYPE_CIPHER          |  |
| CRYPTO_KEY_TYPE_KEY_EXCHANGE    |  |
| CRYPTO_KEY_TYPE_KEY_DERIVATION  |  |
| CRYPTO_KEY_TYPE_KEY_GENERATE    |  |
| CRYPTO_KEY_TYPE_KEY_IMPORT      |  |
| CRYPTO_KEY_TYPE_KEY_EXPORT      |  |
| CRYPTO_KEY_TYPE_KEY_REMOVE      |  |
| CRYPTO_KEY_TYPE_KEY_STATUS      |  |
| CRYPTO_KEY_TYPE_KEY_COPY        |  |
| CRYPTO_KEY_TYPE_CERTIFICATE_KEY |  |
| CRYPTO_KEY_TYPE_MAC             |  |
| CRYPTO_KEY_TYPE_INVALID         |  |

Definition at line 160 of file eHSM\_If\_Asr\_Types\_Ip.h.

## 4.27.3.2 ehsm\_key\_type\_e

enum [ehsm\\_key\\_type\\_e](#)

## Enumerator

|                       |  |
|-----------------------|--|
| EHSM_KEY_TYPE_SHE     |  |
| EHSM_KEY_TYPE_EVITA   |  |
| EHSM_KEY_TYPE_INVALID |  |

Definition at line 140 of file eHSM\_If\_Asr\_Types\_Ip.h.

## 4.27.3.3 ehsm\_key\_use\_state\_e

enum [ehsm\\_key\\_use\\_state\\_e](#)

## Enumerator

|                           |  |
|---------------------------|--|
| EHSM_KEY_USE_STATE_FREE   |  |
| EHSM_KEY_USE_STATE_IN_USE |  |

Definition at line 154 of file eHSM\_If\_Asr\_Types\_Ip.h.

## 4.27.3.4 key\_storage\_type\_e

```
enum key_storage_type_e
```

Enumerator

|                      |  |
|----------------------|--|
| KEY_STORAGE_TYPE_RAM |  |
| KEY_STORAGE_TYPE_NVM |  |
| KEY_STORAGE_END      |  |

Definition at line 147 of file eHSM\_If\_Asr\_Types\_Ip.h.

## 4.27.4 Function Documentation

## 4.27.4.1 CryIf\_CallbackNotification()

```
void CryIf_CallbackNotification (
 Crypto_JobType * job,
 Std_HsmReturnType result)
```

## 4.28 eHSM\_If\_Evita\_AsymCper\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_Mgr_Ctx_Ip.h"
```

## Functions

- [ehsm\\_uint32\\_t Sign\\_Init](#) ([ehsm\\_asym\\_alg\\_e](#) algorithm\_identifier, [ehsm\\_uint32\\_t](#) hash\_algorithm\_identifier, [padding\\_scheme\\_e](#) padding, [ehsm\\_uint32\\_t](#) total\_message\_length, [ehsm\\_bool\\_t](#) time\_stamp\_signature, [ehsm\\_uint32\\_t](#) key\_handle, [ehsm\\_uint32\\_t](#) key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*key\_authorization\_value, [ehsm\\_ctx\\_session\\_st](#) \*session\_handle)  
*EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the [Sign\\_Update\(\)](#) later.*
- [ehsm\\_uint32\\_t Sign\\_Update](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) chunk\_size, [ehsm\\_uint8\\_t](#) \*chunk\_data)  
*EVITA asymmetric cipher update, aimed to process the data in chunks.*
- [ehsm\\_uint32\\_t Sign\\_Finish](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [signature\\_st](#) \*signature)  
*EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.*

- `ehsm_uint32_t Verify_Init` (`ehsm_asym_alg_e` algorithm\_identifier, `ehsm_uint32_t` hash\_algorithm\_identifier, `padding_scheme_e` padding, `ehsm_uint32_t` total\_message\_length, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization\_value, `signature_st` \*signature, `ehsm_ctx_session_st` \*session\_handle)  
*EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the [Verify\\_Update\(\)](#) later.*
- `ehsm_uint32_t Verify_Update` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` chunk\_size, `ehsm_uint8_t` \*chunk\_data)  
*EVITA asymmetric cipher update, aimed to process the data in chunks.*
- `ehsm_uint32_t Verify_Finish` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_utc_time_t` \*utc\_time\_stamp, `ehsm_bool_t` \*sign\_match)  
*EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.*

## 4.28.1 Function Documentation

### 4.28.1.1 Sign\_Finish()

```
ehsm_uint32_t Sign_Finish (
 ehsm_ctx_session_st * session_handle,
 signature_st * signature)
```

EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.

#### Parameters

|     |                       |                                                    |
|-----|-----------------------|----------------------------------------------------|
| in  | <i>session_handle</i> | Session reference from <a href="#">Sign_Init()</a> |
| out | <i>signature</i>      | Pointer to signature.                              |

#### Returns

`ehsm_uint32_t`

#### Return values

|                                                   |                                      |
|---------------------------------------------------|--------------------------------------|
| <i><a href="#">EVITA_OK</a>(value)</i>            | 0) For success.                      |
| <i><a href="#">EVITA_WRONG_SESSION_HANDLE</a></i> | Given key handle is unknown or wrong |

Definition at line 373 of file `eHSM_If_Evita_AsymCper_Ip.c`.

### 4.28.1.2 Sign\_Init()

```
ehsm_uint32_t Sign_Init (
 ehsm_asym_alg_e algorithm_identifier,
 ehsm_uint32_t hash_algorithm_identifier,
 padding_scheme_e padding,
 ehsm_uint32_t total_message_length,
 ehsm_bool_t time_stamp_signature,
```

```
ehsm_uint32_t key_handle,
ehsm_uint32_t key_authorization_size,
ehsm_uint8_t * key_authorization_value,
ehsm_ctx_session_st * session_handle)
```

EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the [Sign\\_Update\(\)](#) later.

#### Parameters

|     |                                            |                                                                           |
|-----|--------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                | Reference to associated asymmetric algorithm.                             |
| in  | <i>hash_algorithm_identifier</i>           | Indicate underlying hash algorithm.                                       |
| in  | <i>padding</i>                             | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>                | Give total message length(can be req. by padding scheme).                 |
| in  | <i>key_handle</i>                          | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>              | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>             | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle</i>                      | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i>   | Maxium size of a chunk on AsymCipher_Process().                           |
| out | <i>session_handle-&gt;chunk_block_size</i> | chunk has to be a multiple of this block size(1 to max).                  |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                    |                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>    | 0) For success.                                                                  |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional session                                      |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                  |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                              |

Definition at line 194 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

#### 4.28.1.3 Sign\_Update()

```
ehsm_uint32_t Sign_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA asymmetric cipher update, aimed to process the data in chunks.

#### Parameters

|    |                       |                                                    |
|----|-----------------------|----------------------------------------------------|
| in | <i>session_handle</i> | Session reference from <a href="#">Sign_Init()</a> |
| in | <i>chunk_size</i>     | Size of chunk data.                                |
| in | <i>chunk_data</i>     | Pointer to chunk data.                             |



## Returns

ehsm\_uint32\_t

## Return values

|                                            |                                                           |
|--------------------------------------------|-----------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                                           |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given key handle is unknown or wrong                      |
| <a href="#">EVITA_WRONG_CHUNK_SIZE</a>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 320 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.28.1.4 Verify\_Finish()

```
ehsm_uint32_t Verify_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_utc_time_t * utc_time_stamp,
 ehsm_bool_t * sign_match)
```

EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.

## Parameters

|     |                       |                                                      |
|-----|-----------------------|------------------------------------------------------|
| in  | <i>session_handle</i> | Session reference from <a href="#">Verify_Init()</a> |
| out | <i>utc_time_stamp</i> | Verified UTC time stamp(if available)                |
| out | <i>sign_match</i>     | Indicate whether the signature match or not.         |

## Returns

ehsm\_uint32\_t

## Return values

|                                            |                                      |
|--------------------------------------------|--------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                      |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given key handle is unknown or wrong |

Definition at line 640 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.28.1.5 Verify\_Init()

```
ehsm_uint32_t Verify_Init (
 ehsm_asym_alg_e algorithm_identifier,
 ehsm_uint32_t hash_algorithm_identifier,
 padding_scheme_e padding,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
```

```
ehsm_uint8_t * key_authorization_value,
signature_st * signiture,
ehsm_ctx_session_st * session_handle)
```

EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the [Verify\\_Update\(\)](#) later.

#### Parameters

|     |                                            |                                                                           |
|-----|--------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                | Reference to associated asymmetric algorithm.                             |
| in  | <i>hash_algorithm_identifier</i>           | Indicate underlying hash algorithm.                                       |
| in  | <i>padding</i>                             | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>                | Give total message length(can be req. by padding scheme).                 |
| in  | <i>key_handle</i>                          | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>              | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>             | Key usage authorization(i.e., password).                                  |
| in  | <i>signature</i>                           | Pointer to signature.                                                     |
| out | <i>session_handle</i>                      | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i>   | Maxium size of a chunk on AsymCipher_Process().                           |
| out | <i>session_handle-&gt;chunk_block_size</i> | chunk has to be a multiple of this block size(1 to max).                  |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                             |                                                                                  |
|---------------------------------------------|----------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>             | 0) For success.                                                                  |
| <a href="#">EVITA_WRONG_KEY_HANDLE</a>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <a href="#">EVITA_ALL_SESSIONS_OCCUPIED</a> | No resources left for an additional session                                      |
| <a href="#">EVITA_ALGORITHM_ERROR</a>       | Given algorithm or algorithm mode not available                                  |
| <a href="#">EVITA_AUTHORIZATION_FAILE</a>   | Given authorization value was wrong                                              |

Definition at line 455 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

#### 4.28.1.6 Verify\_Update()

```
ehsm_uint32_t Verify_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA asymmetric cipher update, aimed to process the data in chunks.

#### Parameters

|    |                       |                                                      |
|----|-----------------------|------------------------------------------------------|
| in | <i>session_handle</i> | Session reference from <a href="#">Verify_Init()</a> |
| in | <i>chunk_size</i>     | Size of chunk data.                                  |
| in | <i>chunk_data</i>     | Pointer to chunk data.                               |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                                           |
|----------------------------------------|-----------------------------------------------------------|
| <a href="#"><i>EVITA_OK</i>(value)</a> | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i>      | Given key handle is unknown or wrong                      |
| <i>EVITA_WRONG_CHUNK_SIZE</i>          | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 586 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.29 eHSM\_If\_Evita\_Counter\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
```

## 4.30 eHSM\_If\_Evita\_ErrCode\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_Debug_Ip.h"
```

### Functions

- [`ehsm\_uint32\_t ehsm\_evita\_convert\_ret\_code`](#) ([`ehsm\_uint32\_t`](#) ret)  
*convert a eHSM error code into EVITA error code*
- [`ehsm\_uint32\_t Evita\_Check\_Key\_Handle`](#) ([`ehsm\_uint32\_t`](#) key\_handle)  
*Check the key handle in EVITA interface.*
- [`ehsm\_uint32\_t Evita\_Check\_Authorization\_Code`](#) ([`ehsm\_uint32\_t`](#) authorization\_size, [`ehsm\_uint8\_t`](#) \*authorization↔\_value)  
*Check the authorization code in EVITA interface.*

### 4.30.1 Function Documentation

#### 4.30.1.1 ehsm\_evita\_convert\_ret\_code()

```
ehsm_uint32_t ehsm_evita_convert_ret_code (
 ehsm_uint32_t ret)
```

convert a eHSM error code into EVITA error code

## Parameters

|    |     |                                                                                |
|----|-----|--------------------------------------------------------------------------------|
| in | ret | error code of eHSM, refers to error code in <a href="#">eHSM_Err_Code_Ip.h</a> |
|----|-----|--------------------------------------------------------------------------------|

## Returns

EVITA error code, refers to error code in [eHSM\\_If\\_Evita\\_Types\\_Ip.h](#)

Definition at line 52 of file eHSM\_If\_Evita\_ErrCode\_Ip.c.

## 4.30.1.2 Evita\_Check\_Authorization\_Code()

```
ehsm_uint32_t Evita_Check_Authorization_Code (
 ehsm_uint32_t authorization_size,
 ehsm_uint8_t * authorization_value)
```

Check the authorization code in EVITA interface.

## Parameters

|    |                     |  |
|----|---------------------|--|
| in | authorization_size  |  |
| in | authorization_value |  |

## Returns

ehsm\_uint32\_t

## Return values

|                            |  |
|----------------------------|--|
| EVITA_OK                   |  |
| EVITA_AUTHORIZATION_FAILED |  |

## Note

Definition at line 163 of file eHSM\_If\_Evita\_ErrCode\_Ip.c.

## 4.30.1.3 Evita\_Check\_Key\_Handle()

```
ehsm_uint32_t Evita_Check_Key_Handle (
 ehsm_uint32_t key_handle)
```

Check the key handle in EVITA interface.

## Parameters

|    |            |  |
|----|------------|--|
| in | key_handle |  |
|----|------------|--|

## Returns

ehsm\_uint32\_t

## Return values

|                        |  |
|------------------------|--|
| EVITA_OK               |  |
| EVITA_WRONG_KEY_HANDLE |  |

## Note

Definition at line 149 of file eHSM\_If\_Evita\_ErrCode\_Ip.c.

## 4.31 eHSM\_If\_Evita\_ErrCode\_Ip.h File Reference

```
#include "eHSM_Types_Ip.h"
```

### Functions

- [ehsm\\_uint32\\_t ehsm\\_evita\\_convert\\_ret\\_code](#) (ehsm\_uint32\_t ret)  
*convert a eHSM error code into EVITA error code*
- [ehsm\\_uint32\\_t Evita\\_Check\\_Key\\_Handle](#) (ehsm\_uint32\_t key\_handle)  
*Check the key handle in EVITA interface.*
- [ehsm\\_uint32\\_t Evita\\_Check\\_Authorization\\_Code](#) (ehsm\_uint32\_t authorization\_size, ehsm\_uint8\_t \*authorization↔\_value)  
*Check the authorization code in EVITA interface.*

#### 4.31.1 Function Documentation

##### 4.31.1.1 ehsm\_evita\_convert\_ret\_code()

```
ehsm_uint32_t ehsm_evita_convert_ret_code (
 ehsm_uint32_t ret)
```

convert a eHSM error code into EVITA error code

## Parameters

|    |     |                                                                                |
|----|-----|--------------------------------------------------------------------------------|
| in | ret | error code of eHSM, refers to error code in <a href="#">eHSM_Err_Code_Ip.h</a> |
|----|-----|--------------------------------------------------------------------------------|

## Returns

EVITA error code, refers to error code in [eHSM\\_If\\_Evita\\_Types\\_Ip.h](#)

Definition at line 52 of file eHSM\_If\_Evita\_ErrCode\_Ip.c.

## 4.31.1.2 Evita\_Check\_Authorization\_Code()

```
ehsm_uint32_t Evita_Check_Authorization_Code (
 ehsm_uint32_t authorization_size,
 ehsm_uint8_t * authorization_value)
```

Check the authorization code in EVITA interface.

## Parameters

|    |                     |  |
|----|---------------------|--|
| in | authorization_size  |  |
| in | authorization_value |  |

## Returns

ehsm\_uint32\_t

## Return values

|                            |  |
|----------------------------|--|
| EVITA_OK                   |  |
| EVITA_AUTHORIZATION_FAILED |  |

## Note

Definition at line 163 of file eHSM\_If\_Evita\_ErrCode\_Ip.c.

## 4.31.1.3 Evita\_Check\_Key\_Handle()

```
ehsm_uint32_t Evita_Check_Key_Handle (
 ehsm_uint32_t key_handle)
```

Check the key handle in EVITA interface.

## Parameters

|    |            |  |
|----|------------|--|
| in | key_handle |  |
|----|------------|--|

## Returns

ehsm\_uint32\_t

## Return values

|                        |  |
|------------------------|--|
| EVITA_OK               |  |
| EVITA_WRONG_KEY_HANDLE |  |

## Note

Definition at line 149 of file eHSM\_If\_Evita\_ErrCode\_Ip.c.

## 4.32 eHSM\_If\_Evita\_Hash\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Debug_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_Mgr_Ctx_Ip.h"
```

## Macros

- #define [HASH\\_SIZE](#)
- #define [X](#)(hash\_type, block\_sz, sz)
- #define [X](#)(hash\_type, block\_sz, sz)

## Functions

- [ehsm\\_uint32\\_t Hash\\_Init](#) ([ehsm\\_uint32\\_t](#) algorithm\_identifier, [hash\\_mode\\_e](#) hash\_mode, [ehsm\\_uint32\\_t](#) key↔\_handle, [ehsm\\_uint32\\_t](#) key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*key\_authorization\_value, [ehsm\\_ctx\\_session↔\\_st](#) \*session\_handle)  
*EVITA Hash initialization, aimed to make all necessary preparations for the calculation of hash values and HMACs or verification of HMACs belonging to longer messages that cannot be processed within one function call.*
- [ehsm\\_uint32\\_t Hash\\_Update](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) chunk\_size, [ehsm\\_uint8\\_t](#)↔ \*chunk\_data)  
*EVITA Hash update, aimed to process one or several blocks(up to the maximum allowed chunk size delivered by Hash↔\_Init)within a hash or HMAC calculation or HMAC verification for longer messages.*
- [ehsm\\_uint32\\_t Hash\\_Finish](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [hash\\_hmac\\_st](#) \*hash\_hmac, [ehsm\\_bool\\_t](#)↔ \*hmac\_match)  
*EVITA Hash finish, aimed to terminate the hash or HMAC calculation or HMAC verification after processing the last message block.*

### 4.32.1 Macro Definition Documentation

#### 4.32.1.1 HASH\_SIZE

```
#define HASH_SIZE
```

**Value:**

```
X(EHSM_SM3, 64U, 32U) \
 X(EHSM_MD5, 64U, 16U) \
 X(EHSM_SHA1, 64U, 20U) \
 X(EHSM_SHA256, 64U, 32U) \
 X(EHSM_SHA224, 64U, 28U) \
 X(EHSM_SHA384, 128U, 48U) \
 X(EHSM_SHA512, 128U, 64U) \
 X(EHSM_SHA512_224, 128U, 28U) \
 X(EHSM_SHA512_256, 128U, 32U) \
 X(EHSM_SHA3_224, 144U, 28U) \
 X(EHSM_SHA3_256, 136U, 32U) \
 X(EHSM_SHA3_384, 104U, 48U) \
 X(EHSM_SHA3_512, 72U, 64U) \
 X(EHSM_INVALID_ALG, 0U, 0U)
```

Definition at line 26 of file eHSM\_If\_Evita\_Hash\_Ip.c.

#### 4.32.1.2 X [1/2]

```
#define X(
 hash_type,
 block_sz,
 sz)
```

**Value:**

```
case hash_type:
{
 msg_block_sz = block_sz;
 break;
}
```

#### 4.32.1.3 X [2/2]

```
#define X(
 hash_type,
 block_sz,
 sz)
```

**Value:**

```
case hash_type:
{
 ret = EVITA_OK;
 break;
}
```



### 4.32.2 Function Documentation

#### 4.32.2.1 Hash\_Finish()

```
ehsm_uint32_t Hash_Finish (
 ehsm_ctx_session_st * session_handle,
 hash_hmac_st * hash_hmac,
 ehsm_bool_t * hmac_match)
```

EVITA Hash finish, aimed to terminate the hash or HMAC calculation or HMAC verification after processing the last message block.

##### Parameters

|     |                       |                                                                                                             |
|-----|-----------------------|-------------------------------------------------------------------------------------------------------------|
| in  | <i>session_handle</i> | Session handle to be released.                                                                              |
|     | <i>[in/out]</i>       | hash_hmac Data structure include HASH/HMAC data and size(optional with UTC_time_stamp in HMAC verify mode). |
| out | <i>hmac_match</i>     | True if calculated HMAC matches the given Reference HMAC(only for HMAC verify).                             |

##### Returns

ehsm\_uint32\_t

##### Return values

|                                   |                                      |
|-----------------------------------|--------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                      |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given key handle is unknown or wrong |

Definition at line 321 of file eHSM\_If\_Evita\_Hash\_Ip.c.

#### 4.32.2.2 Hash\_Init()

```
ehsm_uint32_t Hash_Init (
 ehsm_uint32_t algorithm_identifier,
 hash_mode_e hash_mode,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA Hash initialization, aimed to make all necessary preparations for the calculation of hash values and HMACs or verification of HMACs belonging to longer messages that cannot be processed within one function call.

##### Parameters

|    |                             |                                                      |
|----|-----------------------------|------------------------------------------------------|
| in | <i>algorithm_identifier</i> | Reference to associated hash algorithm.              |
| in | <i>hash_mode</i>            | Indicate hash or HMAC creation or verification mode. |

## Parameters

|     |                                            |                                                                                               |
|-----|--------------------------------------------|-----------------------------------------------------------------------------------------------|
| in  | <i>key_handle</i>                          | Refer to internal key that will be used(only for HMAC, set to 0 otherwise).                   |
| in  | <i>key_authorization_size</i>              | Size of HAMC key usage authorization value(0 for none).                                       |
| in  | <i>key_authorization_value</i>             | HMAC key usage authorization(i.e., password).                                                 |
| out | <i>session_handle</i>                      | Enables interruption & parallel processing and/or session authentication.                     |
| out | <i>session_handle-&gt;max_chunk_size</i>   | Maxium chunk size possible on update() (note that hash function have inherent padding scheme) |
| out | <i>session_handle-&gt;chunk_block_size</i> | Chunk has to be a multiple of this block size(1 to max).                                      |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------|
| <i>EVITA_OK(value)</i>             | 0) For success.                                                                  |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional session                                      |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                  |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                              |

Definition at line 142 of file eHSM\_If\_Evita\_Hash\_Ip.c.

## 4.32.2.3 Hash\_Update()

```
ehsm_uint32_t Hash_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA Hash update, aimed to process one or several blocks(up to the maximum allowed chunk size delivered by Hash\_↵ Init)within a hash or HAMC calculation or HMAC verification for longer messages.

## Parameters

|    |                       |                                                                 |
|----|-----------------------|-----------------------------------------------------------------|
| in | <i>session_handle</i> | Session reference from init() to enable parallel hash sessions. |
| in | <i>chunk_size</i>     | Size of chunk used for hash update.                             |
| in | <i>chunk_data</i>     | Data chunk byte array used for hash update.                     |

## Returns

ehsm\_uint32\_t

## Return values

|                        |                 |
|------------------------|-----------------|
| <i>EVITA_OK(value)</i> | 0) For success. |
|------------------------|-----------------|

## Return values

|                                         |                                                           |
|-----------------------------------------|-----------------------------------------------------------|
| <code>EVITA_WRONG_SESSION_HANDLE</code> | Given key handle is unknown or wrong                      |
| <code>EVITA_WRONG_CHUNK_SIZE</code>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 260 of file eHSM\_If\_Evita\_Hash\_Ip.c.

## 4.33 eHSM\_If\_Evita\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_Mgr_Ctx_Ip.h"
```

### Functions

- ehsm\_uint32\_t Create\_Random\_Key** (ehsm\_uint32\_t target\_algorithm, ehsm\_uint32\_t key\_size, ehsm\_key\_mem↔\_type\_e type, ehsm\_uint32\_t key\_usages\_size, ehsm\_uint8\_t \*key\_usages\_data, ehsm\_uint32\_t \*key\_handle)  
*Create a random key according to the specified algorithm.*
- ehsm\_uint32\_t Create\_Dh\_Key** (ehsm\_uint32\_t target\_algorithm, ehsm\_uint32\_t key\_size, ehsm\_key\_mem↔\_type\_e type, ehsm\_uint32\_t key\_usages\_size, ehsm\_uint8\_t \*key\_usages\_data, ehsm\_uint32\_t local\_key\_handle, ehsm\_uint32\_t local\_key\_auth\_size, ehsm\_uint8\_t \*local\_key\_auth\_value, ehsm\_uint32\_t remote\_key\_handle, ehsm\_uint32\_t remote\_key\_auth\_size, ehsm\_uint8\_t \*remote\_key\_auth\_value, ehsm\_uint32\_t \*key\_handle)  
*Create a Diffie-Hellman key according to the specified algorithm.*
- ehsm\_uint32\_t Key\_Import** (ehsm\_uint32\_t transport\_key\_handle, ehsm\_uint32\_t transport\_key\_authorization↔\_size, ehsm\_uint8\_t \*transport\_key\_authorization, ehsm\_uint32\_t authenticity\_key\_handle, ehsm\_uint32\_t↔ authenticity\_key\_authorization\_size, ehsm\_uint8\_t \*authenticity\_key\_authorization, ehsm\_key\_mem\_type\_e type, ehsm\_uint32\_t encrypted\_key\_size, ehsm\_uint8\_t \*encrypted\_key, ehsm\_uint32\_t key\_authenticity\_code\_size, ehsm\_uint8\_t \*key\_authenticity\_code, ehsm\_uint32\_t \*key\_handle)  
*This function is used for importing keys into eHSM that were generated and exported by another EVITA module or another external trusted party.*
- ehsm\_uint32\_t Key\_Export** (ehsm\_uint32\_t key\_handle, key\_act\_use\_flags\_t \*use\_flags, ehsm\_uint32\_t↔ transport\_key\_handle, ehsm\_uint32\_t transport\_key\_authorization\_size, ehsm\_uint8\_t \*transport\_key↔ authorization, ehsm\_uint32\_t authenticity\_key\_handle, ehsm\_uint32\_t authenticity\_key\_authorization\_size, ehsm↔\_uint8\_t \*authenticity\_key\_authorization, ehsm\_uint32\_t \*encrypted\_key\_size, ehsm\_uint8\_t \*encrypted\_key, ehsm\_uint32\_t \*key\_authenticity\_code\_size, ehsm\_uint8\_t \*key\_authenticity\_code)  
*This function is used for exporting keys from eHSM.*
- ehsm\_uint32\_t Create\_Derived\_Key** (ehsm\_uint32\_t key\_derivation\_function\_identifier, ehsm\_uint32\_t key\_size, ehsm\_key\_mem\_type\_e type, ehsm\_uint32\_t key\_usages\_size, ehsm\_uint8\_t \*key\_usages\_data, ehsm\_uint32\_t parent\_key\_handle, ehsm\_uint32\_t parent\_key\_authorization\_size, ehsm\_uint8\_t \*parent\_key\_authorization\_value, ehsm\_uint32\_t salt\_size, ehsm\_uint8\_t \*salt\_data, ehsm\_uint32\_t \*key\_handle)  
*This function creates a key in a similar way as an RNG-based key generation, but with a symmetric parent key as a base.*
- ehsm\_uint32\_t Key\_Remove** (ehsm\_uint32\_t key\_handle, ehsm\_uint32\_t key\_authorization\_size, ehsm\_uint8\_t↔ \*key\_authorization)  
*This function is used for removing loaded keys from the eHSM.*
- ehsm\_uint32\_t Key\_Status** (ehsm\_uint32\_t key\_handle, ehsm\_uint32\_t certification\_key\_handle, ehsm\_uint32\_t↔ certification\_key\_authorization\_size, ehsm\_uint8\_t \*certification\_key\_authorization, ehsm\_uint32\_t \*key\_status↔\_size, ehsm\_uint8\_t \*key\_status)  
*This function is used for obtaining all public properties about a key loaded by the eHSM.*

- `ehsm_uint32_t Create_Random_Dh_Key_Pair` (`ehsm_uint32_t` key\_size, `ehsm_uint32_t` valid\_until, `ehsm_key_↵` mem\_type\_e type, `ehsm_uint32_t` key\_usages\_size, `ehsm_uint8_t` \*key\_usages\_data, `ehsm_uint8_t` \*p, `ehsm_↵` uint32\_t p\_size, `ehsm_uint8_t` \*q, `ehsm_uint32_t` q\_size, `ehsm_uint8_t` \*g, `ehsm_uint32_t` g\_size, `ehsm_uint32_t` \*key\_handle)

*This function is used for creating a DH key pair.*

- `ehsm_uint32_t RNG_Get_Random` (`ehsm_uint32_t` algorithm\_identifier, `ehsm_uint32_t` random\_byte\_request\_size, `ehsm_uint8_t` \*random\_bytes)

*Generate the random number.*

- `ehsm_uint32_t Cipher_Init` (`ehsm_uint32_t` algorithm\_identifier, `cipher_mode_e` cipher\_mode, `operation_mode_↵` \_e operation\_mode, `padding_scheme_e` padding, `ehsm_uint32_t` total\_message\_length, `ehsm_uint32_t` iv\_↵ size, `ehsm_uint8_t` \*iv, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_↵ \_authorization\_value, `ehsm_ctx_session_st` \*session\_handle)

*EVITA cipher initialization, aimed to prepare all the data that used in the `Cipher_Process()` later.*

- `ehsm_uint32_t Cipher_Process` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` input\_data\_size, const `ehsm_uint8_t` \*input\_data, `ehsm_uint32_t` \*output\_data\_size, `ehsm_uint8_t` \*output\_data)

*EVITA cipher process, aimed to process one or more block within the encryption and decryption process of longer messages.*

- `ehsm_uint32_t Cipher_Finish` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` \*output\_data\_size, `ehsm_↵` uint8\_t \*output\_data)

*EVITA cipher finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by `Cipher_Process()`.*

- `ehsm_uint32_t Aead_Init` (`ehsm_uint32_t` algorithm\_identifier, `cipher_mode_e` cipher\_mode, `operation_mode_↵` e operation\_mode, `ehsm_uint32_t` total\_message\_length, `ehsm_uint32_t` iv\_size, `ehsm_uint8_t` \*iv, `ehsm_uint32_↵` \_t aad\_size, `ehsm_uint32_t` tag\_size, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_↵` uint8\_t \*key\_authorization\_value, `ehsm_ctx_session_st` \*session\_handle)

*EVITA aead initialization, aimed to prepare all the data that used in the `Aead_Process()` later.*

- `ehsm_uint32_t Aead_Process` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` input\_data\_size, const `ehsm_uint8_t` \*input\_data, `ehsm_uint32_t` aad\_size, const `ehsm_uint8_t` \*aad, `ehsm_uint32_t` \*output\_data\_size, `ehsm_uint8_t` \*output\_data)

*EVITA aead process, aimed to process one or more block within the encryption and decryption process of longer messages.*

- `ehsm_uint32_t Aead_Finish` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` \*output\_data\_size, `ehsm_↵` uint8\_t \*output\_data, `ehsm_uint32_t` \*match)

*EVITA aead finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by `Aead_Process()`.*

- `ehsm_uint32_t MAC_Init` (`ehsm_uint32_t` algorithm\_identifier, `mac_mode_e` mac\_mode, `operation_mode_↵` \_e operation\_mode, `padding_scheme_e` padding\_scheme, `ehsm_uint32_t` total\_message\_length, `ehsm_↵` uint32\_t mac\_length, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_↵ authorization\_value, `ehsm_ctx_session_st` \*session\_handle)

*EVITA MAC initialization, aimed to prepare all the data that used in the `MAC_Update()` later.*

- `ehsm_uint32_t MAC_Update` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` chunk\_size, `ehsm_uint8_↵` t \*chunk\_data)

*EVITA MAC update, aimed to process one or more block within a MAC generation or verification process of longer messages.*

- `ehsm_uint32_t MAC_Finish` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` \*mac\_size, `mac_st` \*mac, `ehsm_bool_t` \*mac\_match)

*EVITA MAC finish, aimed to terminate the MAC generation or verification process after the last block has been processed by `MAC_Update`.*

- `ehsm_uint32_t Hash_Init` (`ehsm_uint32_t` algorithm\_identifier, `hash_mode_e` hash\_mode, `ehsm_uint32_t` key\_↵ \_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization\_value, `ehsm_ctx_session_↵` st \*session\_handle)

*EVITA Hash initialization, aimed to make all necessary preparations for the calculation of hash values and HMACs or verification of HMACs belonging to longer messages that cannot be processed within one function call.*

- `ehsm_uint32_t Hash_Update` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` chunk\_size, `ehsm_uint8_↵` t \*chunk\_data)

*EVITA Hash update, aimed to process one or several blocks (up to the maximum allowed chunk size delivered by `Hash_↵` Init) within a hash or HMAC calculation or HMAC verification for longer messages.*

- `ehsm_uint32_t Hash_Finish` (`ehsm_ctx_session_st` \*session\_handle, `hash_hmac_st` \*hash\_hmac, `ehsm_bool_↵` t \*hmac\_match)

*EVITA Hash finish, aimed to terminate the hash or HMAC calculation or HMAC verification after processing the last message block.*

- `ehsm_uint32_t Sign_Init` (`ehsm_asym_alg_e` algorithm\_identifier, `ehsm_uint32_t` hash\_algorithm\_identifier, `padding_scheme_e` padding, `ehsm_uint32_t` total\_message\_length, `ehsm_bool_t` time\_stamp\_signature, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization\_value, `ehsm_ctx_session_st` \*session\_handle)

*EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the `Sign_Update()` later.*

- `ehsm_uint32_t Sign_Update` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` chunk\_size, `ehsm_uint8_t` \*chunk\_data)

*EVITA asymmetric cipher update, aimed to process the data in chunks.*

- `ehsm_uint32_t Sign_Finish` (`ehsm_ctx_session_st` \*session\_handle, `signature_st` \*signature)

*EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.*

- `ehsm_uint32_t Verify_Init` (`ehsm_asym_alg_e` algorithm\_identifier, `ehsm_uint32_t` hash\_algorithm\_identifier, `padding_scheme_e` padding, `ehsm_uint32_t` total\_message\_length, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization\_value, `signature_st` \*signature, `ehsm_ctx_session_st` \*session\_handle)

*EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the `Verify_Update()` later.*

- `ehsm_uint32_t Verify_Update` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_uint32_t` chunk\_size, `ehsm_uint8_t` \*chunk\_data)

*EVITA asymmetric cipher update, aimed to process the data in chunks.*

- `ehsm_uint32_t Verify_Finish` (`ehsm_ctx_session_st` \*session\_handle, `ehsm_utc_time_t` \*utc\_time\_stamp, `ehsm_bool_t` \*sign\_match)

*EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.*

- `ehsm_uint32_t Create_Counter` (`ehsm_uint32_t` access\_authorization\_size, const `ehsm_uint8_t` \*access\_authorization\_value, `ehsm_uint32_t` \*counter\_identifier, `ehsm_counter_value_st` \*counter\_initial\_value)

*This function is used for the initial creation of a counter.*

- `ehsm_uint32_t Read_Counter` (`ehsm_uint32_t` counter\_identifier, `ehsm_counter_value_st` \*counter\_current\_value)

*This function is used to read the value of a counter.*

- `ehsm_uint32_t Increment_Counter` (`ehsm_uint32_t` counter\_identifier, `ehsm_uint32_t` access\_authorization\_size, `ehsm_uint8_t` \*access\_authorization\_value, const `ehsm_counter_value_st` \*counter\_incrementation, `ehsm_counter_value_st` \*counter\_new\_value)

*This function is used to increment an existing counter.*

- `ehsm_uint32_t Delete_Counter` (`ehsm_uint32_t` counter\_identifier, `ehsm_uint32_t` access\_authorization\_size, `ehsm_uint8_t` \*access\_authorization\_value)

*With this function, a previously created counter is deleted again.*

- `ehsm_uint32_t Create_Time_Stamp` (`ehsm_uint32_t` msg\_imprint\_size, const `ehsm_uint8_t` \*msg\_imprint, `ehsm_uint32_t` signature\_key\_handle, `ehsm_uint32_t` signature\_key\_authorization\_size, `ehsm_uint8_t` \*signature\_key\_authorization\_value, `ehsm_uint32_t` \*signature\_size, `signature_st` \*signature)

*This function allows to sign arbitrary data and to include a time stamp (in form of UTC) into the signature.*

- `ehsm_uint32_t Check_Time_Stamp` (`ehsm_uint32_t` msg\_imprint\_size, const `ehsm_uint8_t` \*msg\_imprint, `ehsm_uint32_t` verification\_key\_handle, `ehsm_uint32_t` verification\_key\_authorization\_size, `ehsm_uint8_t` \*verification\_key\_authorization\_value, const `signature_st` \*time\_stamp, `ehsm_bool_t` \*time\_stamp\_vry, `ehsm_uint32_t` \*delta)

*If a time stamp has been created by the `Create_Time_Stamp` function described before, the `Check_Time_Stamp` function may be used to check if the time stamp is valid.*

- `ehsm_uint32_t Get_Time_Sync_Challenge` (`ehsm_uint32_t` \*time\_sync\_challenge\_size, `ehsm_uint8_t` time\_sync\_challenge[])

*This function is used to obtain a challenge for checking the freshness of external time synchronization.*

- `ehsm_uint32_t Set_UTC_Time` (`ehsm_utc_time_t` utc\_time, `ehsm_uint32_t` signature\_size, const `ehsm_uint8_t` \*signature, `ehsm_uint32_t` verification\_key\_handle, `ehsm_uint32_t` verification\_key\_authorization\_size, `ehsm_uint8_t` \*verification\_key\_authorization\_value)

*This function is used for setting the UTC time of the EVITA module.*

- `ehsm_uint32_t Get_UTC_Time` (`ehsm_utc_time_t` \*utc\_time)

*With this function, the UTC time value is obtained from the EVITA module.*

- `ehsm_uint32_t Get_Tick_Count` (`ehsm_tick_value_st` \*tick\_value)

*This function is used for obtaining the actual value of the EVITA tick counter.*

- `ehsm_uint32_t Module_Status` (`ehsm_uint32_t` flag, `ehsm_uint32_t` alg, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_auth\_size, const `ehsm_uint8_t` \*key\_auth\_value, `ehsm_uint32_t` \*status\_size, `ehsm_uint8_t` \*status, `ehsm_uint32_t` \*sign\_size, `ehsm_uint8_t` \*sign)

*Get the status of eHSM.*

- `ehsm_uint32_t ehsm_self_test` (`ehsm_uint32_t` flag)

*The self test service.*

- `ehsm_uint32_t ehsm_get_pub_from_priv` (`ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization, `ehsm_uint8_t` \*pub\_value, `ehsm_uint32_t` \*pub\_size, `ehsm_uint32_t` priv\_key\_algo)

*The function use to get key public party that calculation from private party.*

### 4.33.1 Function Documentation

#### 4.33.1.1 Aead\_Finish()

```
ehsm_uint32_t Aead_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data,
 ehsm_uint32_t * match)
```

EVITA aead finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by [Aead\\_Process\(\)](#).

##### Parameters

|     |                         |                                                                                                                |
|-----|-------------------------|----------------------------------------------------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session handle to be released.                                                                                 |
| out | <i>output_data_size</i> | Size of last output data(can be 0).                                                                            |
| out | <i>output_data</i>      | Last encrypted or decrypted output data(e.g., due to padding scheme).                                          |
|     | <i>[in/out]</i>         | match Only valid for aead verification. If verification success, match is 1. Otherwise, this value is invalid. |

##### Returns

`ehsm_uint32_t`

##### Return values

|                                 |                         |
|---------------------------------|-------------------------|
| <a href="#">EVITA_OK(value)</a> | 0) For success. And the |
|---------------------------------|-------------------------|

##### Parameters

|              |                                  |
|--------------|----------------------------------|
| <i>match</i> | will be updated with the result. |
|--------------|----------------------------------|

##### Return values

|                                            |                                          |
|--------------------------------------------|------------------------------------------|
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given session handle is unknown or wrong |
|--------------------------------------------|------------------------------------------|

Definition at line 866 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

#### 4.33.1.2 Aead\_Init()

```
ehsm_uint32_t Aead_Init (
 ehsm_uint32_t algorithm_identifier,
 cipher_mode_e cipher_mode,
 operation_mode_e operation_mode,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t iv_size,
 ehsm_uint8_t * iv,
 ehsm_uint32_t aad_size,
 ehsm_uint32_t tag_size,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA aead initialization, aimed to prepare all the data that used in the [Aead\\_Process\(\)](#) later.

##### Parameters

|     |                                          |                                                                           |
|-----|------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>              | Reference to associated symmetric algorithm.                              |
| in  | <i>cipher_mode</i>                       | Indicate decryption pr encryption mode.                                   |
| in  | <i>operation_mode</i>                    | Indicate cipher mode of operation.                                        |
| in  | <i>total_message_length</i>              | Give total message length(can be req. by padding scheme).                 |
| in  | <i>iv_size</i>                           | Size of given initialization vector(can be 0).                            |
| in  | <i>iv</i>                                | Set initialization vector(it's public).                                   |
| in  | <i>key_handle</i>                        | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>            | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>           | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle-&gt;session_id</i>     | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i> | Maxium size of a chunk on <a href="#">Cipher_Process()</a> .              |
| out | <i>session_handle-&gt;block_sz</i>       | chunk has to be a multiple of this block size(1 to max).                  |

##### Returns

ehsm\_uint32\_t

##### Return values

|                                    |                                                                                         |
|------------------------------------|-----------------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>    | 0) For success.                                                                         |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode)        |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional parallel session (or no parallel processing at all) |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                         |
| <i>EVITA_WRONG_IV</i>              | Given IV does not fit the given algorithm                                               |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                                     |

**Note**

iv\_size should be 12 (96 bits)

Definition at line 642 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

**4.33.1.3 Aead\_Process()**

```
ehsm_uint32_t Aead_Process (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t input_data_size,
 const ehsm_uint8_t * input_data,
 ehsm_uint32_t aad_size,
 const ehsm_uint8_t * aad,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data)
```

EVITA aead process, aimed to process one or more block within the encryption and decryption process of longer messages.

**Parameters**

|     |                         |                                                                                |
|-----|-------------------------|--------------------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session reference from cipher_Init().                                          |
| in  | <i>input_data_size</i>  | Size of input data.                                                            |
| in  | <i>input_data</i>       | Input data to be encrypted or decrypted.                                       |
| in  | <i>aad_size</i>         | Input aad size. Can be 0.                                                      |
| in  | <i>aad</i>              | Input aad. Can be null.                                                        |
| out | <i>output_data_size</i> | Size of output data(can be different to input size due to padding/de-padding). |
| out | <i>output_data</i>      | Encrypted or decrypted output data.                                            |

**Returns**

ehsm\_uint32\_t

**Return values**

|                                   |                                                           |
|-----------------------------------|-----------------------------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given session handle is unknown or wrong                  |
| <i>EVITA_WRONG_CHUNK_SIZE</i>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 796 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

**4.33.1.4 Check\_Time\_Stamp()**

```
ehsm_uint32_t Check_Time_Stamp (
 ehsm_uint32_t msg_imprint_size,
 const ehsm_uint8_t * msg_imprint,
 ehsm_uint32_t verification_key_handle,
```



```

ehsm_uint32_t verification_key_authorization_size,
ehsm_uint8_t * verification_key_authorization_value,
const signature_st * time_stamp,
ehsm_bool_t * time_stamp_vry,
ehsm_uint32_t * delta)

```

If a time stamp has been created by the Create\_Time\_Stamp function described before, the Check\_Time\_Stamp function may be used to check if the time stamp is valid.

#### Parameters

|     |                                             |                                                            |
|-----|---------------------------------------------|------------------------------------------------------------|
| in  | <i>msg_imprint_size</i>                     | size of message imprint (e.g. hash) to become time stamped |
| in  | <i>msg_imprint</i>                          | message imprint (e.g. hash) to become time stamped         |
| in  | <i>verification_key_handle</i>              | reference key that should verify the time stamp            |
| in  | <i>verification_key_authorization_size</i>  | size of verification key authorization (if set)            |
| in  | <i>verification_key_authorization_value</i> | verification key authorization (if set)                    |
|     | <i>[in/out]</i>                             | time_stamp size of signed time stamp                       |
| out | <i>time_stamp_vry</i>                       | true if verification succeeded, otherwise false            |
| out | <i>delta</i>                                | delta between actual UTC time and time_stamp time          |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                     |                                                     |
|-------------------------------------|-----------------------------------------------------|
| <i>EVITA_OK(value)</i>              | 0) For success.                                     |
| <i>EVITA_INVALID_TIME_STAMP</i>     | Given time stamp could not be interpreted correctly |
| <i>EVITA_CLOCK_NOT_SYNCHRONIZED</i> | EVITA UTC clock is not synchronized yet             |

#### 4.33.1.5 Cipher\_Finish()

```

ehsm_uint32_t Cipher_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data)

```

EVITA cipher finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by [Cipher\\_Process\(\)](#).

#### Parameters

|     |                         |                                                                       |
|-----|-------------------------|-----------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session handle to be released.                                        |
| out | <i>output_data_size</i> | Size of last output data(can be 0).                                   |
| out | <i>output_data</i>      | Last encrypted or decrypted output data(e.g., due to padding scheme). |

#### Returns

ehsm\_uint32\_t

## Return values

|                                            |                                          |
|--------------------------------------------|------------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                          |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given session handle is unknown or wrong |

Definition at line 490 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.33.1.6 Cipher\_Init()

```
ehsm_uint32_t Cipher_Init (
 ehsm_uint32_t algorithm_identifier,
 cipher_mode_e cipher_mode,
 operation_mode_e operation_mode,
 padding_scheme_e padding,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t iv_size,
 ehsm_uint8_t * iv,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA cipher initialization, aimed to prepare all the data that used in the [Cipher\\_Process\(\)](#) later.

## Parameters

|     |                                          |                                                                           |
|-----|------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>              | Reference to associated symmetric algorithm.                              |
| in  | <i>cipher_mode</i>                       | Indicate decryption pr encryption mode.                                   |
| in  | <i>operation_mode</i>                    | Indicate cipher mode of operation.                                        |
| in  | <i>padding</i>                           | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>              | Give total message length(can be req. by padding scheme).                 |
| in  | <i>iv_size</i>                           | Size of given initialization vector(can be 0).                            |
| in  | <i>iv</i>                                | Set initialization vector(it's public).                                   |
| in  | <i>key_handle</i>                        | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>            | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>           | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle-&gt;session_id</i>     | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i> | Maxium size of a chunk on <a href="#">Cipher_Process()</a> .              |
| out | <i>session_handle-&gt;block_sz</i>       | chunk has to be a multiple of this block size(1 to max).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                             |                                                                                         |
|---------------------------------------------|-----------------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>             | 0) For success.                                                                         |
| <a href="#">EVITA_WRONG_KEY_HANDLE</a>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode)        |
| <a href="#">EVITA_ALL_SESSIONS_OCCUPIED</a> | No resources left for an additional parallel session (or no parallel processing at all) |

## Return values

|                                  |                                                 |
|----------------------------------|-------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>     | Given algorithm or algorithm mode not available |
| <i>EVITA_WRONG_IV</i>            | Given IV does not fit the given algorithm       |
| <i>EVITA_AUTHORIZATION_FAILE</i> | Given authorization value was wron              |

Definition at line 233 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.33.1.7 Cipher\_Process()

```
ehsm_uint32_t Cipher_Process (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t input_data_size,
 const ehsm_uint8_t * input_data,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data)
```

EVITA cipher process, aimed to process one or more block within the encryption and decryption process of longer messages.

## Parameters

|     |                         |                                                                                |
|-----|-------------------------|--------------------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session reference from cipher_Init().                                          |
| in  | <i>input_data_size</i>  | Size of input data.                                                            |
| in  | <i>input_data</i>       | Input data to be encrypted or decrypted.                                       |
| out | <i>output_data_size</i> | Size of output data(can be different to input size due to padding/de-padding). |
| out | <i>output_data</i>      | Encrypted or decrypted output data.                                            |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                           |
|-----------------------------------|-----------------------------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given session handle is unknown or wrong                  |
| <i>EVITA_WRONG_CHUNK_SIZE</i>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 423 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.33.1.8 Create\_Counter()

```
ehsm_uint32_t Create_Counter (
 ehsm_uint32_t access_authorization_size,
 const ehsm_uint8_t * access_authorization_value,
 ehsm_uint32_t * counter_identifier,
 ehsm_counter_value_st * counter_initial_value)
```

This function is used for the initial creation of a counter.

## Parameters

|     |                                   |                                                          |
|-----|-----------------------------------|----------------------------------------------------------|
| in  | <i>access_authorization_size</i>  | size of counter access authorization value               |
| in  | <i>access_authorization_value</i> | counter access authorization value (i.e., password hash) |
| out | <i>counter_identifier</i>         | counter id for later reference                           |
| out | <i>counter_initial_value</i>      | initial counter value                                    |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                                   |
|----------------------------------------|---------------------------------------------------|
| <a href="#"><i>EVITA_OK(value)</i></a> | 0) For success.                                   |
| <i>EVITA_ALL_COUNTERS_OCCUPIED</i>     | No resources left to create an additional counter |

## 4.33.1.9 Create\_Derived\_Key()

```
ehsm_uint32_t Create_Derived_Key (
 ehsm_uint32_t key_derivation_function_identifier,
 ehsm_uint32_t key_size,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t key_usages_size,
 ehsm_uint8_t * key_usages_data,
 ehsm_uint32_t parent_key_handle,
 ehsm_uint32_t parent_key_authorization_size,
 ehsm_uint8_t * parent_key_authorization_value,
 ehsm_uint32_t salt_size,
 ehsm_uint8_t * salt_data,
 ehsm_uint32_t * key_handle)
```

This function creates a key in a similar way as an RNG-based key generation, but with a symmetric parent key as a base.

## Parameters

|     |                                           |                                                                                  |
|-----|-------------------------------------------|----------------------------------------------------------------------------------|
| in  | <i>key_derivation_function_identifier</i> | reference to underlying key derivation function (KDF)                            |
| in  | <i>key_size</i>                           | Must not be 0 if a key shorter than KDF output is allowed, otherwise error.      |
| in  | <i>valid_until</i>                        | define key life limitation as UTC time                                           |
| in  | <i>type</i>                               | memory_target non-volatile or RAM                                                |
| in  | <i>key_usages_size</i>                    | key usages size                                                                  |
| in  | <i>key_usages_data</i>                    | key usages data to define set of allowed key usages, cf. data structures section |
| in  | <i>parent_key_handle</i>                  | refers to internal secret key that will be used as parent                        |
| in  | <i>parent_key_authorization_size</i>      | size of local key usage authorization value (0 for none)                         |
| in  | <i>parent_key_authorization_value</i>     | local key usage authorization (i.e., password)                                   |
| in  | <i>salt_size</i>                          | size of cryptographic salt                                                       |
| in  | <i>salt_data</i>                          | random data for cryptographic salt                                               |
| out | <i>key_handle</i>                         | return key handle of derived key for later use in other functions                |

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>        | Given algorithm or algorithm mode not available                             |
| <i>EVITA_INVALID_KEY_SIZE</i>       | Given key size is invalid (e.g., too small or too large)                    |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key                               |
| <i>EVITA_INVALID_KEY_FLAG</i>       | Given key flag is invalid (e.g., wrong combination)                         |
| <i>EVITA_WRONG_AUTHORIZATION</i>    | Given authorization structure does not fit key flags                        |
| <i>EVITA_WRONG_KEY_HANDLE</i>       | Given key handle is unknown or wrong (e.g., not for this algorithm or mode) |
| <i>EVITA_AUTHORIZATION_FAILED</i>   | Given authorization value was wrong                                         |

## Note

Definition at line 325 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.10 Create\_Dh\_Key()

```
ehsm_uint32_t Create_Dh_Key (
 ehsm_uint32_t target_algorithm,
 ehsm_uint32_t key_size,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t key_usages_size,
 ehsm_uint8_t * key_usages_data,
 ehsm_uint32_t local_key_handle,
 ehsm_uint32_t local_key_auth_size,
 ehsm_uint8_t * local_key_auth_value,
 ehsm_uint32_t remote_key_handle,
 ehsm_uint32_t remote_key_auth_size,
 ehsm_uint8_t * remote_key_auth_value,
 ehsm_uint32_t * key_handle)
```

Create a Diffie-Hellman key according to the specified algorithm.

## Parameters

|     |                              |                                                                                                                                               |
|-----|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>target_algorithm</i>      | reference to target algorithm for key generation/usage based on DH outputs, otherwise = 0                                                     |
| in  | <i>key_size</i>              | if algorithm_identifier = 0 then define key length in bits, otherwise ignored; must not be greater than DH parent key lengths otherwise error |
| in  | <i>valid_until</i>           | define key life limitation as UTC time                                                                                                        |
| in  | <i>type</i>                  | memory_target non-volatile or RAM                                                                                                             |
| in  | <i>key_usages_size</i>       | key usages size                                                                                                                               |
| in  | <i>key_usages_data</i>       | key usages data, refers to <a href="#">ehsm_key_usages_st</a>                                                                                 |
| in  | <i>local_key_handle</i>      | refers to internal private key that will be used by DH                                                                                        |
| in  | <i>local_key_auth_size</i>   | size of local key usage authorization value (0 for none)                                                                                      |
| in  | <i>local_key_auth_value</i>  | local key usage authorization (i.e.,password)                                                                                                 |
| in  | <i>remote_key_handle</i>     | refers to public remote key that will be used by DH                                                                                           |
| in  | <i>remote_key_auth_size</i>  | size of remote key usage authorization value (0 for none)                                                                                     |
| in  | <i>remote_key_auth_value</i> | remote key usage authorization (i.e.,password)                                                                                                |
| out | <i>key_handle</i>            | return key handle of DH-calculated shared secret for later use in other functions                                                             |

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                                                                                                  |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <i>EVITA_OK</i>                      | Request successful                                                                                               |
| <i>EVITA_ALGORITHM_ERROR</i>         | Given algorithm or algorithm mode not available                                                                  |
| <i>EVITA_INVALID_KEY_SIZE</i>        | Given key size is invalid (e.g., too small or too large)                                                         |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i>  | No resources left to create an additional key                                                                    |
| <i>EVITA_INVALID_KEY_FLAG</i>        | Given key flag is invalid (e.g., wrong combination)                                                              |
| <i>EVITA_WRONG_AUTHORIZATION</i>     | Given authorization structure does not fit key flags (e.g., authorization definition for a certain flag missing) |
| <i>EVITA_WRONG_KEY_HANDLE</i>        | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)                                      |
| <i>EVITA_WRONG_REMOTE_KEY_HANDLE</i> | Given remote key handle is unknown or wrong (e.g., not for this algorithm or this mode)                          |
| <i>EVITA_WRONG_KEY_COMBINATION</i>   | Keys do not fit the algorithm (e.g., RSA key vs. ECDH)                                                           |
| <i>EVITA_AUTHORIZATION_FAILED</i>    | Given authorization value was wrong                                                                              |

## Note

Definition at line 174 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.11 Create\_Random\_Dh\_Key\_Pair()

```
ehsm_uint32_t Create_Random_Dh_Key_Pair (
 ehsm_uint32_t key_size,
 ehsm_uint32_t valid_until,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t key_usages_size,
 ehsm_uint8_t * key_usages_data,
 ehsm_uint8_t * p,
 ehsm_uint32_t p_size,
 ehsm_uint8_t * q,
 ehsm_uint32_t q_size,
 ehsm_uint8_t * g,
 ehsm_uint32_t g_size,
 ehsm_uint32_t * key_handle)
```

This function is used for creating a DH key pair.

## Parameters

|    |                        |                                                                                  |
|----|------------------------|----------------------------------------------------------------------------------|
| in | <i>key_size</i>        | Must not be 0 if a key shorter than KDF output is allowed, otherwise error.      |
| in | <i>valid_until</i>     | define key life limitation as UTC time                                           |
| in | <i>type</i>            | memory_target non-volatile or RAM                                                |
| in | <i>key_usages_size</i> | key usages size                                                                  |
| in | <i>key_usages_data</i> | key usages data to define set of allowed key usages, cf. data structures section |
| in | <i>p</i>               | a prime defining the GF(p)                                                       |

## Parameters

|     |                   |                                               |
|-----|-------------------|-----------------------------------------------|
| in  | <i>p_size</i>     | size of q                                     |
| in  | <i>q</i>          | a prime factor of p-1, aka order of g         |
| in  | <i>q_size</i>     | size of q                                     |
| in  | <i>g</i>          | a generator of the q-order subgroup of GF(p)* |
| in  | <i>g_size</i>     | size of g                                     |
| out | <i>key_handle</i> | return key handle of created key pair         |

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                                          |
|-------------------------------------|----------------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>        | Given algorithm or algorithm mode not available          |
| <i>EVITA_INVALID_KEY_SIZE</i>       | Given key size is invalid (e.g., too small or too large) |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key            |
| <i>EVITA_INVALID_KEY_FLAG</i>       | Given key flag is invalid (e.g., wrong combination)      |
| <i>EVITA_WRONG_AUTHORIZATION</i>    | Given authorization structure does not fit key flags     |

## Note

Definition at line 125 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.12 Create\_Random\_Key()

```
ehsm_uint32_t Create_Random_Key (
 ehsm_uint32_t target_algorithm,
 ehsm_uint32_t key_size,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t key_usages_size,
 ehsm_uint8_t * key_usages_data,
 ehsm_uint32_t * key_handle)
```

Create a random key according to the specified algorithm.

## Parameters

|     |                         |                                                                                                                                     |
|-----|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>target_algorithm</i> | reference to target algorithm for key generation/usage based on RNG outputs, otherwise = 0 (cf. hardware interface data structures) |
| in  | <i>key_size</i>         | if algorithm_identifier = 0 (just random string) then define key length in bits, otherwise ignored                                  |
| in  | <i>valid_until</i>      | define key life limitation as UTC time                                                                                              |
| in  | <i>type</i>             | memory_target non-volatile or RAM                                                                                                   |
| in  | <i>key_usages_size</i>  | key usages size                                                                                                                     |
| in  | <i>key_usages_data</i>  | key usages data, refers to <a href="#">ehsm_key_usages_st</a>                                                                       |
| out | <i>key_handle</i>       | return key handle for later use in other functions                                                                                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                                                                                                  |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <i>EVITA_OK</i>                     | Request successful                                                                                               |
| <i>EVITA_ALGORITHM_ERROR</i>        | Given algorithm or algorithm mode not available                                                                  |
| <i>EVITA_INVALID_KEY_SIZE</i>       | Given key size is invalid (e.g., too small or too large)                                                         |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key                                                                    |
| <i>EVITA_INVALID_KEY_FLAG</i>       | Given key flag is invalid (e.g., wrong combination)                                                              |
| <i>EVITA_WRONG_AUTHORIZATION</i>    | Given authorization structure does not fit key flags (e.g., authorization definition for a certain flag missing) |

## Note

Definition at line 88 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.13 Create\_Time\_Stamp()

```
ehsm_uint32_t Create_Time_Stamp (
 ehsm_uint32_t msg_imprint_size,
 const ehsm_uint8_t * msg_imprint,
 ehsm_uint32_t signature_key_handle,
 ehsm_uint32_t signature_key_authorization_size,
 ehsm_uint8_t * signature_key_authorization_value,
 ehsm_uint32_t * signature_size,
 signature_st * signature)
```

This function allows to sign arbitrary data and to include a time stamp (in form of UTC) into the signature.

## Parameters

|     |                                          |                                                            |
|-----|------------------------------------------|------------------------------------------------------------|
| in  | <i>msg_imprint_size</i>                  | size of message imprint (e.g. hash) to become time stamped |
| in  | <i>msg_imprint</i>                       | message imprint (e.g. hash) to become time stamped         |
| in  | <i>signature_key_handle</i>              | reference key that should sign the time stamp              |
| in  | <i>signature_key_authorization_size</i>  | size of signature key authorization (if set)               |
| in  | <i>signature_key_authorization_value</i> | signature key authorization (if set)                       |
| out | <i>signature_size</i>                    | size of signed time stamp                                  |
| out | <i>signature</i>                         | signed time stamp (MAC or ECDSA depending on key_handle)   |

## Returns

ehsm\_uint32\_t

## Return values

|                        |                 |
|------------------------|-----------------|
| <i>EVITA_OK(value)</i> | 0) For success. |
|------------------------|-----------------|



## Return values

|                                     |                                                                                  |
|-------------------------------------|----------------------------------------------------------------------------------|
| <i>EVITA_INVALID_MSG_SIZE</i>       | Given message imprint size is invalid (e.g., too small or too large)             |
| <i>EVITA_WRONG_KEY_HANDLE</i>       | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <i>EVITA_CLOCK_NOT_SYNCHRONIZED</i> | EVITA UTC clock is not synchronized yet                                          |
| <i>EVITA_AUTHORIZATION_FAILED</i>   | Given authorization value was wrong                                              |

## 4.33.1.14 Delete\_Counter()

```
ehsm_uint32_t Delete_Counter (
 ehsm_uint32_t counter_identifier,
 ehsm_uint32_t access_authorization_size,
 ehsm_uint8_t * access_authorization_value)
```

With this function, a previously created counter is deleted again.

## Parameters

|    |                                   |                                                          |
|----|-----------------------------------|----------------------------------------------------------|
| in | <i>counter_identifier</i>         | id of counter to be deleted                              |
| in | <i>access_authorization_size</i>  | size of counter access authorization value               |
| in | <i>access_authorization_value</i> | counter access authorization value (i.e., password hash) |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                     |
|-----------------------------------|-------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                     |
| <i>EVITA_UNKNOWN_COUNTER_ID</i>   | Given counter identifier is unknown |
| <i>EVITA_AUTHORIZATION_FAILED</i> | Given authorization value was wrong |

## 4.33.1.15 ehsm\_get\_pub\_from\_priv()

```
ehsm_uint32_t ehsm_get_pub_from_priv (
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization,
 ehsm_uint8_t * pub_value,
 ehsm_uint32_t * pub_size,
 ehsm_uint32_t priv_key_algo)
```

The function use to get key public party that calculation from private party.

## Parameters

|     |                               |                                                       |
|-----|-------------------------------|-------------------------------------------------------|
| in  | <i>key_handle</i>             | reference to the internal key used for get public key |
| in  | <i>key_authorization_size</i> | size of key usage authorization                       |
| in  | <i>key_authorization</i>      | key usage authorization (i.e.,password)               |
| out | <i>pub_value</i>              | the output public key                                 |
|     | <i>[in/out]</i>               | pub_size the size of public key                       |
| in  | <i>priv_key_algo</i>          | the private key algorithm(only support sm2/ecc)       |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                 |
|-----------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error                                        |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.               |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | This function is called with no authentication. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Some algorithm self test failed.                |

Definition at line 531 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.16 ehsm\_self\_test()

```
ehsm_uint32_t ehsm_self_test (
 ehsm_uint32_t flag)
```

The self test service.

## Parameters

|    |             |                                      |
|----|-------------|--------------------------------------|
| in | <i>flag</i> | The flag of algorithms to be tested. |
|----|-------------|--------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                 |
|-----------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error                                        |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.               |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | This function is called with no authentication. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Some algorithm self test failed.                |

Definition at line 594 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

4.33.1.17 Get\_Tick\_Count()

```
ehsm_uint32_t Get_Tick_Count (
 ehsm_tick_value_st * tick_value)
```

This function is used for obtaining the actual value of the EVITA tick counter.

Parameters

|     |                   |                               |
|-----|-------------------|-------------------------------|
| out | <i>tick_value</i> | actual value of tick counter. |
|-----|-------------------|-------------------------------|

Returns

ehsm\_uint32\_t

Return values

|                       |                 |
|-----------------------|-----------------|
| <i>EVITA_OK(value</i> | 0) For success. |
|-----------------------|-----------------|

4.33.1.18 Get\_Time\_Sync\_Challenge()

```
ehsm_uint32_t Get_Time_Sync_Challenge (
 ehsm_uint32_t * time_sync_challenge_size,
 ehsm_uint8_t time_sync_challenge[])
```

This function is used to obtain a challenge for checking the freshness of external time synchronization.

Parameters

|     |                                 |                          |
|-----|---------------------------------|--------------------------|
| out | <i>time_sync_challenge_size</i> | size of challenge value  |
| out | <i>time_sync_challenge</i>      | obtained challenge value |

Returns

ehsm\_uint32\_t

Return values

|                       |                 |
|-----------------------|-----------------|
| <i>EVITA_OK(value</i> | 0) For success. |
|-----------------------|-----------------|

4.33.1.19 Get\_UTC\_Time()

```
ehsm_uint32_t Get_UTC_Time (
 ehsm_utc_time_t * utc_time)
```

With this function, the UTC time value is obtained from the EVITA module.

## Parameters

|     |                 |                                      |
|-----|-----------------|--------------------------------------|
| out | <i>utc_time</i> | UNIX UTC time seconds since 1.1.1970 |
|-----|-----------------|--------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                         |
|-------------------------------------|-----------------------------------------|
| <i>EVITA_OK(value)</i>              | 0) For success.                         |
| <i>EVITA_CLOCK_NOT_SYNCHRONIZED</i> | EVITA UTC clock is not synchronized yet |

## 4.33.1.20 Hash\_Finish()

```
ehsm_uint32_t Hash_Finish (
 ehsm_ctx_session_st * session_handle,
 hash_hmac_st * hash_hmac,
 ehsm_bool_t * hmac_match)
```

EVITA Hash finish, aimed to terminate the hash or HMAC calculation or HMAC verification after processing the last message block.

## Parameters

|     |                       |                                                                                                             |
|-----|-----------------------|-------------------------------------------------------------------------------------------------------------|
| in  | <i>session_handle</i> | Session handle to be released.                                                                              |
|     | <i>[in/out]</i>       | hash_hmac Data structure include HASH/HMAC data and size(optional with UTC_time_stamp in HMAC verify mode). |
| out | <i>hmac_match</i>     | True if calculated HMAC matches the given Reference HMAC(only for HMAC verify).                             |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                      |
|-----------------------------------|--------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                      |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given key handle is unknown or wrong |

Definition at line 321 of file eHSM\_If\_Evita\_Hash\_Ip.c.

## 4.33.1.21 Hash\_Init()

```
ehsm_uint32_t Hash_Init (
 ehsm_uint32_t algorithm_identifier,
```

```

hash_mode_e hash_mode,
ehsm_uint32_t key_handle,
ehsm_uint32_t key_authorization_size,
ehsm_uint8_t * key_authorization_value,
ehsm_ctx_session_st * session_handle)

```

EVITA Hash initialization, aimed to make all necessary preparations for the calculation of hash values and HMACS or verification of HAMCs belonging to longer messages that cannot be processed within one function call.

#### Parameters

|     |                                            |                                                                                               |
|-----|--------------------------------------------|-----------------------------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                | Reference to associated hash algorithm.                                                       |
| in  | <i>hash_mode</i>                           | Indicate hash or HMAC creation or verification mode.                                          |
| in  | <i>key_handle</i>                          | Refer to internal key that will be used(only for HMAC, set to 0 otherwise).                   |
| in  | <i>key_authorization_size</i>              | Size of HMAC key usage authorization value(0 for none).                                       |
| in  | <i>key_authorization_value</i>             | HMAC key usage authorization(i.e., password).                                                 |
| out | <i>session_handle</i>                      | Enables interruption & parallel processing and/or session authentication.                     |
| out | <i>session_handle-&gt;max_chunk_size</i>   | Maxium chunk size possible on update() (note that hash function have inherent padding scheme) |
| out | <i>session_handle-&gt;chunk_block_size</i> | Chunk has to be a multiple of this block size(1 to max).                                      |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                    |                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------|
| <i>EVITA_OK(value)</i>             | 0) For success.                                                                  |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional session                                      |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                  |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                              |

Definition at line 142 of file eHSM\_If\_Evita\_Hash\_Ip.c.

#### 4.33.1.22 Hash\_Update()

```

ehsm_uint32_t Hash_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)

```

EVITA Hash update, aimed to process one or several blocks(up to the maximum allowed chunk size delivered by Hash\_↔Init)within a hash or HMAC calculation or HMAC verification for longer messages.

#### Parameters

|    |                       |                                                                 |
|----|-----------------------|-----------------------------------------------------------------|
| in | <i>session_handle</i> | Session reference from init() to enable parallel hash sessions. |
| in | <i>chunk_size</i>     | Size of chunk used for hash update.                             |
| in | <i>chunk_data</i>     | Data chunk byte array used for hash update.                     |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                           |
|-----------------------------------|-----------------------------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given key handle is unknown or wrong                      |
| <i>EVITA_WRONG_CHUNK_SIZE</i>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 260 of file eHSM\_Lf\_Evita\_Hash\_Ip.c.

## 4.33.1.23 Increment\_Counter()

```
ehsm_uint32_t Increment_Counter (
 ehsm_uint32_t counter_identifier,
 ehsm_uint32_t access_authorization_size,
 ehsm_uint8_t * access_authorization_value,
 const ehsm_counter_value_st * counter_incrementation,
 ehsm_counter_value_st * counter_new_value)
```

This function is used to increment an existing counter.

## Parameters

|     |                                   |                                                          |
|-----|-----------------------------------|----------------------------------------------------------|
| in  | <i>counter_identifier</i>         | id of counter to be incremented                          |
| in  | <i>access_authorization_size</i>  | size of counter access authorization value               |
| in  | <i>access_authorization_value</i> | counter access authorization value (i.e., password hash) |
| in  | <i>counter_incrementation</i>     | counter incrementation value                             |
| out | <i>counter_new_value</i>          | new counter value after incrementation                   |

## Returns

ehsm\_uint32\_t

## Return values

|                                             |                                                           |
|---------------------------------------------|-----------------------------------------------------------|
| <i>EVITA_OK(value)</i>                      | 0) For success.                                           |
| <i>EVITA_UNKNOWN_COUNTER_ID</i>             | Given counter identifier is unknown                       |
| <i>EVITA_AUTHORIZATION_FAILED</i>           | Given authorization value was wrong                       |
| <i>EVITA_INVALID_COUNTER_INCREMENTATION</i> | Given counter incrementation is invalid (e.g., too large) |

## 4.33.1.24 Key\_Export()

```
ehsm_uint32_t Key_Export (
 ehsm_uint32_t key_handle,
```

```

key_act_use_flags_t * use_flags,
ehsm_uint32_t transport_key_handle,
ehsm_uint32_t transport_key_authorization_size,
ehsm_uint8_t * transport_key_authorization,
ehsm_uint32_t authenticity_key_handle,
ehsm_uint32_t authenticity_key_authorization_size,
ehsm_uint8_t * authenticity_key_authorization,
ehsm_uint32_t * encrypted_key_size,
ehsm_uint8_t * encrypted_key,
ehsm_uint32_t * key_authenticity_code_size,
ehsm_uint8_t * key_authenticity_code)

```

This function is used for exporting keys from eHSM.

#### Parameters

|     |                                            |                                                                                          |
|-----|--------------------------------------------|------------------------------------------------------------------------------------------|
| in  | <i>key_handle</i>                          | reference to the internal key that becomes exported                                      |
| in  | <i>use_flags</i>                           | define set of key usages to become exported                                              |
| in  | <i>transport_key_handle</i>                | reference to the key used for transport protection                                       |
| in  | <i>transport_key_authorization_size</i>    | size of transport key usage authorization                                                |
| in  | <i>transport_key_authorization</i>         | transport key usage authorization (i.e., password)                                       |
| in  | <i>authenticity_key_handle</i>             | reference to the key used for authenticity code verification (use_flag = verify)         |
| in  | <i>authenticity_key_authorization_size</i> | size of authenticity key usage authorization                                             |
| in  | <i>authenticity_key_authorization</i>      | authenticity key usage authorization (i.e., password)                                    |
| out | <i>encrypted_key_size</i>                  | returned encrypted key blob size                                                         |
| out | <i>encrypted_key</i>                       | returned encrypted key blob                                                              |
| out | <i>key_authenticity_code_size</i>          | size of key authenticity code (signature or MAC) created by transport key                |
| out | <i>key_authenticity_code</i>               | key authenticity code (signature or MAC) to enforce and proof module internal protection |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                   |                                                                                                  |
|-----------------------------------|--------------------------------------------------------------------------------------------------|
| <i>EVITA_AUTHORIZATION_FAILED</i> | Given authorization value was wrong                                                              |
| <i>EVITA_WRONG_KEY_HANDLE</i>     | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)                      |
| <i>EVITA_TRANSPORT_IMPOSSIBLE</i> | Given transport key is not a capable transport key or use flag cannot be transported or migrated |

#### Note

Definition at line 270 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.25 Key\_Import()

```

ehsm_uint32_t Key_Import (
 ehsm_uint32_t transport_key_handle,
 ehsm_uint32_t transport_key_authorization_size,
 ehsm_uint8_t * transport_key_authorization,
 ehsm_uint32_t authenticity_key_handle,
 ehsm_uint32_t authenticity_key_authorization_size,
 ehsm_uint8_t * authenticity_key_authorization,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t encrypted_key_size,
 ehsm_uint8_t * encrypted_key,
 ehsm_uint32_t key_authenticity_code_size,
 ehsm_uint8_t * key_authenticity_code,
 ehsm_uint32_t * key_handle)

```

This function is used for importing keys into eHSM that was were generated and exported by another EVITA module or another external trusted party.

## Parameters

|     |                                            |                                                                                  |
|-----|--------------------------------------------|----------------------------------------------------------------------------------|
| in  | <i>transport_key_handle</i>                | reference to the internal key used for transport protection                      |
| in  | <i>transport_key_authorization_size</i>    | size of transport key usage authorization                                        |
| in  | <i>transport_key_authorization</i>         | transport key usage authorization (i.e., password)                               |
| in  | <i>authenticity_key_handle</i>             | reference to the key used for authenticity code verification (use_flag = verify) |
| in  | <i>authenticity_key_authorization_size</i> | size of authenticity key usage authorization                                     |
| in  | <i>authenticity_key_authorization</i>      | authenticity key usage authorization (i.e., password)                            |
| in  | <i>type</i>                                | memory_target {nv ram}                                                           |
| in  | <i>encrypted_key_size</i>                  | given encrypted key blob size                                                    |
| in  | <i>encrypted_key</i>                       | given encrypted key blob                                                         |
| in  | <i>key_authenticity_code_size</i>          | given key authenticity code (signature or MAC)                                   |
| in  | <i>key_authenticity_code</i>               | given key authenticity code (signature or MAC)                                   |
| out | <i>key_handle</i>                          | reference to the (now) internal key that was imported                            |

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                                                                                  |
|-------------------------------------|--------------------------------------------------------------------------------------------------|
| <i>EVITA_AUTHORIZATION_FAILED</i>   | Given authorization value was wrong                                                              |
| <i>EVITA_WRONG_KEY_HANDLE</i>       | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)                      |
| <i>EVITA_TRANSPORT_IMPOSSIBLE</i>   | Given transport key is not a capable transport key or use flag cannot be transported or migrated |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key                                                    |

## Note

Definition at line 230 of file eHSM\_If\_Evita\_Key\_Ip.c.



## 4.33.1.26 Key\_Remove()

```
ehsm_uint32_t Key_Remove (
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization)
```

This function is used for removing loaded keys from the eHSM.

## Parameters

|    |                               |                                          |
|----|-------------------------------|------------------------------------------|
| in | <i>key_handle</i>             | reference to key that should be removed  |
| in | <i>key_authorization_size</i> | size of key remove authorization         |
| in | <i>key_authorization</i>      | key remove authorization (i.e. password) |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                                             |
|-----------------------------------|-----------------------------------------------------------------------------|
| <i>EVITA_WRONG_KEY_HANDLE</i>     | Given key handle is unknown or wrong (e.g., not for this algorithm or mode) |
| <i>EVITA_AUTHORIZATION_FAILED</i> | Given authorization value was wrong                                         |
| <i>EVITA_REMOVE_IMPOSSIBLE</i>    | Key is not allowed to become removed                                        |

## Note

Definition at line 380 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.27 Key\_Status()

```
ehsm_uint32_t Key_Status (
 ehsm_uint32_t key_handle,
 ehsm_uint32_t certification_key_handle,
 ehsm_uint32_t certification_key_authorization_size,
 ehsm_uint8_t * certification_key_authorization,
 ehsm_uint32_t * key_status_size,
 ehsm_uint8_t * key_status)
```

This function is used for obtaining all public properties about a key loaded by the eHSM.

## Parameters

|     |                                             |                                                                                  |
|-----|---------------------------------------------|----------------------------------------------------------------------------------|
| in  | <i>key_handle</i>                           | reference to key whose status is to be returned                                  |
| in  | <i>certification_key_handle</i>             | reference to key that should sign the returned key status (NULL = w/o signature) |
| in  | <i>certification_key_authorization_size</i> | size of certification key usage authorization                                    |
| in  | <i>certification_key_authorization</i>      | certification key usage authorization (i.e. password)                            |
| out | <i>key_status_size</i>                      | size of (certified) key status                                                   |
| out | <i>key_status</i>                           | (certified) key status = { public info about key    signature (optional) }       |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                                                    |
|------------------------------------|------------------------------------------------------------------------------------|
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)        |
| <i>EVITA_WRONG_CERT_KEY_HANDLE</i> | Given certification key handle is unknown or wrong (e.g., not enabled for signing) |
| <i>EVITA_AUTHORIZATION_FAILED</i>  | Given authorization value was wrong                                                |

## Note

Definition at line 408 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.28 MAC\_Finish()

```
ehsm_uint32_t MAC_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t * mac_size,
 mac_st * mac,
 ehsm_bool_t * mac_match)
```

EVITA MAC finish, aimed to terminate the MAC generation or verification process after the last block has been processed by MAC\_Update.

## Parameters

|     |                       |                                                                                                 |
|-----|-----------------------|-------------------------------------------------------------------------------------------------|
| in  | <i>session_handle</i> | Session handle to be released.                                                                  |
|     | <i>[in/out]</i>       | mac_size Size of Data structure, as an input in verify mode and as an output in sign mode.      |
|     | <i>[in/out]</i>       | mac Data structure include mac_size and mac value(optional with UTC_time_stamp in verify mode). |
| out | <i>mac_match</i>      | True if calculated MAC mateches the given reference MAC(only for verify mode).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                          |
|-----------------------------------|------------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                          |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given session handle is unknown or wrong |

Definition at line 1158 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.33.1.29 MAC\_Init()

```
ehsm_uint32_t MAC_Init (
 ehsm_uint32_t algorithm_identifier,
 mac_mode_e mac_mode,
 operation_mode_e operation_mode,
 padding_scheme_e padding_scheme,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t mac_length,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA MAC initialization, aimed to prepare all the data that used in the [MAC\\_Update\(\)](#) later.

## Parameters

|     |                                                                           |                                                                           |
|-----|---------------------------------------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                                               | Reference to associated MAC algorithm.                                    |
| in  | <i>mac_mode</i> { <i>sign</i>   <i>timestamped_sign</i>   <i>verify</i> } | Indicate MAC creation or verification mode.                               |
| in  | <i>operation_mode</i>                                                     | Indicate MAC type.                                                        |
| in  | <i>padding_scheme</i>                                                     | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>                                               | Give total message length(can be req. by padding scheme).                 |
| in  | <i>mac_length</i>                                                         | Length of MAC(eg., if MAC < AES block size)                               |
| in  | <i>key_handle</i>                                                         | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>                                             | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>                                            | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle</i>                                                     | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i>                                  | Maxium size of a chunk on update().                                       |
| out | <i>session_handle-&gt;chunk_block_size</i>                                | Chunk has to be a multiple of this block size(1 to max).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                                                         |
|------------------------------------|-----------------------------------------------------------------------------------------|
| <a href="#">EVITA_OK</a> (value)   | 0) For success.                                                                         |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode)        |
| <i>EVITA_MAC_LENGTH_OVERSIZE</i>   | Given MAC length for verification is greater than MAC                                   |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional parallel session (or no parallel processing at all) |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                         |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                                     |

Definition at line 959 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.33.1.30 MAC\_Update()

```
ehsm_uint32_t MAC_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA MAC update, aimed to process one or more block within a MAC generation or verification process of longer messages.

## Parameters

|    |                       |                                                                                         |
|----|-----------------------|-----------------------------------------------------------------------------------------|
| in | <i>session_handle</i> | Session reference from <a href="#">MAC_Init()</a> .                                     |
| in | <i>chunk_size</i>     | Size of data chunk used for MAC update.                                                 |
| in | <i>chunk_data</i>     | Data chunk byte array used for MAC update(data to be verified or protected with a MAC). |

## Returns

ehsm\_uint32\_t

## Return values

|                                            |                                                           |
|--------------------------------------------|-----------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                                           |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given key handle is unknown or wrong                      |
| <a href="#">EVITA_WRONG_CHUNK_SIZE</a>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 1102 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.33.1.31 Module\_Status()

```
ehsm_uint32_t Module_Status (
 ehsm_uint32_t flag,
 ehsm_uint32_t alg,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_auth_size,
 const ehsm_uint8_t * key_auth_value,
 ehsm_uint32_t * status_size,
 ehsm_uint8_t * status,
 ehsm_uint32_t * sign_size,
 ehsm_uint8_t * sign)
```

Get the status of eHSM.

## Parameters

|                      |                       |                                                                       |
|----------------------|-----------------------|-----------------------------------------------------------------------|
| in                   | <i>flag</i>           | Indicate type of module status info to be returned.                   |
| in                   | <i>alg</i>            | Reference to signing algorithm (could also be a MAC scheme).          |
| in                   | <i>key_handle</i>     | Reference key that signs returned status (NULL = w/o signature).      |
| in                   | <i>key_auth_size</i>  | Size of key usage authorization value (0 for none).                   |
| in                   | <i>key_auth_value</i> | Key usage authorization (i.e., password).                             |
|                      | <i>[in/out]</i>       | <i>status_size</i> : Size of module status structure.                 |
|                      | <i>[in/out]</i>       | <i>status</i> : Module status structure data.                         |
| Generated by Doxygen | <i>[in/out]</i>       | <i>sign_size</i> : Status signature size (if requested, otherwise 0). |
|                      | <i>[in/out]</i>       | <i>sign</i> : Status signature (if requested, otherwise NULL).        |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                                                    |
|----------------------------------------|--------------------------------------------------------------------|
| <i>EVITA_OK(value)</i>                 | 0) For success.                                                    |
| <i>EVITA_STATUS_TYPE_NOT_AVAILABLE</i> | Requested status type is (currently) not available for this module |
| <i>EVITA_WRONG_KEY_HANDLE</i>          | Given key handle is unknown or wrong                               |
| <i>EVITA_AUTHORIZATION_FAILED</i>      | Given authorization value was wrong                                |

Definition at line 493 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.33.1.32 Read\_Counter()

```
ehsm_uint32_t Read_Counter (
 ehsm_uint32_t counter_identifier,
 ehsm_counter_value_st * counter_current_value)
```

This function is used to read the value of a counter.

## Parameters

|     |                              |                                  |
|-----|------------------------------|----------------------------------|
| in  | <i>counter_identifier</i>    | counter id of counter to be read |
| out | <i>counter_current_value</i> | current counter value            |

## Returns

ehsm\_uint32\_t

## Return values

|                                 |                                     |
|---------------------------------|-------------------------------------|
| <i>EVITA_OK(value)</i>          | 0) For success.                     |
| <i>EVITA_UNKNOWN_COUNTER_ID</i> | Given counter identifier is unknown |

## 4.33.1.33 RNG\_Get\_Random()

```
ehsm_uint32_t RNG_Get_Random (
 ehsm_uint32_t algorithm_identifier,
 ehsm_uint32_t random_byte_request_size,
 ehsm_uint8_t * random_bytes)
```

Generate the random number.

## Parameters

|    |                                 |                                                                         |
|----|---------------------------------|-------------------------------------------------------------------------|
| in | <i>algorithm_identifier</i>     | reference to associated (pseudo) random algorithm (e.g., CTRDRBG, TRNG) |
| in | <i>random_byte_request_size</i> | number of random bytes to be returned                                   |
| in | <i>random_bytes</i>             | returned random bytes                                                   |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                     |
|------------------------------------|-----------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available     |
| <i>EVITA_PRNG_REQUEST_OVERSIZE</i> | Requested number of random bytes exceeds PRNG limit |

## Returns

EVITA\_TRNG\_SEED\_FAILURE PRNG was unable to retrieve true random seed from TRNG

## Note

Definition at line 54 of file eHSM\_If\_Evita\_Rng\_Ip.c.

## 4.33.1.34 Set.UTC\_Time()

```
ehsm_uint32_t Set.UTC_Time (
 ehsm_utc_time_t utc_time,
 ehsm_uint32_t signature_size,
 const ehsm_uint8_t * signature,
 ehsm_uint32_t verification_key_handle,
 ehsm_uint32_t verification_key_authorization_size,
 ehsm_uint8_t * verification_key_authorization_value)
```

This function is used for setting the UTC time of the EVITA module.

## Parameters

|    |                                             |                                                    |
|----|---------------------------------------------|----------------------------------------------------|
| in | <i>utc_time</i>                             | UNIX UTC time (seconds since 1.1.1970)             |
| in | <i>signature_size</i>                       | size of signature over UTC time and challenge      |
| in | <i>signature</i>                            | signature over UTC time and challenge              |
| in | <i>verification_key_handle</i>              | reference key that should verify the UTC reference |
| in | <i>verification_key_authorization_size</i>  | size of verification key authorization (if set)    |
| in | <i>verification_key_authorization_value</i> | verification key authorization (if set)            |

## Returns

ehsm\_uint32\_t

## Return values

|                               |                                                                                  |
|-------------------------------|----------------------------------------------------------------------------------|
| <i>EVITA_OK(value)</i>        | 0) For success.                                                                  |
| <i>EVITA_INVALID_UTC_TIME</i> | Given UTC time could not be interpreted correctly                                |
| <i>EVITA_WRONG_KEY_HANDLE</i> | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |

## Return values

|                                         |                                                                       |
|-----------------------------------------|-----------------------------------------------------------------------|
| <i>EVITA_UTC_CHALLENGE_EXPIRED</i>      | Challenge was not requested or is expired already                     |
| <i>EVITA_UTC_SYNCHRONIZATION_FAILED</i> | Synchronization failed due to wrong signature or wrong challenge etc. |
| <i>EVITA_AUTHORIZATION_FAILED</i>       | Given authorization value was wrong                                   |

## 4.33.1.35 Sign\_Finish()

```
ehsm_uint32_t Sign_Finish (
 ehsm_ctx_session_st * session_handle,
 signature_st * signature)
```

EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.

## Parameters

|     |                       |                                                    |
|-----|-----------------------|----------------------------------------------------|
| in  | <i>session_handle</i> | Session reference from <a href="#">Sign_Init()</a> |
| out | <i>signature</i>      | Pointer to signature.                              |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                      |
|-----------------------------------|--------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                      |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given key handle is unknown or wrong |

Definition at line 373 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.33.1.36 Sign\_Init()

```
ehsm_uint32_t Sign_Init (
 ehsm_asym_alg_e algorithm_identifier,
 ehsm_uint32_t hash_algorithm_identifier,
 padding_scheme_e padding,
 ehsm_uint32_t total_message_length,
 ehsm_bool_t time_stamp_signature,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the [Sign\\_Update\(\)](#) later.

## Parameters

|     |                                            |                                                                           |
|-----|--------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                | Reference to associated asymmetric algorithm.                             |
| in  | <i>hash_algorithm_identifier</i>           | Indicate underlying hash algorithm.                                       |
| in  | <i>padding</i>                             | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>                | Give total message length(can be req. by padding scheme).                 |
| in  | <i>key_handle</i>                          | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>              | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>             | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle</i>                      | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i>   | Maxium size of a chunk on AsymCipher_Process().                           |
| out | <i>session_handle-&gt;chunk_block_size</i> | chunk has to be a multiple of this block size(1 to max).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>    | 0) For success.                                                                  |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional session                                      |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                  |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                              |

Definition at line 194 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.33.1.37 Sign\_Update()

```
ehsm_uint32_t Sign_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA asymmetric cipher update, aimed to process the data in chunks.

## Parameters

|    |                       |                                                    |
|----|-----------------------|----------------------------------------------------|
| in | <i>session_handle</i> | Session reference from <a href="#">Sign_Init()</a> |
| in | <i>chunk_size</i>     | Size of chunk data.                                |
| in | <i>chunk_data</i>     | Pointer to chunk data.                             |

## Returns

ehsm\_uint32\_t



## Return values

|                                        |                                                           |
|----------------------------------------|-----------------------------------------------------------|
| <a href="#"><i>EVITA_OK(value)</i></a> | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i>      | Given key handle is unknown or wrong                      |
| <i>EVITA_WRONG_CHUNK_SIZE</i>          | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 320 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.33.1.38 Verify\_Finish()

```
ehsm_uint32_t Verify_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_utc_time_t * utc_time_stamp,
 ehsm_bool_t * sign_match)
```

EVITA asymmetric cipher finalization, aimed to finalize the asymmetric cipher process.

## Parameters

|     |                       |                                                      |
|-----|-----------------------|------------------------------------------------------|
| in  | <i>session_handle</i> | Session reference from <a href="#">Verify_Init()</a> |
| out | <i>utc_time_stamp</i> | Verified UTC time stamp(if available)                |
| out | <i>sign_match</i>     | Indicate whether the signature match or not.         |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                      |
|----------------------------------------|--------------------------------------|
| <a href="#"><i>EVITA_OK(value)</i></a> | 0) For success.                      |
| <i>EVITA_WRONG_SESSION_HANDLE</i>      | Given key handle is unknown or wrong |

Definition at line 640 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.33.1.39 Verify\_Init()

```
ehsm_uint32_t Verify_Init (
 ehsm_asym_alg_e algorithm_identifier,
 ehsm_uint32_t hash_algorithm_identifier,
 padding_scheme_e padding,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 signature_st * signature,
 ehsm_ctx_session_st * session_handle)
```

EVITA asymmetric cipher initialization, aimed to prepare all the data that used in the [Verify\\_Update\(\)](#) later.

## Parameters

|     |                                            |                                                                           |
|-----|--------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                | Reference to associated asymmetric algorithm.                             |
| in  | <i>hash_algorithm_identifier</i>           | Indicate underlying hash algorithm.                                       |
| in  | <i>padding</i>                             | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>                | Give total message length(can be req. by padding scheme).                 |
| in  | <i>key_handle</i>                          | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>              | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>             | Key usage authorization(i.e., password).                                  |
| in  | <i>signature</i>                           | Pointer to signature.                                                     |
| out | <i>session_handle</i>                      | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i>   | Maxium size of a chunk on AsymCipher_Process().                           |
| out | <i>session_handle-&gt;chunk_block_size</i> | chunk has to be a multiple of this block size(1 to max).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>    | 0) For success.                                                                  |
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode) |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i> | No resources left for an additional session                                      |
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available                                  |
| <i>EVITA_AUTHORIZATION_FAILE</i>   | Given authorization value was wrong                                              |

Definition at line 455 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.33.1.40 Verify\_Update()

```
ehsm_uint32_t Verify_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA asymmetric cipher update, aimed to process the data in chunks.

## Parameters

|    |                       |                                                      |
|----|-----------------------|------------------------------------------------------|
| in | <i>session_handle</i> | Session reference from <a href="#">Verify_Init()</a> |
| in | <i>chunk_size</i>     | Size of chunk data.                                  |
| in | <i>chunk_data</i>     | Pointer to chunk data.                               |

## Returns

ehsm\_uint32\_t

## Return values

|                                            |                                                           |
|--------------------------------------------|-----------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                                           |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given key handle is unknown or wrong                      |
| <a href="#">EVITA_WRONG_CHUNK_SIZE</a>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 586 of file eHSM\_If\_Evita\_AsymCper\_Ip.c.

## 4.34 eHSM\_If\_Evita\_Key\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include <string.h>
#include "eHSM_Debug_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_If_Ext_Types_Ip.h"
```

### Functions

- [void translate\\_bool\\_array\\_to\\_use\\_flags](#) ([ehsm\\_uint32\\_t](#) \*export\_use\_flags, [key\\_act\\_use\\_flags\\_t](#) \*use\_flags)
- [ehsm\\_uint32\\_t Create\\_Random\\_Key](#) ([ehsm\\_uint32\\_t](#) target\_algorithm, [ehsm\\_uint32\\_t](#) key\_size, [ehsm\\_key\\_mem\\_type\\_e](#) type, [ehsm\\_uint32\\_t](#) key\_usages\_size, [ehsm\\_uint8\\_t](#) \*key\_usages\_data, [ehsm\\_uint32\\_t](#) \*key\_handle)  
*Create a random key according to the specified algorithm.*
- [ehsm\\_uint32\\_t Create\\_Random\\_Dh\\_Key\\_Pair](#) ([ehsm\\_uint32\\_t](#) key\_size, [ehsm\\_uint32\\_t](#) valid\_until, [ehsm\\_key\\_mem\\_type\\_e](#) type, [ehsm\\_uint32\\_t](#) key\_usages\_size, [ehsm\\_uint8\\_t](#) \*key\_usages\_data, [ehsm\\_uint8\\_t](#) \*p, [ehsm\\_uint32\\_t](#) p\_size, [ehsm\\_uint8\\_t](#) \*q, [ehsm\\_uint32\\_t](#) q\_size, [ehsm\\_uint8\\_t](#) \*g, [ehsm\\_uint32\\_t](#) g\_size, [ehsm\\_uint32\\_t](#) \*key\_handle)  
*This function is used for creating a DH key pair.*
- [ehsm\\_uint32\\_t Create\\_Dh\\_Key](#) ([ehsm\\_uint32\\_t](#) target\_algorithm, [ehsm\\_uint32\\_t](#) key\_size, [ehsm\\_key\\_mem\\_type\\_e](#) type, [ehsm\\_uint32\\_t](#) key\_usages\_size, [ehsm\\_uint8\\_t](#) \*key\_usages\_data, [ehsm\\_uint32\\_t](#) local\_key\_handle, [ehsm\\_uint32\\_t](#) local\_key\_auth\_size, [ehsm\\_uint8\\_t](#) \*local\_key\_auth\_value, [ehsm\\_uint32\\_t](#) remote\_key\_handle, [ehsm\\_uint32\\_t](#) remote\_key\_auth\_size, [ehsm\\_uint8\\_t](#) \*remote\_key\_auth\_value, [ehsm\\_uint32\\_t](#) \*key\_handle)  
*Create a Diffie-Hellman key according to the specified algorithm.*
- [ehsm\\_uint32\\_t Key\\_Import](#) ([ehsm\\_uint32\\_t](#) transport\_key\_handle, [ehsm\\_uint32\\_t](#) transport\_key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*transport\_key\_authorization, [ehsm\\_uint32\\_t](#) authenticity\_key\_handle, [ehsm\\_uint32\\_t](#) authenticity\_key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*authenticity\_key\_authorization, [ehsm\\_key\\_mem\\_type\\_e](#) type, [ehsm\\_uint32\\_t](#) encrypted\_key\_size, [ehsm\\_uint8\\_t](#) \*encrypted\_key, [ehsm\\_uint32\\_t](#) key\_authenticity\_code\_size, [ehsm\\_uint8\\_t](#) \*key\_authenticity\_code, [ehsm\\_uint32\\_t](#) \*key\_handle)  
*This function is used for importing keys into eHSM that was were generated and exported by another EVITA module or another external trusted party.*
- [ehsm\\_uint32\\_t Key\\_Export](#) ([ehsm\\_uint32\\_t](#) key\_handle, [key\\_act\\_use\\_flags\\_t](#) \*use\_flags, [ehsm\\_uint32\\_t](#) transport\_key\_handle, [ehsm\\_uint32\\_t](#) transport\_key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*transport\_key\_authorization, [ehsm\\_uint32\\_t](#) authenticity\_key\_handle, [ehsm\\_uint32\\_t](#) authenticity\_key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*authenticity\_key\_authorization, [ehsm\\_uint32\\_t](#) \*encrypted\_key\_size, [ehsm\\_uint8\\_t](#) \*encrypted\_key, [ehsm\\_uint32\\_t](#) \*key\_authenticity\_code\_size, [ehsm\\_uint8\\_t](#) \*key\_authenticity\_code)  
*This function is used for exporting keys from eHSM.*

- `ehsm_uint32_t Create_Derived_Key` (`ehsm_uint32_t` key\_derivation\_function\_identifier, `ehsm_uint32_t` key\_size, `ehsm_key_mem_type_e` type, `ehsm_uint32_t` key\_usages\_size, `ehsm_uint8_t` \*key\_usages\_data, `ehsm_uint32_t` parent\_key\_handle, `ehsm_uint32_t` parent\_key\_authorization\_size, `ehsm_uint8_t` \*parent\_key\_authorization\_value, `ehsm_uint32_t` salt\_size, `ehsm_uint8_t` \*salt\_data, `ehsm_uint32_t` \*key\_handle)

*This function creates a key in a similar way as an RNG-based key generation, but with a symmetric parent key as a base.*

- `ehsm_uint32_t Key_Remove` (`ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization)

*This function is used for removing loaded keys from the eHSM.*

- `ehsm_uint32_t Key_Status` (`ehsm_uint32_t` key\_handle, `ehsm_uint32_t` certification\_key\_handle, `ehsm_uint32_t` certification\_key\_authorization\_size, `ehsm_uint8_t` \*certification\_key\_authorization, `ehsm_uint32_t` \*key\_status\_size, `ehsm_uint8_t` \*key\_status)

*This function is used for obtaining all public properties about a key loaded by the eHSM.*

- `ehsm_uint32_t Module_Status` (`ehsm_uint32_t` type, `ehsm_uint32_t` algo\_id, `ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_auth\_size, const `ehsm_uint8_t` \*key\_auth\_value, `ehsm_uint32_t` \*status\_size, `ehsm_uint8_t` \*status, `ehsm_uint32_t` \*sign\_size, `ehsm_uint8_t` \*sign)

*Get the status of eHSM.*

- `ehsm_uint32_t ehsm_get_pub_from_priv` (`ehsm_uint32_t` key\_handle, `ehsm_uint32_t` key\_authorization\_size, `ehsm_uint8_t` \*key\_authorization, `ehsm_uint8_t` \*partner\_pub\_value, `ehsm_uint32_t` \*partner\_pub\_size, `ehsm_uint32_t` priv\_key\_algo)

*The function use to get key public party that calculation from private party.*

## 4.34.1 Function Documentation

### 4.34.1.1 \_translate\_bool\_array\_to\_use\_flags()

```
void _translate_bool_array_to_use_flags (
 ehsm_uint32_t * export_use_flags,
 key_act_use_flags_t * use_flags)
```

Definition at line 444 of file eHSM\_If\_Evita\_Key\_Ip.c.

### 4.34.1.2 Create\_Derived\_Key()

```
ehsm_uint32_t Create_Derived_Key (
 ehsm_uint32_t key_derivation_function_identifier,
 ehsm_uint32_t key_size,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t key_usages_size,
 ehsm_uint8_t * key_usages_data,
 ehsm_uint32_t parent_key_handle,
 ehsm_uint32_t parent_key_authorization_size,
 ehsm_uint8_t * parent_key_authorization_value,
 ehsm_uint32_t salt_size,
 ehsm_uint8_t * salt_data,
 ehsm_uint32_t * key_handle)
```

This function creates a key in a similar way as an RNG-based key generation, but with a symmetric parent key as a base.

## Parameters

|     |                                           |                                                                                  |
|-----|-------------------------------------------|----------------------------------------------------------------------------------|
| in  | <i>key_derivation_function_identifier</i> | reference to underlying key derivation function (KDF)                            |
| in  | <i>key_size</i>                           | Must not be 0 if a key shorter than KDF output is allowed, otherwise error.      |
| in  | <i>valid_until</i>                        | define key life limitation as UTC time                                           |
| in  | <i>type</i>                               | memory_target non-volatile or RAM                                                |
| in  | <i>key_usages_size</i>                    | key usages size                                                                  |
| in  | <i>key_usages_data</i>                    | key usages data to define set of allowed key usages, cf. data structures section |
| in  | <i>parent_key_handle</i>                  | refers to internal secret key that will be used as parent                        |
| in  | <i>parent_key_authorization_size</i>      | size of local key usage authorization value (0 for none)                         |
| in  | <i>parent_key_authorization_value</i>     | local key usage authorization (i.e., password)                                   |
| in  | <i>salt_size</i>                          | size of cryptographic salt                                                       |
| in  | <i>salt_data</i>                          | random data for cryptographic salt                                               |
| out | <i>key_handle</i>                         | return key handle of derived key for later use in other functions                |

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>        | Given algorithm or algorithm mode not available                             |
| <i>EVITA_INVALID_KEY_SIZE</i>       | Given key size is invalid (e.g., too small or too large)                    |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key                               |
| <i>EVITA_INVALID_KEY_FLAG</i>       | Given key flag is invalid (e.g., wrong combination)                         |
| <i>EVITA_WRONG_AUTHORIZATION</i>    | Given authorization structure does not fit key flags                        |
| <i>EVITA_WRONG_KEY_HANDLE</i>       | Given key handle is unknown or wrong (e.g., not for this algorithm or mode) |
| <i>EVITA_AUTHORIZATION_FAILED</i>   | Given authorization value was wrong                                         |

## Note

Definition at line 325 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.3 Create\_Dh\_Key()

```
ehsm_uint32_t Create_Dh_Key (
 ehsm_uint32_t target_algorithm,
 ehsm_uint32_t key_size,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t key_usages_size,
 ehsm_uint8_t * key_usages_data,
 ehsm_uint32_t local_key_handle,
 ehsm_uint32_t local_key_auth_size,
 ehsm_uint8_t * local_key_auth_value,
 ehsm_uint32_t remote_key_handle,
 ehsm_uint32_t remote_key_auth_size,
 ehsm_uint8_t * remote_key_auth_value,
 ehsm_uint32_t * key_handle)
```

Create a Diffie-Hellman key according to the specified algorithm.

## Parameters

|     |                              |                                                                                                                                               |
|-----|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>target_algorithm</i>      | reference to target algorithm for key generation/usage based on DH outputs, otherwise = 0                                                     |
| in  | <i>key_size</i>              | if algorithm_identifier = 0 then define key length in bits, otherwise ignored; must not be greater than DH parent key lengths otherwise error |
| in  | <i>valid_until</i>           | define key life limitation as UTC time                                                                                                        |
| in  | <i>type</i>                  | memory_target non-volatile or RAM                                                                                                             |
| in  | <i>key_usages_size</i>       | key usages size                                                                                                                               |
| in  | <i>key_usages_data</i>       | key usages data, refers to <a href="#">ehsm_key_usages_st</a>                                                                                 |
| in  | <i>local_key_handle</i>      | refers to internal private key that will be used by DH                                                                                        |
| in  | <i>local_key_auth_size</i>   | size of local key usage authorization value (0 for none)                                                                                      |
| in  | <i>local_key_auth_value</i>  | local key usage authorization (i.e.,password)                                                                                                 |
| in  | <i>remote_key_handle</i>     | refers to public remote key that will be used by DH                                                                                           |
| in  | <i>remote_key_auth_size</i>  | size of remote key usage authorization value (0 for none)                                                                                     |
| in  | <i>remote_key_auth_value</i> | remote key usage authorization (i.e.,password)                                                                                                |
| out | <i>key_handle</i>            | return key handle of DH-calculated shared secret for later use in other functions                                                             |

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                                                                                                  |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <i>EVITA_OK</i>                      | Request successful                                                                                               |
| <i>EVITA_ALGORITHM_ERROR</i>         | Given algorithm or algorithm mode not available                                                                  |
| <i>EVITA_INVALID_KEY_SIZE</i>        | Given key size is invalid (e.g., too small or too large)                                                         |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i>  | No resources left to create an additional key                                                                    |
| <i>EVITA_INVALID_KEY_FLAG</i>        | Given key flag is invalid (e.g., wrong combination)                                                              |
| <i>EVITA_WRONG_AUTHORIZATION</i>     | Given authorization structure does not fit key flags (e.g., authorization definition for a certain flag missing) |
| <i>EVITA_WRONG_KEY_HANDLE</i>        | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)                                      |
| <i>EVITA_WRONG_REMOTE_KEY_HANDLE</i> | Given remote key handle is unknown or wrong (e.g., not for this algorithm or this mode)                          |
| <i>EVITA_WRONG_KEY_COMBINATION</i>   | Keys do not fit the algorithm (e.g., RSA key vs. ECDH)                                                           |
| <i>EVITA_AUTHORIZATION_FAILED</i>    | Given authorization value was wrong                                                                              |

## Note

Definition at line 174 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.4 Create\_Random\_Dh\_Key\_Pair()

```
ehsm_uint32_t Create_Random_Dh_Key_Pair (
 ehsm_uint32_t key_size,
```

```

ehsm_uint32_t valid_until,
ehsm_key_mem_type_e type,
ehsm_uint32_t key_usages_size,
ehsm_uint8_t * key_usages_data,
ehsm_uint8_t * p,
ehsm_uint32_t p_size,
ehsm_uint8_t * q,
ehsm_uint32_t q_size,
ehsm_uint8_t * g,
ehsm_uint32_t g_size,
ehsm_uint32_t * key_handle)

```

This function is used for creating a DH key pair.

#### Parameters

|     |                        |                                                                                  |
|-----|------------------------|----------------------------------------------------------------------------------|
| in  | <i>key_size</i>        | Must not be 0 if a key shorter than KDF output is allowed, otherwise error.      |
| in  | <i>valid_until</i>     | define key life limitation as UTC time                                           |
| in  | <i>type</i>            | memory_target non-volatile or RAM                                                |
| in  | <i>key_usages_size</i> | key usages size                                                                  |
| in  | <i>key_usages_data</i> | key usages data to define set of allowed key usages, cf. data structures section |
| in  | <i>p</i>               | a prime defining the GF(p)                                                       |
| in  | <i>p_size</i>          | size of q                                                                        |
| in  | <i>q</i>               | a prime factor of p-1, aka order of g                                            |
| in  | <i>q_size</i>          | size of q                                                                        |
| in  | <i>g</i>               | a generator of the q-order subgroup of GF(p)*                                    |
| in  | <i>g_size</i>          | size of g                                                                        |
| out | <i>key_handle</i>      | return key handle of created key pair                                            |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                     |                                                          |
|-------------------------------------|----------------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>        | Given algorithm or algorithm mode not available          |
| <i>EVITA_INVALID_KEY_SIZE</i>       | Given key size is invalid (e.g., too small or too large) |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key            |
| <i>EVITA_INVALID_KEY_FLAG</i>       | Given key flag is invalid (e.g., wrong combination)      |
| <i>EVITA_WRONG_AUTHORIZATION</i>    | Given authorization structure does not fit key flags     |

#### Note

Definition at line 125 of file eHSM\_If\_Evita\_Key\_Ip.c.

#### 4.34.1.5 Create\_Random\_Key()

```

ehsm_uint32_t Create_Random_Key (
 ehsm_uint32_t target_algorithm,

```

```
ehsm_uint32_t key_size,
ehsm_key_mem_type_e type,
ehsm_uint32_t key_usages_size,
ehsm_uint8_t * key_usages_data,
ehsm_uint32_t * key_handle)
```

Create a random key according to the specified algorithm.

#### Parameters

|     |                         |                                                                                                                                     |
|-----|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>target_algorithm</i> | reference to target algorithm for key generation/usage based on RNG outputs, otherwise = 0 (cf. hardware interface data structures) |
| in  | <i>key_size</i>         | if algorithm_identifier = 0 (just random string) then define key length in bits, otherwise ignored                                  |
| in  | <i>valid_until</i>      | define key life limitation as UTC time                                                                                              |
| in  | <i>type</i>             | memory_target non-volatile or RAM                                                                                                   |
| in  | <i>key_usages_size</i>  | key usages size                                                                                                                     |
| in  | <i>key_usages_data</i>  | key usages data, refers to <a href="#">ehsm_key_usages_st</a>                                                                       |
| out | <i>key_handle</i>       | return key handle for later use in other functions                                                                                  |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                     |                                                                                                                  |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <i>EVITA_OK</i>                     | Request successful                                                                                               |
| <i>EVITA_ALGORITHM_ERROR</i>        | Given algorithm or algorithm mode not available                                                                  |
| <i>EVITA_INVALID_KEY_SIZE</i>       | Given key size is invalid (e.g., too small or too large)                                                         |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key                                                                    |
| <i>EVITA_INVALID_KEY_FLAG</i>       | Given key flag is invalid (e.g., wrong combination)                                                              |
| <i>EVITA_WRONG_AUTHORIZATION</i>    | Given authorization structure does not fit key flags (e.g., authorization definition for a certain flag missing) |

#### Note

Definition at line 88 of file eHSM\_If\_Evita\_Key\_Ip.c.

#### 4.34.1.6 ehsm\_get\_pub\_from\_priv()

```
ehsm_uint32_t ehsm_get_pub_from_priv (
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization,
 ehsm_uint8_t * pub_value,
 ehsm_uint32_t * pub_size,
 ehsm_uint32_t priv_key_algo)
```

The function use to get key public party that calculation from private party.



## Parameters

|     |                               |                                                       |
|-----|-------------------------------|-------------------------------------------------------|
| in  | <i>key_handle</i>             | reference to the internal key used for get public key |
| in  | <i>key_authorization_size</i> | size of key usage authorization                       |
| in  | <i>key_authorization</i>      | key usage authorization (i.e.,password)               |
| out | <i>pub_value</i>              | the output public key                                 |
|     | <i>[in/out]</i>               | pub_size the size of public key                       |
| in  | <i>priv_key_algo</i>          | the private key algorithm(only support sm2/ecc)       |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                 |
|-----------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error                                        |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.               |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | This function is called with no authentication. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Some algorithm self test failed.                |

Definition at line 531 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.7 Key\_Export()

```
ehsm_uint32_t Key_Export (
 ehsm_uint32_t key_handle,
 key_act_use_flags_t * use_flags,
 ehsm_uint32_t transport_key_handle,
 ehsm_uint32_t transport_key_authorization_size,
 ehsm_uint8_t * transport_key_authorization,
 ehsm_uint32_t authenticity_key_handle,
 ehsm_uint32_t authenticity_key_authorization_size,
 ehsm_uint8_t * authenticity_key_authorization,
 ehsm_uint32_t * encrypted_key_size,
 ehsm_uint8_t * encrypted_key,
 ehsm_uint32_t * key_authenticity_code_size,
 ehsm_uint8_t * key_authenticity_code)
```

This function is used for exporting keys from eHSM.

## Parameters

|    |                                            |                                                                                  |
|----|--------------------------------------------|----------------------------------------------------------------------------------|
| in | <i>key_handle</i>                          | reference to the internal key that becomes exported                              |
| in | <i>use_flags</i>                           | define set of key usages to become exported                                      |
| in | <i>transport_key_handle</i>                | reference to the key used for transport protection                               |
| in | <i>transport_key_authorization_size</i>    | size of transport key usage authorization                                        |
| in | <i>transport_key_authorization</i>         | transport key usage authorization (i.e., password)                               |
| in | <i>authenticity_key_handle</i>             | reference to the key used for authenticity code verification (use_flag = verify) |
| in | <i>authenticity_key_authorization_size</i> | size of authenticity key usage authorization                                     |

## Parameters

|     |                                       |                                                                                          |
|-----|---------------------------------------|------------------------------------------------------------------------------------------|
| in  | <i>authenticity_key_authorization</i> | authenticity key usage authorization (i.e.,password)                                     |
| out | <i>encrypted_key_size</i>             | returned encrypted key blob size                                                         |
| out | <i>encrypted_key</i>                  | returned encrypted key blob                                                              |
| out | <i>key_authenticity_code_size</i>     | size of key authenticity code (signature or MAC) created by transport key                |
| out | <i>key_authenticity_code</i>          | key authenticity code (signature or MAC) to enforce and proof module internal protection |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                                                                  |
|-----------------------------------|--------------------------------------------------------------------------------------------------|
| <i>EVITA_AUTHORIZATION_FAILED</i> | Given authorization value was wrong                                                              |
| <i>EVITA_WRONG_KEY_HANDLE</i>     | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)                      |
| <i>EVITA_TRANSPORT_IMPOSSIBLE</i> | Given transport key is not a capable transport key or use flag cannot be transported or migrated |

## Note

Definition at line 270 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.8 Key\_Import()

```
ehsm_uint32_t Key_Import (
 ehsm_uint32_t transport_key_handle,
 ehsm_uint32_t transport_key_authorization_size,
 ehsm_uint8_t * transport_key_authorization,
 ehsm_uint32_t authenticity_key_handle,
 ehsm_uint32_t authenticity_key_authorization_size,
 ehsm_uint8_t * authenticity_key_authorization,
 ehsm_key_mem_type_e type,
 ehsm_uint32_t encrypted_key_size,
 ehsm_uint8_t * encrypted_key,
 ehsm_uint32_t key_authenticity_code_size,
 ehsm_uint8_t * key_authenticity_code,
 ehsm_uint32_t * key_handle)
```

This function is used for importing keys into eHSM that was were generated and exported by another EVITA module or another external trusted party.

## Parameters

|    |                                         |                                                             |
|----|-----------------------------------------|-------------------------------------------------------------|
| in | <i>transport_key_handle</i>             | reference to the internal key used for transport protection |
| in | <i>transport_key_authorization_size</i> | size of transport key usage authorization                   |
| in | <i>transport_key_authorization</i>      | transport key usage authorization (i.e.,password)           |

## Parameters

|     |                                            |                                                                                  |
|-----|--------------------------------------------|----------------------------------------------------------------------------------|
| in  | <i>authenticity_key_handle</i>             | reference to the key used for authenticity code verification (use_flag = verify) |
| in  | <i>authenticity_key_authorization_size</i> | size of authenticity key usage authorization                                     |
| in  | <i>authenticity_key_authorization</i>      | authenticity key usage authorization (i.e., password)                            |
| in  | <i>type</i>                                | memory_target {nv ram}                                                           |
| in  | <i>encrypted_key_size</i>                  | given encrypted key blob size                                                    |
| in  | <i>encrypted_key</i>                       | given encrypted key blob                                                         |
| in  | <i>key_authenticity_code_size</i>          | given key authenticity code (signature or MAC)                                   |
| in  | <i>key_authenticity_code</i>               | given key authenticity code (signature or MAC)                                   |
| out | <i>key_handle</i>                          | reference to the (now) internal key that was imported                            |

## Returns

ehsm\_uint32\_t

## Return values

|                                     |                                                                                                  |
|-------------------------------------|--------------------------------------------------------------------------------------------------|
| <i>EVITA_AUTHORIZATION_FAILED</i>   | Given authorization value was wrong                                                              |
| <i>EVITA_WRONG_KEY_HANDLE</i>       | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)                      |
| <i>EVITA_TRANSPORT_IMPOSSIBLE</i>   | Given transport key is not a capable transport key or use flag cannot be transported or migrated |
| <i>EVITA_ALL_KEY_SPACE_OCCUPIED</i> | No resources left to create an additional key                                                    |

## Note

Definition at line 230 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.9 Key\_Remove()

```
ehsm_uint32_t Key_Remove (
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization)
```

This function is used for removing loaded keys from the eHSM.

## Parameters

|    |                               |                                          |
|----|-------------------------------|------------------------------------------|
| in | <i>key_handle</i>             | reference to key that should be removed  |
| in | <i>key_authorization_size</i> | size of key remove authorization         |
| in | <i>key_authorization</i>      | key remove authorization (i.e. password) |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                                             |
|-----------------------------------|-----------------------------------------------------------------------------|
| <i>EVITA_WRONG_KEY_HANDLE</i>     | Given key handle is unknown or wrong (e.g., not for this algorithm or mode) |
| <i>EVITA_AUTHORIZATION_FAILED</i> | Given authorization value was wrong                                         |
| <i>EVITA_REMOVE_IMPOSSIBLE</i>    | Key is not allowed to become removed                                        |

## Note

Definition at line 380 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.10 Key\_Status()

```
ehsm_uint32_t Key_Status (
 ehsm_uint32_t key_handle,
 ehsm_uint32_t certification_key_handle,
 ehsm_uint32_t certification_key_authorization_size,
 ehsm_uint8_t * certification_key_authorization,
 ehsm_uint32_t * key_status_size,
 ehsm_uint8_t * key_status)
```

This function is used for obtaining all public properties about a key loaded by the eHSM.

## Parameters

|     |                                             |                                                                                  |
|-----|---------------------------------------------|----------------------------------------------------------------------------------|
| in  | <i>key_handle</i>                           | reference to key whose status is to be returned                                  |
| in  | <i>certification_key_handle</i>             | reference to key that should sign the returned key status (NULL = w/o signature) |
| in  | <i>certification_key_authorization_size</i> | size of certification key usage authorization                                    |
| in  | <i>certification_key_authorization</i>      | certification key usage authorization (i.e. password)                            |
| out | <i>key_status_size</i>                      | size of (certified) key status                                                   |
| out | <i>key_status</i>                           | (certified) key status = { public info about key    signature (optional) }       |

## Returns

ehsm\_uint32\_t

## Return values

|                                    |                                                                                    |
|------------------------------------|------------------------------------------------------------------------------------|
| <i>EVITA_WRONG_KEY_HANDLE</i>      | Given key handle is unknown or wrong (e.g., not for this algorithm or mode)        |
| <i>EVITA_WRONG_CERT_KEY_HANDLE</i> | Given certification key handle is unknown or wrong (e.g., not enabled for signing) |
| <i>EVITA_AUTHORIZATION_FAILED</i>  | Given authorization value was wrong                                                |

## Note

Definition at line 408 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.34.1.11 Module\_Status()

```
ehsm_uint32_t Module_Status (
 ehsm_uint32_t flag,
 ehsm_uint32_t alg,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_auth_size,
 const ehsm_uint8_t * key_auth_value,
 ehsm_uint32_t * status_size,
 ehsm_uint8_t * status,
 ehsm_uint32_t * sign_size,
 ehsm_uint8_t * sign)
```

Get the status of eHSM.

## Parameters

|    |                       |                                                                  |
|----|-----------------------|------------------------------------------------------------------|
| in | <i>flag</i>           | Indicate type of module status info to be returned.              |
| in | <i>alg</i>            | Reference to signing algorithm (could also be a MAC scheme).     |
| in | <i>key_handle</i>     | Reference key that signs returned status (NULL = w/o signature). |
| in | <i>key_auth_size</i>  | Size of key usage authorization value (0 for none).              |
| in | <i>key_auth_value</i> | Key usage authorization (i.e., password).                        |
|    | [in/out]              | status_size: Size of module status structure.                    |
|    | [in/out]              | status: Module status structure data.                            |
|    | [in/out]              | sign_size: Status signature size (if requested, otherwise 0).    |
|    | [in/out]              | sign: Status signature (if requested, otherwise NULL).           |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                                                    |
|----------------------------------------|--------------------------------------------------------------------|
| <i>EVITA_OK</i> (value)                | 0) For success.                                                    |
| <i>EVITA_STATUS_TYPE_NOT_AVAILABLE</i> | Requested status type is (currently) not available for this module |
| <i>EVITA_WRONG_KEY_HANDLE</i>          | Given key handle is unknown or wrong                               |
| <i>EVITA_AUTHORIZATION_FAILED</i>      | Given authorization value was wrong                                |

Definition at line 493 of file eHSM\_If\_Evita\_Key\_Ip.c.

## 4.35 eHSM\_If\_Evita\_Rng\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
```

```
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
```

## Functions

- [ehsm\\_uint32\\_t RNG\\_Get\\_Random](#) ([ehsm\\_uint32\\_t](#) algorithm\_identifier, [ehsm\\_uint32\\_t](#) random\_byte\_request\_size, [ehsm\\_uint8\\_t](#) \*random\_bytes)

*Generate the random number.*

### 4.35.1 Function Documentation

#### 4.35.1.1 RNG\_Get\_Random()

```
ehsm_uint32_t RNG_Get_Random (
 ehsm_uint32_t algorithm_identifier,
 ehsm_uint32_t random_byte_request_size,
 ehsm_uint8_t * random_bytes)
```

Generate the random number.

#### Parameters

|    |                                 |                                                                         |
|----|---------------------------------|-------------------------------------------------------------------------|
| in | <i>algorithm_identifier</i>     | reference to associated (pseudo) random algorithm (e.g., CTRDRBG, TRNG) |
| in | <i>random_byte_request_size</i> | number of random bytes to be returned                                   |
| in | <i>random_bytes</i>             | returned random bytes                                                   |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                    |                                                     |
|------------------------------------|-----------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>       | Given algorithm or algorithm mode not available     |
| <i>EVITA_PRNG_REQUEST_OVERSIZE</i> | Requested number of random bytes exceeds PRNG limit |

#### Returns

EVITA\_TRNG\_SEED\_FAILURE PRNG was unable to retrieve true random seed from TRNG

#### Note

Definition at line 54 of file eHSM\_If\_Evita\_Rng\_Ip.c.

## 4.36 eHSM\_If\_Evita\_SymCper\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Debug_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_Mgr_Ctx_Ip.h"
```

### Functions

- [ehsm\\_uint32\\_t Cipher\\_Init](#) ([ehsm\\_uint32\\_t](#) algorithm\_identifier, [cipher\\_mode\\_e](#) cipher\_mode, [operation\\_mode\\_e](#) operation\_mode, [padding\\_scheme\\_e](#) padding, [ehsm\\_uint32\\_t](#) total\_message\_length, [ehsm\\_uint32\\_t](#) iv\_size, [ehsm\\_uint8\\_t](#) \*iv, [ehsm\\_uint32\\_t](#) key\_handle, [ehsm\\_uint32\\_t](#) key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*key\_authorization\_value, [ehsm\\_ctx\\_session\\_st](#) \*session\_handle)  
*EVITA cipher initialization, aimed to prepare all the data that used in the [Cipher\\_Process\(\)](#) later.*
- [ehsm\\_uint32\\_t Cipher\\_Process](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) input\_data\_size, const [ehsm\\_uint8\\_t](#) \*input\_data, [ehsm\\_uint32\\_t](#) \*output\_data\_size, [ehsm\\_uint8\\_t](#) \*output\_data)  
*EVITA cipher process, aimed to process one or more block within the encryption and decryption process of longer messages.*
- [ehsm\\_uint32\\_t Cipher\\_Finish](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) \*output\_data\_size, [ehsm\\_uint8\\_t](#) \*output\_data)  
*EVITA cipher finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by [Cipher\\_Process\(\)](#).*
- [ehsm\\_uint32\\_t Aead\\_Init](#) ([ehsm\\_uint32\\_t](#) algorithm\_identifier, [cipher\\_mode\\_e](#) cipher\_mode, [operation\\_mode\\_e](#) operation\_mode, [ehsm\\_uint32\\_t](#) total\_message\_length, [ehsm\\_uint32\\_t](#) iv\_size, [ehsm\\_uint8\\_t](#) \*iv, [ehsm\\_uint32\\_t](#) aad\_size, [ehsm\\_uint32\\_t](#) tag\_size, [ehsm\\_uint32\\_t](#) key\_handle, [ehsm\\_uint32\\_t](#) key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*key\_authorization\_value, [ehsm\\_ctx\\_session\\_st](#) \*session\_handle)  
*EVITA aead initialization, aimed to prepare all the data that used in the [Aead\\_Process\(\)](#) later.*
- [ehsm\\_uint32\\_t Aead\\_Process](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) input\_data\_size, const [ehsm\\_uint8\\_t](#) \*input\_data, [ehsm\\_uint32\\_t](#) aad\_size, const [ehsm\\_uint8\\_t](#) \*aad, [ehsm\\_uint32\\_t](#) \*output\_data\_size, [ehsm\\_uint8\\_t](#) \*output\_data)  
*EVITA aead process, aimed to process one or more block within the encryption and decryption process of longer messages.*
- [ehsm\\_uint32\\_t Aead\\_Finish](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) \*output\_data\_size, [ehsm\\_uint8\\_t](#) \*output\_data, [ehsm\\_uint32\\_t](#) \*match)  
*EVITA aead finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by [Aead\\_Process\(\)](#).*
- [ehsm\\_uint32\\_t MAC\\_Init](#) ([ehsm\\_uint32\\_t](#) algorithm\_identifier, [mac\\_mode\\_e](#) mac\_mode, [operation\\_mode\\_e](#) operation\_mode, [padding\\_scheme\\_e](#) padding\_scheme, [ehsm\\_uint32\\_t](#) total\_message\_length, [ehsm\\_uint32\\_t](#) mac\_length, [ehsm\\_uint32\\_t](#) key\_handle, [ehsm\\_uint32\\_t](#) key\_authorization\_size, [ehsm\\_uint8\\_t](#) \*key\_authorization\_value, [ehsm\\_ctx\\_session\\_st](#) \*session\_handle)  
*EVITA MAC initialization, aimed to prepare all the data that used in the [MAC\\_Update\(\)](#) later.*
- [ehsm\\_uint32\\_t MAC\\_Update](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) chunk\_size, [ehsm\\_uint8\\_t](#) \*chunk\_data)  
*EVITA MAC update, aimed to process one or more block within a MAC generation or verification process of longer messages.*
- [ehsm\\_uint32\\_t MAC\\_Finish](#) ([ehsm\\_ctx\\_session\\_st](#) \*session\_handle, [ehsm\\_uint32\\_t](#) \*mac\_size, [mac\\_st](#) \*mac, [ehsm\\_bool\\_t](#) \*mac\_match)  
*EVITA MAC finish, aimed to terminate the MAC generation or verification process after the last block has been processed by [MAC\\_Update](#).*

### 4.36.1 Function Documentation

#### 4.36.1.1 Aead\_Finish()

```
ehsm_uint32_t Aead_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data,
 ehsm_uint32_t * match)
```

EVITA aead finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by [Aead\\_Process\(\)](#).

##### Parameters

|     |                         |                                                                                                                |
|-----|-------------------------|----------------------------------------------------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session handle to be released.                                                                                 |
| out | <i>output_data_size</i> | Size of last output data(can be 0).                                                                            |
| out | <i>output_data</i>      | Last encrypted or decrypted output data(e.g., due to padding scheme).                                          |
|     | <i>[in/out]</i>         | match Only valid for aead verification. If verification success, match is 1. Otherwise, this value is invalid. |

##### Returns

ehsm\_uint32\_t

##### Return values

|                                |                         |
|--------------------------------|-------------------------|
| <a href="#">EVITA_OK(value</a> | 0) For success. And the |
|--------------------------------|-------------------------|

##### Parameters

|              |                                  |
|--------------|----------------------------------|
| <i>match</i> | will be updated with the result. |
|--------------|----------------------------------|

##### Return values

|                                            |                                          |
|--------------------------------------------|------------------------------------------|
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given session handle is unknown or wrong |
|--------------------------------------------|------------------------------------------|

Definition at line 866 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

#### 4.36.1.2 Aead\_Init()

```
ehsm_uint32_t Aead_Init (
 ehsm_uint32_t algorithm_identifier,
 cipher_mode_e cipher_mode,
 operation_mode_e operation_mode,
 ehsm_uint32_t total_message_length,
```



```

ehsm_uint32_t iv_size,
ehsm_uint8_t * iv,
ehsm_uint32_t aad_size,
ehsm_uint32_t tag_size,
ehsm_uint32_t key_handle,
ehsm_uint32_t key_authorization_size,
ehsm_uint8_t * key_authorization_value,
ehsm_ctx_session_st * session_handle)

```

EVITA aead initialization, aimed to prepare all the data that used in the [Aead\\_Process\(\)](#) later.

#### Parameters

|     |                                          |                                                                           |
|-----|------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>              | Reference to associated symmetric algorithm.                              |
| in  | <i>cipher_mode</i>                       | Indicate decryption pr encryption mode.                                   |
| in  | <i>operation_mode</i>                    | Indicate cipher mode of operation.                                        |
| in  | <i>total_message_length</i>              | Give total message length(can be req. by padding scheme).                 |
| in  | <i>iv_size</i>                           | Size of given initialization vector(can be 0).                            |
| in  | <i>iv</i>                                | Set initialization vector(it's public).                                   |
| in  | <i>key_handle</i>                        | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>            | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>           | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle-&gt;session_id</i>     | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i> | Maxium size of a chunk on <a href="#">Cipher_Process()</a> .              |
| out | <i>session_handle-&gt;block_sz</i>       | chunk has to be a multiple of this block size(1 to max).                  |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                             |                                                                                         |
|---------------------------------------------|-----------------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>             | 0) For success.                                                                         |
| <a href="#">EVITA_WRONG_KEY_HANDLE</a>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode)        |
| <a href="#">EVITA_ALL_SESSIONS_OCCUPIED</a> | No resources left for an additional parallel session (or no parallel processing at all) |
| <a href="#">EVITA_ALGORITHM_ERROR</a>       | Given algorithm or algorithm mode not available                                         |
| <a href="#">EVITA_WRONG_IV</a>              | Given IV does not fit the given algorithm                                               |
| <a href="#">EVITA_AUTHORIZATION_FAILE</a>   | Given authorization value was wrong                                                     |

#### Note

iv\_size should be 12 (96 bits)

Definition at line 642 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

#### 4.36.1.3 Aead\_Process()

```

ehsm_uint32_t Aead_Process (
 ehsm_ctx_session_st * session_handle,

```

```
ehsm_uint32_t input_data_size,
const ehsm_uint8_t * input_data,
ehsm_uint32_t aad_size,
const ehsm_uint8_t * aad,
ehsm_uint32_t * output_data_size,
ehsm_uint8_t * output_data)
```

EVITA aead process, aimed to process one or more block within the encryption and decryption process of longer messages.

#### Parameters

|     |                         |                                                                                |
|-----|-------------------------|--------------------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session reference from cipher_Init().                                          |
| in  | <i>input_data_size</i>  | Size of input data.                                                            |
| in  | <i>input_data</i>       | Input data to be encrypted or decrypted.                                       |
| in  | <i>aad_size</i>         | Input aad size. Can be 0.                                                      |
| in  | <i>aad</i>              | Input aad. Can be null.                                                        |
| out | <i>output_data_size</i> | Size of output data(can be different to input size due to padding/de-padding). |
| out | <i>output_data</i>      | Encrypted or decrypted output data.                                            |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                   |                                                           |
|-----------------------------------|-----------------------------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given session handle is unknown or wrong                  |
| <i>EVITA_WRONG_CHUNK_SIZE</i>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 796 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

#### 4.36.1.4 Cipher\_Finish()

```
ehsm_uint32_t Cipher_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data)
```

EVITA cipher finish, aimed to terminate the encryption or decryption process after the last input block to become encrypted or decrypted has been processed by [Cipher\\_Process\(\)](#).

#### Parameters

|     |                         |                                                                       |
|-----|-------------------------|-----------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session handle to be released.                                        |
| out | <i>output_data_size</i> | Size of last output data(can be 0).                                   |
| out | <i>output_data</i>      | Last encrypted or decrypted output data(e.g., due to padding scheme). |

#### Returns

ehsm\_uint32\_t

## Return values

|                                            |                                          |
|--------------------------------------------|------------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                          |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given session handle is unknown or wrong |

Definition at line 490 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.36.1.5 Cipher\_Init()

```
ehsm_uint32_t Cipher_Init (
 ehsm_uint32_t algorithm_identifier,
 cipher_mode_e cipher_mode,
 operation_mode_e operation_mode,
 padding_scheme_e padding,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t iv_size,
 ehsm_uint8_t * iv,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA cipher initialization, aimed to prepare all the data that used in the [Cipher\\_Process\(\)](#) later.

## Parameters

|     |                                          |                                                                           |
|-----|------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>              | Reference to associated symmetric algorithm.                              |
| in  | <i>cipher_mode</i>                       | Indicate decryption pr encryption mode.                                   |
| in  | <i>operation_mode</i>                    | Indicate cipher mode of operation.                                        |
| in  | <i>padding</i>                           | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>              | Give total message length(can be req. by padding scheme).                 |
| in  | <i>iv_size</i>                           | Size of given initialization vector(can be 0).                            |
| in  | <i>iv</i>                                | Set initialization vector(it's public).                                   |
| in  | <i>key_handle</i>                        | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>            | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>           | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle-&gt;session_id</i>     | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i> | Maxium size of a chunk on <a href="#">Cipher_Process()</a> .              |
| out | <i>session_handle-&gt;block_sz</i>       | chunk has to be a multiple of this block size(1 to max).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                             |                                                                                         |
|---------------------------------------------|-----------------------------------------------------------------------------------------|
| <a href="#">EVITA_OK(value)</a>             | 0) For success.                                                                         |
| <a href="#">EVITA_WRONG_KEY_HANDLE</a>      | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode)        |
| <a href="#">EVITA_ALL_SESSIONS_OCCUPIED</a> | No resources left for an additional parallel session (or no parallel processing at all) |

## Return values

|                                  |                                                 |
|----------------------------------|-------------------------------------------------|
| <i>EVITA_ALGORITHM_ERROR</i>     | Given algorithm or algorithm mode not available |
| <i>EVITA_WRONG_IV</i>            | Given IV does not fit the given algorithm       |
| <i>EVITA_AUTHORIZATION_FAILE</i> | Given authorization value was wron              |

Definition at line 233 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.36.1.6 Cipher\_Process()

```
ehsm_uint32_t Cipher_Process (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t input_data_size,
 const ehsm_uint8_t * input_data,
 ehsm_uint32_t * output_data_size,
 ehsm_uint8_t * output_data)
```

EVITA cipher process, aimed to process one or more block within the encryption and decryption process of longer messages.

## Parameters

|     |                         |                                                                                |
|-----|-------------------------|--------------------------------------------------------------------------------|
| in  | <i>session_handle</i>   | Session reference from cipher_Init().                                          |
| in  | <i>input_data_size</i>  | Size of input data.                                                            |
| in  | <i>input_data</i>       | Input data to be encrypted or decrypted.                                       |
| out | <i>output_data_size</i> | Size of output data(can be different to input size due to padding/de-padding). |
| out | <i>output_data</i>      | Encrypted or decrypted output data.                                            |

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                           |
|-----------------------------------|-----------------------------------------------------------|
| <i>EVITA_OK(value)</i>            | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i> | Given session handle is unknown or wrong                  |
| <i>EVITA_WRONG_CHUNK_SIZE</i>     | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 423 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.36.1.7 MAC\_Finish()

```
ehsm_uint32_t MAC_Finish (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t * mac_size,
 mac_st * mac,
 ehsm_bool_t * mac_match)
```

EVITA MAC finish, aimed to terminate the MAC generation or verification process after the last block has been processed by MAC\_Update.

## Parameters

|     |                       |                                                                                                 |
|-----|-----------------------|-------------------------------------------------------------------------------------------------|
| in  | <i>session_handle</i> | Session handle to be released.                                                                  |
|     | <i>[in/out]</i>       | mac_size Size of Data structure, as an input in verify mode and as an output in sign mode.      |
|     | <i>[in/out]</i>       | mac Data structure include mac_size and mac value(optional with UTC_time_stamp in verify mode). |
| out | <i>mac_match</i>      | True if calculated MAC matches the given reference MAC(only for verify mode).                   |

## Returns

ehsm\_uint32\_t

## Return values

|                                            |                                          |
|--------------------------------------------|------------------------------------------|
| <a href="#">EVITA_OK(value)</a>            | 0) For success.                          |
| <a href="#">EVITA_WRONG_SESSION_HANDLE</a> | Given session handle is unknown or wrong |

Definition at line 1158 of file eHSM\_Lf\_Evita\_SymCper\_Ip.c.

## 4.36.1.8 MAC\_Init()

```
ehsm_uint32_t MAC_Init (
 ehsm_uint32_t algorithm_identifier,
 mac_mode_e mac_mode,
 operation_mode_e operation_mode,
 padding_scheme_e padding_scheme,
 ehsm_uint32_t total_message_length,
 ehsm_uint32_t mac_length,
 ehsm_uint32_t key_handle,
 ehsm_uint32_t key_authorization_size,
 ehsm_uint8_t * key_authorization_value,
 ehsm_ctx_session_st * session_handle)
```

EVITA MAC initialization, aimed to prepare all the data that used in the [MAC\\_Update\(\)](#) later.

## Parameters

|     |                                               |                                                                           |
|-----|-----------------------------------------------|---------------------------------------------------------------------------|
| in  | <i>algorithm_identifier</i>                   | Reference to associated MAC algorithm.                                    |
| in  | <i>mac_mode{sign timestamped_sign verify}</i> | Indicate MAC creation or verification mode.                               |
| in  | <i>operation_mode</i>                         | Indicate MAC type.                                                        |
| in  | <i>padding_scheme</i>                         | Indicate padding scheme.                                                  |
| in  | <i>total_message_length</i>                   | Give total message length(can be req. by padding scheme).                 |
| in  | <i>mac_length</i>                             | Length of MAC(eg., if MAC < AES block size)                               |
| in  | <i>key_handle</i>                             | Refer to internal key that will be used.                                  |
| in  | <i>key_authorization_size</i>                 | Size of key usage authorization value(0 for none).                        |
| in  | <i>key_authorization_value</i>                | Key usage authorization(i.e., password).                                  |
| out | <i>session_handle</i>                         | Enables interruption & parallel processing and/or session authentication. |
| out | <i>session_handle-&gt;max_chunk_size</i>      | Maxium size of a chunk on update().                                       |
| out | <i>session_handle-&gt;chunk_block_size</i>    | Chunk has to be a multiple of this block size(1 to max).                  |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------|
| <a href="#"><i>EVITA_OK(value)</i></a> | 0) For success.                                                                         |
| <i>EVITA_WRONG_KEY_HANDLE</i>          | Given key handle is unknown or wrong (e.g., not for this algorithm or this mode)        |
| <i>EVITA_MAC_LENGTH_OVERSIZE</i>       | Given MAC length for verification is greater than MAC                                   |
| <i>EVITA_ALL_SESSIONS_OCCUPIED</i>     | No resources left for an additional parallel session (or no parallel processing at all) |
| <i>EVITA_ALGORITHM_ERROR</i>           | Given algorithm or algorithm mode not available                                         |
| <i>EVITA_AUTHORIZATION_FAILE</i>       | Given authorization value was wrong                                                     |

Definition at line 959 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.36.1.9 MAC\_Update()

```
ehsm_uint32_t MAC_Update (
 ehsm_ctx_session_st * session_handle,
 ehsm_uint32_t chunk_size,
 ehsm_uint8_t * chunk_data)
```

EVITA MAC update, aimed to process one or more block within a MAC generation or verification process of longer messages.

## Parameters

|    |                       |                                                                                         |
|----|-----------------------|-----------------------------------------------------------------------------------------|
| in | <i>session_handle</i> | Session reference from <a href="#"><i>MAC_Init()</i></a> .                              |
| in | <i>chunk_size</i>     | Size of data chunk used for MAC update.                                                 |
| in | <i>chunk_data</i>     | Data chunk byte array used for MAC update(data to be verified or protected with a MAC). |

## Returns

ehsm\_uint32\_t

## Return values

|                                        |                                                           |
|----------------------------------------|-----------------------------------------------------------|
| <a href="#"><i>EVITA_OK(value)</i></a> | 0) For success.                                           |
| <i>EVITA_WRONG_SESSION_HANDLE</i>      | Given key handle is unknown or wrong                      |
| <i>EVITA_WRONG_CHUNK_SIZE</i>          | Given chunk size is wrong (cf. returns on initialization) |

Definition at line 1102 of file eHSM\_If\_Evita\_SymCper\_Ip.c.

## 4.37 eHSM\_If\_Evita\_Timer\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
```

```
#include "eHSM_IntCfg_Ip.h"
```

## 4.38 eHSM\_If\_Evita\_Types\_Ip.h File Reference

```
#include "eHSM_Com_Struct_Ip.h"
```

### Classes

- struct [key\\_act\\_use\\_flags\\_t](#)
- struct [ehsm\\_dh\\_param\\_size\\_info](#)
- struct [key\\_info\\_st](#)
- struct [ehsm\\_key\\_attr\\_data\\_](#)
- struct [ehsm\\_key\\_signature\\_](#)
- struct [ehsm\\_ecc\\_pubkey](#)
- struct [ehsm\\_rsa\\_pubkey](#)
- struct [ehsm\\_dh\\_param](#)
- struct [ehsm\\_dh\\_pubkey](#)
- union [ehsm\\_pubkey\\_data\\_](#)
- struct [ehsm\\_dh\\_prikey](#)
- struct [ehsm\\_rsa crt\\_param\\_](#)
- union [ehsm\\_prikey\\_data\\_](#)
- struct [ehsm\\_key\\_status\\_](#)
- struct [ehsm\\_internal\\_key\\_](#)
- struct [ehsm\\_external\\_key\\_](#)
- struct [ehsm\\_export\\_pub\\_key\\_](#)
- struct [ehsm\\_sym\\_key\\_size\\_](#)
- struct [ehsm\\_ecc\\_key\\_size\\_](#)
- struct [ehsm\\_rsa\\_key\\_size\\_](#)
- struct [ehsm\\_rsa\\_dh\\_key\\_size\\_](#)
- struct [ehsm\\_se\\_key\\_](#)
- struct [ehsm\\_pub\\_key\\_](#)
- struct [counter\\_value\\_64\\_t](#)
- struct [hash\\_hmac\\_t](#)
- struct [signature\\_t](#)
- struct [mac\\_t](#)
- struct [ehsm\\_tick\\_value](#)

### Macros

- #define [ALIGN\\_BYTE\(x\) \\_\\_attribute\\_\\_\(\(packed, aligned\(x\)\)\)](#)
- #define [EVITA\\_KEY\\_MAX\\_SIZE](#) (sizeof(ehsm\_internal\_key\_st))
- #define [EVITA\\_SALT\\_VALUE\\_MAX\\_SIZE](#) (64)
- #define [EVITA\\_MAX\\_RANDOM\\_KEY\\_SIZE](#) (1024)
- #define [EVITA\\_OTP\\_SYM\\_KEY\\_SIZE](#) (sizeof(evita\_otp\_sym\_key\_st))
- #define [EVITA\\_OTP\\_ASYM\\_KEY\\_SIZE](#) (sizeof(evita\_otp\_asym\_key\_st))
- #define [EVITA\\_MAX\\_OTP\\_KEY\\_SIZE](#) (EVITA\_OTP\_ASYM\_KEY\_SIZE)
- #define [EVITA\\_KEY\\_SIGNATRUE\\_SIZE](#) (64)
- #define [EVITA\\_OTP\\_PUBKEY\\_SIZE](#) (64)
- #define [EVITA\\_OTP\\_PRIVKEY\\_SIZE](#) (32)
- #define [EVITA\\_OK](#) 0x0U
- #define [EVITA\\_GENERAL\\_ERROR](#) 0x01U



- #define EVITA\_WRONG\_KEY\_HANDLE 0x02U
- #define EVITA\_ALL\_SESSIONS\_OCCUPIED 0x03U
- #define EVITA\_ALGORITHM\_ERROR 0x04U
- #define EVITA\_WRONG\_IV 0x05U
- #define EVITA\_AUTHORIZATION\_FAILED 0x06U
- #define EVITA\_WRONG\_SESSION\_HANDLE 0x07U
- #define EVITA\_WRONG\_CHUNK\_SIZE 0x08U
- #define EVITA\_MAC\_LENGTH\_OVERSIZE 0x09U
- #define EVITA\_WRONG\_ECR\_INDEX 0x0aU
- Given ECR index is not existing or cannot be extended.*
- #define EVITA\_PRNG\_REQUEST\_OVERSIZE 0x0bU
- #define EVITA\_TRNG\_SEED\_FAILURE 0x0cU
- #define EVITA\_ALL\_COUNTERS\_OCCUPIED 0x0dU
- #define EVITA\_UNKNOWN\_COUNTER\_ID 0x0eU
- #define EVITA\_INVALID\_COUNTER\_INCREMENTATION 0x0fU
- #define EVITA\_STATUS\_TYPE\_NOT\_AVAILABLE 0x10U
- #define EVITA\_TEST\_CASE\_NOT\_AVAILABLE 0x11U
- #define EVITA\_TEST\_CASE\_FAILED 0x12U
- #define EVITA\_INVALID\_KEY\_SIZE 0x13U
- #define EVITA\_ALL\_KEY\_SPACE\_OCCUPIED 0x14U
- #define EVITA\_INVALID\_KEY\_FLAG 0x15U
- #define EVITA\_WRONG\_REMOTE\_KEY\_HANDLE 0x16U
- #define EVITA\_WRONG\_KEY\_COMBINATION 0x17U
- #define EVITA\_WRONG\_AUTHORIZATION 0x18U
- #define EVITA\_TRANSPORT\_IMPOSSIBLE 0x19U
- #define EVITA\_REMOVE\_IMPOSSIBLE 0x1aU
- #define EVITA\_INVALID\_MSG\_SIZE 0x1bU
- #define EVITA\_CLOCK\_NOT\_SYNCHRONIZED 0x1cU
- #define EVITA\_INVALID\_TIME\_STAMP 0x1dU
- #define EVITA\_INVALID\_UTC\_TIME 0x1eU
- #define EVITA\_UTC\_CHALLENGE\_EXPIRED 0x1fU
- #define EVITA\_UTC\_SYNCHRONIZATION\_FAILED 0x20U
- #define EVITA\_WRONG\_CERT\_KEY\_HANDLE 0x21U
- #define EVITA\_KEY\_USE\_FLAG\_SIGN 0x1
- #define EVITA\_KEY\_USE\_FLAG\_VERIFY 0x2
- #define EVITA\_KEY\_USE\_FLAG\_ENCRYPT 0x4
- #define EVITA\_KEY\_USE\_FLAG\_DECRYPT 0x8
- #define EVITA\_KEY\_USE\_FLAG\_TIMESTAMP 0x10
- #define EVITA\_KEY\_USE\_FLAG\_SECUREBOOT 0x20
- #define EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE 0x40
- #define EVITA\_KEY\_USE\_FLAG\_KEYCREATION 0x80
- #define EVITA\_KEY\_USE\_FLAG\_UTCSYNC 0x100
- #define EVITA\_KEY\_USE\_FLAG\_TRANSPORT 0x200
- #define EVITA\_KEY\_USE\_FLAG\_REMOVE 0x400
- #define EVITA\_AUTH\_TYPE\_NONE 0x00U
- #define EVITA\_AUTH\_TYPE\_PASSWD 0x01U
- #define EVITA\_KEY\_DERIVE\_KDFX963 0x00U
- #define EVITA\_KEY\_DERIVE\_PBKDF2 0x01U
- #define EVITA\_KEY\_TRNSP\_INI 0x00U
- #define EVITA\_KEY\_TRNSP\_MIG 0x01U
- #define EVITA\_KEY\_TRNSP\_OEM 0x02U
- #define EVITA\_KEY\_TRNSP\_EXT 0x03U
- #define EHSM\_CONTEXT\_SIZE 512
- #define EVITA\_MAX\_CHUNK\_SIZE 1024U
- #define SKE\_CTX\_BUF\_SIZE 108U
- #define RNG\_REQUEST\_MAX 128
- #define EHSM\_KEY\_AUTH\_VALUE\_MAX\_SIZE (32)

- #define [EHSM\\_SYM\\_KEY\\_PAIR\\_MAX\\_SIZE](#) (64)
- #define [EHSM\\_ECC\\_KEY\\_PAIR\\_MAX\\_SIZE](#) (198)
- #define [EHSM\\_RSA\\_KEY\\_PAIR\\_MAX\\_SIZE](#) (2304)
- #define [EHSM\\_RSA\\_DH\\_KEY\\_PAIR\\_MAX\\_SIZE](#) (2304)
- #define [EHSM\\_SM2\\_SM2\\_KEY\\_MAX\\_SIZE](#) (100)
- #define [EHSM\\_KEY\\_DATA\\_MAX\\_SIZE](#) ([EHSM\\_SYM\\_KEY\\_PAIR\\_MAX\\_SIZE](#))
- #define [EHSM\\_KEY\\_DATA\\_MAX\\_SIZE](#) ([EHSM\\_SM2\\_SM2\\_KEY\\_MAX\\_SIZE](#))
- #define [EHSM\\_KEY\\_DATA\\_MAX\\_SIZE](#) ([EHSM\\_ECC\\_KEY\\_PAIR\\_MAX\\_SIZE](#))
- #define [EHSM\\_KEY\\_SIZE\\_INFO\\_MAX\\_LEN](#) (16)
- #define [EHSM\\_KEY\\_HEAD\\_SIZE](#) (sizeof(ehsm\_se\_key\_st))
- #define [EVITA\\_HASH\\_BUF\\_SIZE](#) 64
- #define [EVITA\\_SIGNATURE\\_BUF\\_SIZE](#) 512
- #define [EVITA\\_MAC\\_BUF\\_SIZE](#) 16
- #define [EHSM\\_EVITA\\_KEY\\_NUM](#) 20

## Typedefs

- typedef struct [ehsm\\_dh\\_param\\_size\\_info](#) ehsm\_dh\_param\_size\_info\_st
- typedef struct [ehsm\\_key\\_attr\\_data](#) ehsm\_key\_attr\_data\_st
- typedef struct [ehsm\\_key\\_signature](#) ehsm\_key\_signature\_st
- typedef struct [ehsm\\_ecc\\_pubkey](#) ehsm\_ecc\_pubkey\_st
- typedef struct [ehsm\\_rsa\\_pubkey](#) ehsm\_rsa\_pubkey\_st
- typedef struct [ehsm\\_dh\\_param](#) ehsm\_dh\_param\_st
- typedef struct [ehsm\\_dh\\_pubkey](#) ehsm\_dh\_pubkey\_st
- typedef union [ehsm\\_pubkey\\_data](#) ehsm\_pubkey\_data\_st
- typedef struct [ehsm\\_dh\\_prikey](#) ehsm\_dh\_prikey\_st
- typedef struct [ehsm\\_rsa\\_crt\\_param](#) ehsm\_rsa\_crt\_st
- typedef union [ehsm\\_prikey\\_data](#) ehsm\_prikey\_data\_st
- typedef struct [ehsm\\_sym\\_key\\_size](#) ehsm\_sym\_key\_size\_st
- typedef struct [ehsm\\_ecc\\_key\\_size](#) ehsm\_ecc\_key\_size\_st
- typedef struct [ehsm\\_rsa\\_key\\_size](#) ehsm\_rsa\_key\_size\_st
- typedef struct [ehsm\\_rsa\\_dh\\_key\\_size](#) ehsm\_rsa\_dh\_key\_size\_st
- typedef [ehsm\\_uint32\\_t](#) ehsm\_utc\_time\_t
- typedef struct [counter\\_value\\_64\\_t](#) ehsm\_counter\_value\_st
- typedef struct [hash\\_hmac\\_t](#) hash\_hmac\_st
- typedef struct [signature\\_t](#) signature\_st
- typedef [signature\\_st](#) time\_stamp\_st
- typedef struct [mac\\_t](#) mac\_st
- typedef struct [ehsm\\_tick\\_value](#) ehsm\_tick\_value\_st

## Enumerations

- enum [storage\\_key\\_type\\_e](#) {  
[EVITA\\_STORAGE\\_NONE\\_KEY\\_TYPE](#), [EVITA\\_STORAGE\\_KDF\\_KEY\\_TYPE](#), [EVITA\\_STORAGE\\_RSA\\_KEY\\_T↵](#)  
[YPE](#), [EVITA\\_STORAGE\\_DH\\_KEY\\_TYPE](#),  
[EVITA\\_STORAGE\\_RANDOM\\_KEY\\_TYPE](#), [EVITA\\_STORAGE\\_SM9\\_MASTER\\_KEY\\_TYPE](#), [EVITA\\_STORAGE↵](#)  
[\\_OTP\\_KEY\\_TYPE](#), [EVITA\\_STORAGE\\_DH\\_KEY\\_PAIR\\_TYPE](#) }
- enum [cipher\\_mode\\_e](#) { [EVITA\\_ENCRYPTION](#) = 0U, [EVITA\\_DECRYPTION](#) = 1U }
- enum [operation\\_mode\\_e](#) {  
[EVITA\\_ECB\\_MODE](#) = 1U, [EVITA\\_XTS\\_MODE](#) = 2U, [EVITA\\_CBC\\_MODE](#) = 3U, [EVITA\\_CFB\\_MODE](#) = 4U,  
[EVITA\\_OFB\\_MODE](#) = 5U, [EVITA\\_CTR\\_MODE](#) = 6U, [EVITA\\_CMAC\\_MODE](#) = 7U, [EVITA\\_CBC\\_MAC\\_MODE](#) = 8U,  
[EVITA\\_GMAC\\_MODE](#) = 9U, [EVITA\\_GCM\\_MODE](#) = 10U, [EVITA\\_CCM\\_MODE](#) = 11U }
- enum [padding\\_scheme\\_e](#) { [EVITA\\_NOPADDING](#) = 0U, [EVITA\\_PSASSA\\_PSS](#) = 1U, [EVITA\\_PKCS7](#) = 2U, [EVIT↵](#)  
[A\\_ONEWITHZEROS](#) = 3U }
- enum [session\\_status\\_e](#) { [EVITA\\_INVALID](#) = 0U, [EVITA\\_INIT](#) = 1U, [EVITA\\_UPDATE](#) = 3U, [EVITA\\_FINISH](#) = 7U }

- enum `mac_mode_e` { `EVITA_MAC_SIGN`, `EVITA_MAC_VERIFY`, `EVITA_MAC_TIMESTAMPED_SIGN` }
- enum `hash_mode_e` { `EVITA_HMAC_SIGN`, `EVITA_HMAC_VERIFY`, `EVITA_HMAC_TIMESTAMP_SIGN`, `EVITA_HMAC_TIMESTAMP_VERIFY` }
- enum `ehsm_evita_key_handle_e` {  
`EHSM_KEY_EVITA_MVK` = 1, `EHSM_KEY_EVITA_IDK` = 2, `EHSM_KEY_EVITA_SRK` = 3, `EHSM_KEY_EVITA_CSK` = 4,  
`EHSM_KEY_EVITA_OVK` = 5, `EHSM_KEY_EVITA_END` = `EHSM_KEY_EVITA_OVK` + `EHSM_EVITA_KEY_NUM`  
}

## Functions

- struct `ehsm_key_status_` `ALIGN_BYTE` (4) `ehsm_key_status_st`

## Variables

- `ehsm_uint8_t` `keyId` [4]
- `ehsm_uint32_t` `keyIdSize`
- `ehsm_uint32_t` `algo_id`
- `ehsm_uint32_t` `valid_util`
- `key_act_use_flags_t` `activeUseFlag`
- `ehsm_uint32_t` `mem_location`
- `ehsm_key_signatrue_st` `key_sign_data`
- `ehsm_pubkey_data_st` `pubkey`
- `ehsm_uint32_t` `cert_size`
- `ehsm_uint8_t` `cert_data` [512]
- `ehsm_key_attr_data_st` `attr`
- `ehsm_key_usages_st` `key_usage`
- `ehsm_uint32_t` `prikey_enc_size`
- `ehsm_prikey_data_st` `prikey`
- `ehsm_key_signatrue_st` `key_signatrue`
- `ehsm_internal_key_st` `evita_internal_key`
- `ehsm_key_signatrue_st` `auth_sign_data`
- union {  
`ehsm_uint16_t` `rsa_e_bytes_size`  
`ehsm_uint16_t` `dh_pubkey_bytes_size`  
} `size_info`
- `ehsm_pubkey_data_st` `key`
- `ehsm_uint32_t` `key_handle`
- `ehsm_uint8_t` `auth_size`
- `ehsm_uint8_t` `reserved` [3]
- `ehsm_uint8_t` `auth_value` [`EHSM_KEY_AUTH_VALUE_MAX_SIZE`]
- `ehsm_uint8_t` `key_size_info` [`EHSM_KEY_SIZE_INFO_MAX_LEN`]
- `ehsm_uint8_t` `key_data` [0]
- `ehsm_uint8_t` `key_pub_data` [0]

### 4.38.1 Macro Definition Documentation

#### 4.38.1.1 ALIGN\_BYTE

```
#define ALIGN_BYTE(
 x) __attribute__\(\(packed, aligned\(x\)\)\)
```

Definition at line 14 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.2 EHSM\_CONTEXT\_SIZE

```
#define EHSM_CONTEXT_SIZE 512
```

Definition at line 162 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.3 EHSM\_ECC\_KEY\_PAIR\_MAX\_SIZE

```
#define EHSM_ECC_KEY_PAIR_MAX_SIZE (198)
```

Definition at line 172 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.4 EHSM\_EVITA\_KEY\_NUM

```
#define EHSM_EVITA_KEY_NUM 20
```

Definition at line 202 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.5 EHSM\_KEY\_AUTH\_VALUE\_MAX\_SIZE

```
#define EHSM_KEY_AUTH_VALUE_MAX_SIZE (32)
```

Definition at line 170 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.6 EHSM\_KEY\_DATA\_MAX\_SIZE [1/3]

```
#define EHSM_KEY_DATA_MAX_SIZE (EHSM_SYM_KEY_PAIR_MAX_SIZE)
```

Definition at line 183 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.7 EHSM\_KEY\_DATA\_MAX\_SIZE** [2/3]

```
#define EHSM_KEY_DATA_MAX_SIZE (EHSM_SM2_SM2_KEY_MAX_SIZE)
```

Definition at line 183 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.8 EHSM\_KEY\_DATA\_MAX\_SIZE** [3/3]

```
#define EHSM_KEY_DATA_MAX_SIZE (EHSM_ECC_KEY_PAIR_MAX_SIZE)
```

Definition at line 183 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.9 EHSM\_KEY\_HEAD\_SIZE**

```
#define EHSM_KEY_HEAD_SIZE (sizeof(ehsm_se_key_st))
```

Definition at line 196 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.10 EHSM\_KEY\_SIZE\_INFO\_MAX\_LEN**

```
#define EHSM_KEY_SIZE_INFO_MAX_LEN (16)
```

Definition at line 195 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.11 EHSM\_RSA\_DH\_KEY\_PAIR\_MAX\_SIZE**

```
#define EHSM_RSA_DH_KEY_PAIR_MAX_SIZE (2304)
```

Definition at line 174 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.12 EHSM\_RSA\_KEY\_PAIR\_MAX\_SIZE**

```
#define EHSM_RSA_KEY_PAIR_MAX_SIZE (2304)
```

Definition at line 173 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.13 EHSM\_SM2\_SM2\_KEY\_MAX\_SIZE

```
#define EHSM_SM2_SM2_KEY_MAX_SIZE (100)
```

Definition at line 175 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.14 EHSM\_SYM\_KEY\_PAIR\_MAX\_SIZE

```
#define EHSM_SYM_KEY_PAIR_MAX_SIZE (64)
```

Definition at line 171 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.15 EVITA\_ALGORITHM\_ERROR

```
#define EVITA_ALGORITHM_ERROR 0x04U
```

Definition at line 40 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.16 EVITA\_ALL\_COUNTERS\_OCCUPIED

```
#define EVITA_ALL_COUNTERS_OCCUPIED 0x0dU
```

Definition at line 71 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.17 EVITA\_ALL\_KEY\_SPACE\_OCCUPIED

```
#define EVITA_ALL_KEY_SPACE_OCCUPIED 0x14U
```

Definition at line 92 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.18 EVITA\_ALL\_SESSIONS\_OCCUPIED

```
#define EVITA_ALL_SESSIONS_OCCUPIED 0x03U
```

Definition at line 37 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.19 EVITA\_AUTH\_TYPE\_NONE

```
#define EVITA_AUTH_TYPE_NONE 0x00U
```

Definition at line 149 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.20 EVITA\_AUTH\_TYPE\_PASSWD

```
#define EVITA_AUTH_TYPE_PASSWD 0x01U
```

Definition at line 150 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.21 EVITA\_AUTHORIZATION\_FAILED

```
#define EVITA_AUTHORIZATION_FAILED 0x06U
```

Definition at line 46 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.22 EVITA\_CLOCK\_NOT\_SYNCHRONIZED

```
#define EVITA_CLOCK_NOT_SYNCHRONIZED 0x1cU
```

Definition at line 118 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.23 EVITA\_GENERAL\_ERROR

```
#define EVITA_GENERAL_ERROR 0x01U
```

Definition at line 31 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.24 EVITA\_HASH\_BUF\_SIZE

```
#define EVITA_HASH_BUF_SIZE 64
```

Definition at line 198 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.25 EVITA\_INVALID\_COUNTER\_INCREMENTATION**

```
#define EVITA_INVALID_COUNTER_INCREMENTATION 0x0fU
```

Definition at line 77 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.26 EVITA\_INVALID\_KEY\_FLAG**

```
#define EVITA_INVALID_KEY_FLAG 0x15U
```

Definition at line 95 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.27 EVITA\_INVALID\_KEY\_SIZE**

```
#define EVITA_INVALID_KEY_SIZE 0x13U
```

Definition at line 89 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.28 EVITA\_INVALID\_MSG\_SIZE**

```
#define EVITA_INVALID_MSG_SIZE 0x1bU
```

Definition at line 115 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.29 EVITA\_INVALID\_TIME\_STAMP**

```
#define EVITA_INVALID_TIME_STAMP 0x1dU
```

Definition at line 121 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.30 EVITA\_INVALID\_UTC\_TIME**

```
#define EVITA_INVALID_UTC_TIME 0x1eU
```

Definition at line 124 of file eHSM\_If\_Evita\_Types\_Ip.h.



#### 4.38.1.31 EVITA\_KEY\_DERIVE\_KDFX963

```
#define EVITA_KEY_DERIVE_KDFX963 0x00U
```

Definition at line 153 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.32 EVITA\_KEY\_DERIVE\_PBKDF2

```
#define EVITA_KEY_DERIVE_PBKDF2 0x01U
```

Definition at line 154 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.33 EVITA\_KEY\_MAX\_SIZE

```
#define EVITA_KEY_MAX_SIZE (sizeof(ehsm_internal_key_st))
```

Definition at line 16 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.34 EVITA\_KEY\_SIGNATRUE\_SIZE

```
#define EVITA_KEY_SIGNATRUE_SIZE (64)
```

Definition at line 22 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.35 EVITA\_KEY\_TRNSP\_EXT

```
#define EVITA_KEY_TRNSP_EXT 0x03U
```

Definition at line 160 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.36 EVITA\_KEY\_TRNSP\_INI

```
#define EVITA_KEY_TRNSP_INI 0x00U
```

Definition at line 157 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.37 EVITA\_KEY\_TRNSP\_MIG

```
#define EVITA_KEY_TRNSP_MIG 0x01U
```

Definition at line 158 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.38 EVITA\_KEY\_TRNSP\_OEM

```
#define EVITA_KEY_TRNSP_OEM 0x02U
```

Definition at line 159 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.39 EVITA\_KEY\_USE\_FLAG\_DECRYPT

```
#define EVITA_KEY_USE_FLAG_DECRYPT 0x8
```

Definition at line 139 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.40 EVITA\_KEY\_USE\_FLAG\_ENCRYPT

```
#define EVITA_KEY_USE_FLAG_ENCRYPT 0x4
```

Definition at line 138 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.41 EVITA\_KEY\_USE\_FLAG\_KEYCREATION

```
#define EVITA_KEY_USE_FLAG_KEYCREATION 0x80
```

Definition at line 143 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.42 EVITA\_KEY\_USE\_FLAG\_REMOVE

```
#define EVITA_KEY_USE_FLAG_REMOVE 0x400
```

Definition at line 146 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.43 EVITA\_KEY\_USE\_FLAG\_SECUREBOOT

```
#define EVITA_KEY_USE_FLAG_SECUREBOOT 0x20
```

Definition at line 141 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.44 EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE

```
#define EVITA_KEY_USE_FLAG_SECURESTORAGE 0x40
```

Definition at line 142 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.45 EVITA\_KEY\_USE\_FLAG\_SIGN

```
#define EVITA_KEY_USE_FLAG_SIGN 0x1
```

Definition at line 136 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.46 EVITA\_KEY\_USE\_FLAG\_TIMESTAMP

```
#define EVITA_KEY_USE_FLAG_TIMESTAMP 0x10
```

Definition at line 140 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.47 EVITA\_KEY\_USE\_FLAG\_TRANSPORT

```
#define EVITA_KEY_USE_FLAG_TRANSPORT 0x200
```

Definition at line 145 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.48 EVITA\_KEY\_USE\_FLAG\_UTCSYNC

```
#define EVITA_KEY_USE_FLAG_UTCSYNC 0x100
```

Definition at line 144 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.49 EVITA\_KEY\_USE\_FLAG\_VERIFY

```
#define EVITA_KEY_USE_FLAG_VERIFY 0x2
```

Definition at line 137 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.50 EVITA\_MAC\_BUF\_SIZE

```
#define EVITA_MAC_BUF_SIZE 16
```

Definition at line 200 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.51 EVITA\_MAC\_LENGTH\_OVERSIZE

```
#define EVITA_MAC_LENGTH_OVERSIZE 0x09U
```

Definition at line 55 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.52 EVITA\_MAX\_CHUNK\_SIZE

```
#define EVITA_MAX_CHUNK_SIZE 1024U
```

Definition at line 163 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.53 EVITA\_MAX\_OTP\_KEY\_SIZE

```
#define EVITA_MAX_OTP_KEY_SIZE (EVITA_OTP_ASYM_KEY_SIZE)
```

Definition at line 21 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.54 EVITA\_MAX\_RANDOM\_KEY\_SIZE

```
#define EVITA_MAX_RANDOM_KEY_SIZE (1024)
```

Definition at line 18 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.55 EVITA\_OK

```
#define EVITA_OK 0x0U
```

Definition at line 28 of file eHSM\_If\_Evita\_Types\_lp.h.

#### 4.38.1.56 EVITA\_OTP\_ASYM\_KEY\_SIZE

```
#define EVITA_OTP_ASYM_KEY_SIZE (sizeof(evita_otp_asym_key_st))
```

Definition at line 20 of file eHSM\_If\_Evita\_Types\_lp.h.

#### 4.38.1.57 EVITA\_OTP\_PRIVKEY\_SIZE

```
#define EVITA_OTP_PRIVKEY_SIZE (32)
```

Definition at line 24 of file eHSM\_If\_Evita\_Types\_lp.h.

#### 4.38.1.58 EVITA\_OTP\_PUBKEY\_SIZE

```
#define EVITA_OTP_PUBKEY_SIZE (64)
```

Definition at line 23 of file eHSM\_If\_Evita\_Types\_lp.h.

#### 4.38.1.59 EVITA\_OTP\_SYM\_KEY\_SIZE

```
#define EVITA_OTP_SYM_KEY_SIZE (sizeof(evita_otp_sym_key_st))
```

Definition at line 19 of file eHSM\_If\_Evita\_Types\_lp.h.

#### 4.38.1.60 EVITA\_PRNG\_REQUEST\_OVERSIZE

```
#define EVITA_PRNG_REQUEST_OVERSIZE 0x0bU
```

Definition at line 65 of file eHSM\_If\_Evita\_Types\_lp.h.

**4.38.1.61 EVITA\_REMOVE\_IMPOSSIBLE**

```
#define EVITA_REMOVE_IMPOSSIBLE 0x1aU
```

Definition at line 112 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.62 EVITA\_SALT\_VALUE\_MAX\_SIZE**

```
#define EVITA_SALT_VALUE_MAX_SIZE (64)
```

Definition at line 17 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.63 EVITA\_SIGNATURE\_BUF\_SIZE**

```
#define EVITA_SIGNATURE_BUF_SIZE 512
```

Definition at line 199 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.64 EVITA\_STATUS\_TYPE\_NOT\_AVAILABLE**

```
#define EVITA_STATUS_TYPE_NOT_AVAILABLE 0x10U
```

Definition at line 80 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.65 EVITA\_TEST\_CASE\_FAILED**

```
#define EVITA_TEST_CASE_FAILED 0x12U
```

Definition at line 86 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.66 EVITA\_TEST\_CASE\_NOT\_AVAILABLE**

```
#define EVITA_TEST_CASE_NOT_AVAILABLE 0x11U
```

Definition at line 83 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.67 EVITA\_TRANSPORT\_IMPOSSIBLE**

```
#define EVITA_TRANSPORT_IMPOSSIBLE 0x19U
```

Definition at line 109 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.68 EVITA\_TRNG\_SEED\_FAILURE**

```
#define EVITA_TRNG_SEED_FAILURE 0x0cU
```

Definition at line 68 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.69 EVITA\_UNKNOWN\_COUNTER\_ID**

```
#define EVITA_UNKNOWN_COUNTER_ID 0x0eU
```

Definition at line 74 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.70 EVITA.UTC\_CHALLENGE\_EXPIRED**

```
#define EVITA.UTC_CHALLENGE_EXPIRED 0x1fU
```

Definition at line 127 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.71 EVITA.UTC\_SYNCHRONIZATION\_FAILED**

```
#define EVITA.UTC_SYNCHRONIZATION_FAILED 0x20U
```

Definition at line 130 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.1.72 EVITA\_WRONG\_AUTHORIZATION**

```
#define EVITA_WRONG_AUTHORIZATION 0x18U
```

Definition at line 105 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.73 EVITA\_WRONG\_CERT\_KEY\_HANDLE

```
#define EVITA_WRONG_CERT_KEY_HANDLE 0x21U
```

Definition at line 133 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.74 EVITA\_WRONG\_CHUNK\_SIZE

```
#define EVITA_WRONG_CHUNK_SIZE 0x08U
```

Definition at line 52 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.75 EVITA\_WRONG\_ECR\_INDEX

```
#define EVITA_WRONG_ECR_INDEX 0x0aU
```

Given ECR index is not existing or cannot be extended.

##### Note

This is not supported now.

Definition at line 62 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.76 EVITA\_WRONG\_IV

```
#define EVITA_WRONG_IV 0x05U
```

Definition at line 43 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.77 EVITA\_WRONG\_KEY\_COMBINATION

```
#define EVITA_WRONG_KEY_COMBINATION 0x17U
```

Definition at line 101 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.78 EVITA\_WRONG\_KEY\_HANDLE

```
#define EVITA_WRONG_KEY_HANDLE 0x02U
```

Definition at line 34 of file eHSM\_If\_Evita\_Types\_Ip.h.



#### 4.38.1.79 EVITA\_WRONG\_REMOTE\_KEY\_HANDLE

```
#define EVITA_WRONG_REMOTE_KEY_HANDLE 0x16U
```

Definition at line 98 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.80 EVITA\_WRONG\_SESSION\_HANDLE

```
#define EVITA_WRONG_SESSION_HANDLE 0x07U
```

Definition at line 49 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.81 RNG\_REQUEST\_MAX

```
#define RNG_REQUEST_MAX 128
```

Definition at line 167 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.1.82 SKE\_CTX\_BUF\_SIZE

```
#define SKE_CTX_BUF_SIZE 108U
```

Definition at line 164 of file eHSM\_If\_Evita\_Types\_Ip.h.

### 4.38.2 Typedef Documentation

#### 4.38.2.1 ehsm\_counter\_value\_st

```
typedef struct counter_value_64_t ehsm_counter_value_st
```

#### 4.38.2.2 ehsm\_dh\_param\_size\_info\_st

```
typedef struct ehsm_dh_param_size_info ehsm_dh_param_size_info_st
```

#### 4.38.2.3 ehsm\_dh\_param\_st

```
typedef struct ehsm_dh_param ehsm_dh_param_st
```

#### 4.38.2.4 ehsm\_dh\_prikey\_st

```
typedef struct ehsm_dh_prikey ehsm_dh_prikey_st
```

#### 4.38.2.5 ehsm\_dh\_pubkey\_st

```
typedef struct ehsm_dh_pubkey ehsm_dh_pubkey_st
```

#### 4.38.2.6 ehsm\_ecc\_key\_size\_st

```
typedef struct ehsm_ecc_key_size_ ehsm_ecc_key_size_st
```

#### 4.38.2.7 ehsm\_ecc\_pubkey\_st

```
typedef struct ehsm_ecc_pubkey ehsm_ecc_pubkey_st
```

#### 4.38.2.8 ehsm\_key\_attr\_data\_st

```
typedef struct ehsm_key_attr_data_ ehsm_key_attr_data_st
```

#### 4.38.2.9 ehsm\_key\_signatrue\_st

```
typedef struct ehsm_key_signature_ ehsm_key_signatrue_st
```

#### 4.38.2.10 ehsm\_prikey\_data\_st

```
typedef union ehsm_prikey_data_ ehsm_prikey_data_st
```

**4.38.2.11 ehsm\_pubkey\_data\_st**

```
typedef union ehsm_pubkey_data_ ehsm_pubkey_data_st
```

**4.38.2.12 ehsm\_rsa\_ctr\_st**

```
typedef struct ehsm_rsa_ctr_param_ ehsm_rsa_ctr_st
```

**4.38.2.13 ehsm\_rsa\_dh\_key\_size\_st**

```
typedef struct ehsm_rsa_dh_key_size_ ehsm_rsa_dh_key_size_st
```

**4.38.2.14 ehsm\_rsa\_key\_size\_st**

```
typedef struct ehsm_rsa_key_size_ ehsm_rsa_key_size_st
```

**4.38.2.15 ehsm\_rsa\_pubkey\_st**

```
typedef struct ehsm_rsa_pubkey ehsm_rsa_pubkey_st
```

**4.38.2.16 ehsm\_sym\_key\_size\_st**

```
typedef struct ehsm_sym_key_size_ ehsm_sym_key_size_st
```

**4.38.2.17 ehsm\_tick\_value\_st**

```
typedef struct ehsm_tick_value ehsm_tick_value_st
```

**4.38.2.18 ehsm\_utc\_time\_t**

```
typedef ehsm_uint32_t ehsm_utc_time_t
```

Definition at line 417 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.2.19 hash\_hmac\_st

```
typedef struct hash_hmac_t hash_hmac_st
```

#### 4.38.2.20 mac\_st

```
typedef struct mac_t mac_st
```

#### 4.38.2.21 signature\_st

```
typedef struct signature_t signature_st
```

#### 4.38.2.22 time\_stamp\_st

```
typedef signature_st time_stamp_st
```

Definition at line 439 of file eHSM\_If\_Evita\_Types\_lp.h.

### 4.38.3 Enumeration Type Documentation

#### 4.38.3.1 cipher\_mode\_e

```
enum cipher_mode_e
```

##### Enumerator

|                  |  |
|------------------|--|
| EVITA_ENCRYPTION |  |
| EVITA_DECRYPTION |  |

Definition at line 456 of file eHSM\_If\_Evita\_Types\_lp.h.

#### 4.38.3.2 ehsm\_evita\_key\_handle\_e

```
enum ehsm_evita_key_handle_e
```

## Enumerator

|                    |  |
|--------------------|--|
| EHSM_KEY_EVITA_MVK |  |
| EHSM_KEY_EVITA_IDK |  |
| EHSM_KEY_EVITA_SRK |  |
| EHSM_KEY_EVITA_CSK |  |
| EHSM_KEY_EVITA_OVK |  |
| EHSM_KEY_EVITA_END |  |

Definition at line 509 of file eHSM\_If\_Evita\_Types\_Ip.h.

## 4.38.3.3 hash\_mode\_e

enum [hash\\_mode\\_e](#)

## Enumerator

|                           |  |
|---------------------------|--|
| EVITA_HMAC_SIGN           |  |
| EVITA_HMAC_VERIFY         |  |
| EVITA_HMAC_TIMESTAMP_SIGN |  |
| EVITA_HASH                |  |

Definition at line 501 of file eHSM\_If\_Evita\_Types\_Ip.h.

## 4.38.3.4 mac\_mode\_e

enum [mac\\_mode\\_e](#)

## Enumerator

|                            |  |
|----------------------------|--|
| EVITA_MAC_SIGN             |  |
| EVITA_MAC_VERIFY           |  |
| EVITA_MAC_TIMESTAMPED_SIGN |  |

Definition at line 493 of file eHSM\_If\_Evita\_Types\_Ip.h.

## 4.38.3.5 operation\_mode\_e

enum [operation\\_mode\\_e](#)

## Enumerator

|                |  |
|----------------|--|
| EVITA_ECB_MODE |  |
| EVITA_XTS_MODE |  |

**Enumerator**

|                    |  |
|--------------------|--|
| EVITA_CBC_MODE     |  |
| EVITA_CFB_MODE     |  |
| EVITA_OFB_MODE     |  |
| EVITA_CTR_MODE     |  |
| EVITA_CMAC_MODE    |  |
| EVITA_CBC_MAC_MODE |  |
| EVITA_GMAC_MODE    |  |
| EVITA_GCM_MODE     |  |
| EVITA_GCM_MODE     |  |

Definition at line 462 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.3.6 padding\_scheme\_e**

```
enum padding_scheme_e
```

**Enumerator**

|                    |  |
|--------------------|--|
| EVITA_NOPADDING    |  |
| EVITA_PSASSA_PSS   |  |
| EVITA_PKCS7        |  |
| EVITA_ONEWITHZEROS |  |

Definition at line 477 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.3.7 session\_status\_e**

```
enum session_status_e
```

**Enumerator**

|               |  |
|---------------|--|
| EVITA_INVALID |  |
| EVITA_INIT    |  |
| EVITA_UPDATE  |  |
| EVITA_FINISH  |  |

Definition at line 485 of file eHSM\_If\_Evita\_Types\_Ip.h.

**4.38.3.8 storage\_key\_type\_e**

```
enum storage_key_type_e
```

## Enumerator

|                                   |  |
|-----------------------------------|--|
| EVITA_STORAGE_NONE_KEY_TYPE       |  |
| EVITA_STORAGE_KDF_KEY_TYPE        |  |
| EVITA_STORAGE_RSA_KEY_TYPE        |  |
| EVITA_STORAGE_DH_KEY_TYPE         |  |
| EVITA_STORAGE_RANDOM_KEY_TYPE     |  |
| EVITA_STORAGE_SM9_MASTER_KEY_TYPE |  |
| EVITA_STORAGE_OTP_KEY_TYPE        |  |
| EVITA_STORAGE_DH_KEY_PAIR_TYPE    |  |

Definition at line 227 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.4 Function Documentation

##### 4.38.4.1 ALIGN\_BYTE()

```
struct ehsm_pub_key_ ALIGN_BYTE (
 4)
```

#### 4.38.5 Variable Documentation

##### 4.38.5.1 activeUseFlag

```
key_act_use_flags_t activeUseFlag
```

Definition at line 362 of file eHSM\_If\_Evita\_Types\_Ip.h.

##### 4.38.5.2 algo\_id

```
ehsm_uint32_t algo_id
```

Definition at line 360 of file eHSM\_If\_Evita\_Types\_Ip.h.

##### 4.38.5.3 attr

```
ehsm_key_attr_data_st attr
```

Definition at line 358 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.4 auth\_sign\_data

`ehsm_key_signatrue_st` `auth_sign_data`

Definition at line 359 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.5 auth\_size

`ehsm_uint8_t` `auth_size`

Definition at line 360 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.6 auth\_value

`ehsm_uint8_t` `auth_value`[`EHSM_KEY_AUTH_VALUE_MAX_SIZE`]

Definition at line 362 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.7 cert\_data

`ehsm_uint8_t` `cert_data`[512]

Definition at line 367 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.8 cert\_size

`ehsm_uint32_t` `cert_size`

Definition at line 366 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.9 dh\_pubkey\_bytes\_size

`ehsm_uint16_t` `dh_pubkey_bytes_size`

Definition at line 361 of file eHSM\_If\_Evita\_Types\_Ip.h.



#### 4.38.5.10 evita\_internal\_key

```
ehsm_internal_key_st evita_internal_key
```

Definition at line 358 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.11 key

```
ehsm_pubkey_data_st key
```

Definition at line 363 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.12 key\_data

```
ehsm_uint8_t key_data[0]
```

Definition at line 364 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.13 key\_handle

```
ehsm_uint32_t key_handle
```

Definition at line 358 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.14 key\_pub\_data

```
ehsm_uint8_t key_pub_data[0]
```

Definition at line 360 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.15 key\_sign\_data

```
ehsm_key_signatrue_st key_sign_data
```

Definition at line 364 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.16 key\_signatrue

`ehsm_key_signatrue_st` `key_signatrue`

Definition at line 363 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.17 key\_size\_info

`ehsm_uint8_t` `key_size_info`

Definition at line 363 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.18 key\_usage

`ehsm_key_usages_st` `key_usage`

Definition at line 359 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.19 keyId

`ehsm_uint8_t` `keyId[4]`

Definition at line 358 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.20 keyIdSize

`ehsm_uint32_t` `keyIdSize`

Definition at line 359 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.21 mem\_location

`ehsm_uint32_t` `mem_location`

Definition at line 363 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.22 prikey

`ehsm_prikey_data_st` prikey

Definition at line 362 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.23 prikey\_enc\_size

`ehsm_uint32_t` prikey\_enc\_size

Definition at line 361 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.24 pubkey

`ehsm_pubkey_data_st` pubkey

Definition at line 365 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.25 reserved

`ehsm_uint8_t` reserved[3]

Definition at line 361 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.26 rsa\_e\_bytes\_size

`ehsm_uint16_t` rsa\_e\_bytes\_size

Definition at line 360 of file eHSM\_If\_Evita\_Types\_Ip.h.

#### 4.38.5.27 size\_info

`union { ... } size_info`

#### 4.38.5.28 valid\_util

`ehsm_uint32_t` valid\_util

Definition at line 361 of file eHSM\_If\_Evita\_Types\_Ip.h.

## 4.39 eHSM\_If\_Ext\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Ext_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
```

### Classes

- struct [ehsm\\_secure\\_boot\\_st](#)
- struct [ehsm\\_fast\\_cmac\\_st](#)

### Enumerations

- enum [ehsm\\_power\\_mode\\_e](#) { [EHSM\\_POWER\\_MODE\\_NORMAL](#) = 0x1, [EHSM\\_POWER\\_MODE\\_LOW\\_POWER](#) }

### Functions

- [ehsm\\_uint32\\_t ehsm\\_get\\_random\\_key](#) ([ehsm\\_fw\\_random\\_key\\_type\\_e](#) key\_type, [ehsm\\_fw\\_random\\_key\\_slot\\_e](#) key\_slot)
 

*This function is used to install random SOC keys.*
- [ehsm\\_uint32\\_t ehsm\\_encrypt\\_key](#) ([ehsm\\_fw\\_encrypt\\_key\\_type\\_e](#) key\_type, [ehsm\\_fw\\_encrypt\\_key\\_slot\\_e](#) key\_slot, const [ehsm\\_uint8\\_t](#) \*key, [ehsm\\_uint32\\_t](#) key\_len)
 

*This function is used to install SOC keys such as upgrade encrypt key.*
- [ehsm\\_uint32\\_t ehsm\\_get\\_challenge](#) ([ehsm\\_get\\_challenge\\_st](#) \*req)
 

*This function is used to get challenge for debug authentication.*
- [ehsm\\_uint32\\_t ehsm\\_debug\\_auth](#) ([ehsm\\_debug\\_auth\\_st](#) \*req)
 

*This function is used for debug authentication.*
- [ehsm\\_uint32\\_t ehsm\\_hsm\\_fw\\_upgrade\\_init](#) (const char \*upg\_info, [ehsm\\_uint32\\_t](#) upg\_info\_size)
 

*Secure upgrade service init. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_hsm\\_fw\\_upgrade\\_update](#) (const [ehsm\\_uint8\\_t](#) \*upg\_encrypted\_img, [ehsm\\_uint32\\_t](#) upg\_img\_size, [ehsm\\_uint8\\_t](#) \*storage\_img, [ehsm\\_uint32\\_t](#) \*storage\_img\_size)
 

*Secure upgrade service update. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_hsm\\_fw\\_upgrade\\_finish](#) (const [ehsm\\_uint8\\_t](#) \*upg\_encrypted\_img, [ehsm\\_uint32\\_t](#) upg\_img\_size, [ehsm\\_uint8\\_t](#) \*storage\_img, [ehsm\\_uint32\\_t](#) \*storage\_img\_size, char \*mac, [ehsm\\_uint32\\_t](#) \*mac\_size)
 

*Secure upgrade service finish. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_hsm\\_fw\\_upgrade\\_verify\\_init](#) (const [ehsm\\_uint8\\_t](#) \*code\_info, [ehsm\\_uint32\\_t](#) code\_info\_size)
 

*Secure upgrade verify service init. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_hsm\\_fw\\_upgrade\\_verify\\_update](#) (const [ehsm\\_uint8\\_t](#) \*encrypted\_img, [ehsm\\_uint32\\_t](#) img\_size)
 

*Secure upgrade verify service update. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_hsm\\_fw\\_upgrade\\_verify\\_finish](#) (const [ehsm\\_uint8\\_t](#) \*encrypted\_img, [ehsm\\_uint32\\_t](#) img\_size)
 

*Secure upgrade verify service finish. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_secure\\_boot](#) ([ehsm\\_soc\\_image\\_verify\\_input\\_st](#) \*req)
 

*Secure boot verification. Only supported by eHSM firmware.*
- [ehsm\\_uint32\\_t ehsm\\_low\\_power](#) ([ehsm\\_power\\_mode\\_e](#) power\_mode)
 

*Request to let eHSM enter low power mode.*
- [ehsm\\_uint32\\_t ehsm\\_set\\_uart\\_baudrate](#) ([ehsm\\_uint32\\_t](#) baudrate)
 

*Set uart baudrate.*

- `ehsm_uint32_t ehsm_write_otp_data (ehsm_uint32_t otp_addr, ehsm_uint8_t *buf, ehsm_uint32_t write_size)`  
*Write data to otp area.*
- `ehsm_uint32_t ehsm_read_otp_data (ehsm_uint32_t otp_addr, ehsm_uint8_t *buf, ehsm_uint32_t read_size)`  
*Read data from otp area.*
- `ehsm_uint32_t ehsm_change_lifecycle (ehsm_lifecycle_e aim)`  
*Request to change lifecycle.*
- `ehsm_uint32_t ehsm_change_controlfield (ehsm_control_field_type_e type, ehsm_uint8_t *value, ehsm_uint16_t size)`  
*Request to change control field.*
- `ehsm_uint32_t ehsm_get_emu_status (ehsm_uint8_t *emu, ehsm_uint32_t *size)`  
*Request to get emu status.*
- `ehsm_uint32_t ehsm_sm9_generate_master_key (ehsm_sm9_master_key_type_e key_type)`  
*Generate KGC's master key pair.*
- `ehsm_uint32_t ehsm_sm9_generate_priv_key (ehsm_sm9_user_privkey_type_e key_type, ehsm_uint8_t *user_id, ehsm_uint32_t id_size, ehsm_key_mem_type_e type, ehsm_uint8_t hid, ehsm_uint32_t *key_handle)`  
*Generate user's private key.*
- `ehsm_uint32_t ehsm_sm9_exchg_key (ehsm_uint8_t role, ehsm_key_mem_type_e storage_type, ehsm_uint32_t shared_key_size, ehsm_uint32_t user_tmp_key_handle, ehsm_uint32_t user_priv_key_handle, const ehsm_uint8_t *peer_tmp_pub, const ehsm_uint8_t *peer_id, ehsm_uint32_t peer_id_size, const ehsm_uint8_t *self_id, ehsm_uint32_t self_id_size, const ehsm_uint8_t *fp12g, ehsm_uint8_t kgc_pub_key[64], ehsm_uint8_t *s1_s2, ehsm_uint32_t *s1_s2_size, ehsm_uint8_t *sa_sb, ehsm_uint32_t *sa_sb_size, ehsm_uint32_t *key_handle)`  
*SM9 Key exchange.*
- `ehsm_uint32_t ehsm_sm9_wrap_key (const ehsm_uint8_t *user_id, ehsm_uint32_t id_size, ehsm_uint8_t *key_addr, ehsm_uint32_t *key_size, const ehsm_uint8_t *fp12g, const ehsm_uint8_t pub_key[64], ehsm_uint8_t hid)`  
*SM9 key encapsulation.*
- `ehsm_uint32_t ehsm_sm9_unwrap_key (ehsm_uint32_t user_priv_key_handle, const ehsm_uint8_t *cipher_addr, ehsm_uint32_t cipher_size, const ehsm_uint8_t *user_id, ehsm_uint32_t id_size, ehsm_uint8_t *key_addr, ehsm_uint32_t *key_size)`  
*SM9 key decapsulation.*
- `ehsm_uint32_t ehsm_sm9_exckey_gen_tmpkey (const ehsm_uint8_t *peer_id, ehsm_uint32_t peer_id_size, ehsm_key_mem_type_e type, ehsm_uint8_t hid, const ehsm_uint8_t kgc_pub_key[64], ehsm_uint32_t *key_handle)`  
*generate user exchange temporary key pair.*
- `ehsm_uint32_t ehsm_sm9_export_key (ehsm_uint32_t key_handle, ehsm_uint8_t *key_blob, ehsm_uint32_t *key_blob_size, ehsm_uint8_t *key_auth_value, ehsm_uint32_t *key_auth_size)`  
*SM9 key export.*
- `ehsm_uint32_t ehsm_sm9_import_key (ehsm_uint32_t *key_handle, const ehsm_uint8_t *key_blob, ehsm_uint32_t key_blob_size, ehsm_key_mem_type_e type, ehsm_uint8_t *key_auth_value, ehsm_uint32_t key_auth_size, ehsm_uint8_t key_is_plain)`  
*SM9 key import.*
- `ehsm_uint32_t ehsm_sm9_gen_mastpubkey_from_mastprivkey (ehsm_sm9_master_key_type_e key_type, ehsm_uint8_t pub_key[128], ehsm_uint32_t *pub_key_size)`  
*Generate KGC's master public key from master private key.*
- `ehsm_uint32_t ehsm_sm9_gen_tmppubkey_from_tmpprivkey (ehsm_uint32_t key_handle, const ehsm_uint8_t *user_id, ehsm_uint32_t id_size, ehsm_uint8_t hid, ehsm_uint8_t pub_key[64])`  
*Generate user's temporary public key from private key for SM9 key-exchange system.*
- `ehsm_int32_t ehsm_sm9_remove_key (ehsm_uint32_t key_handle)`  
*Remove a SM9 key.*
- `ehsm_uint32_t ehsm_sm9_cipher (ehsm_uint8_t enc_type, ehsm_uint8_t dir, ehsm_uint8_t padding, ehsm_uint32_t key2_size, ehsm_uint8_t hid, ehsm_uint32_t key_handle, const ehsm_uint8_t *id_addr, ehsm_uint32_t id_size, const ehsm_uint8_t *fp12g, ehsm_uint8_t const *kgc_pubkey, const ehsm_uint8_t *input, ehsm_uint32_t input_size, ehsm_uint8_t *output, ehsm_uint32_t *output_size)`  
*SM9 encryption and decryption.*
- `ehsm_uint32_t ehsm_sm9_sign (ehsm_uint8_t dir, ehsm_uint8_t hid, ehsm_uint32_t key_handle, const ehsm_uint8_t *id_addr, ehsm_uint32_t id_size, const ehsm_uint8_t *fp12g, const ehsm_uint8_t *kgc_pubkey, const ehsm_uint8_t *input, ehsm_uint32_t input_size, ehsm_uint8_t *signature)`

*SM9 signification and verification.*

- `ehsm_uint32_t ehsm_close_debug (ehsm_challenge_type_e type)`

*This function is used to close authentication result(the previous authentication results will be cleared).*

## 4.39.1 Enumeration Type Documentation

### 4.39.1.1 ehsm\_power\_mode\_e

```
enum ehsm_power_mode_e
```

Enumerator

|                           |  |
|---------------------------|--|
| EHSM_POWER_MODE_NORMAL    |  |
| EHSM_POWER_MODE_LOW_POWER |  |

Definition at line 64 of file eHSM\_If\_Ext\_Ip.h.

## 4.39.2 Function Documentation

### 4.39.2.1 ehsm\_change\_controlfield()

```
ehsm_uint32_t ehsm_change_controlfield (
 ehsm_control_field_type_e type,
 ehsm_uint8_t * value,
 ehsm_uint16_t size)
```

Request to change control field.

Parameters

|    |              |                               |
|----|--------------|-------------------------------|
| in | <i>type</i>  | The control field type.       |
| in | <i>value</i> | The control field value.      |
| in | <i>size</i>  | The control field value size. |

Returns

`ehsm_uint32_t`

Return values

|                                         |                                                                     |
|-----------------------------------------|---------------------------------------------------------------------|
| <code>EHSM_ERR_SW_SUCCESS</code>        | No error.                                                           |
| <code>EHSM_ERR_PARAM_ERROR</code>       | Parameter error.                                                    |
| <code>EHSM_ERR_OUT_OF_MEM</code>        | No memory to handle this request.                                   |
| <code>EHSM_ERR_DEBUG_AUTH_FAILED</code> | No authentication is generated before, or authentication is failed. |

Definition at line 658 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.39.2.2 ehsm\_change\_lifecycle()

```
ehsm_uint32_t ehsm_change_lifecycle (
 ehsm_lifecycle_e aim)
```

Request to change lifecycle.

##### Parameters

|    |            |                                 |
|----|------------|---------------------------------|
| in | <i>aim</i> | The lifecycle to be changed to. |
|----|------------|---------------------------------|

##### Returns

ehsm\_uint32\_t

##### Return values

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error.                                                           |
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                                                    |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.                                   |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | No authentication is generated before, or authentication is failed. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Common error.                                                       |

##### Note

Partial lifecycle switching (user -> debug or user -> destroy) requires authorization with user auth key. So the relate authorization command please invoke firstly.

Definition at line 638 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.39.2.3 ehsm\_close\_debug()

```
ehsm_uint32_t ehsm_close_debug (
 ehsm_challenge_type_e type)
```

This function is used to close authentication result(the previous authentication results will be cleared).

##### Parameters

|    |             |                               |
|----|-------------|-------------------------------|
| in | <i>type</i> | which debug type to be closed |
|----|-------------|-------------------------------|

##### Returns

ehsm\_uint32\_t

## Return values

|                               |                                             |
|-------------------------------|---------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>    | The function has been successfully executed |
| <i>EHSM_ERR_GENERAL_ERROR</i> | Common error.                               |

## Note

Definition at line 710 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.4 ehsm\_debug\_auth()

```
ehsm_uint32_t ehsm_debug_auth (
 ehsm_debug_auth_st * req)
```

This function is used for debug authentication.

## Parameters

|    |     |                                |
|----|-----|--------------------------------|
| in | req | Info for debug authentication. |
|----|-----|--------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                                  |
|--------------------------------------|--------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>           | No error.                                        |
| <i>EHSM_ERR_PARAM_ERROR</i>          | Parameter error.                                 |
| <i>EHSM_ERR_OUT_OF_MEM</i>           | No memory to handle this request.                |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>    | Wrong data length                                |
| <i>EHSM_ERR_WRONG_CHALLENGE_TYPE</i> | Wrong challenge type for getting challenge.      |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>      | Given algorithm or algorithm mode not available. |
| <i>EHSM_ERR_EHSM_LIFECYCLE_LIMIT</i> | function is limited by lifecycle                 |
| <i>EHSM_ERR_WRONG_KEY_TYPE</i>       | Wrong key type.                                  |
| <i>EHSM_ERR_GENERAL_ERROR</i>        | Common error.                                    |
| <i>EHSM_ERR_HASH_WORK_ERROR</i>      | HASH IP work err by crypto software              |
| <i>EHSM_ERR_PKE_WORK_ERROR</i>       | PKE IP work err by crypto software               |

## Note

This function should be called after ehsm\_get\_challenge

Definition at line 216 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.



4.39.2.5 ehsm\_encrypt\_key()

```
ehsm_uint32_t ehsm_encrypt_key (
 ehsm_fw_encrypt_key_type_e key_type,
 ehsm_fw_encrypt_key_slot_e key_slot,
 const ehsm_uint8_t * key,
 ehsm_uint32_t key_len)
```

This functon is used to install SOC keys such as upgrade encrypt key.

Parameters

|    |          |                                                    |
|----|----------|----------------------------------------------------|
| in | key_type | Please see ehsm_fw_encrypt_key_type_e.             |
| in | key_slot | Please see ehsm_fw_encrypt_key_slot_e.             |
| in | key      | The raw key or hash of public key to be encrypted. |
| in | key_len  | The size of key in bytes.                          |

Returns

ehsm\_uint32\_t

Return values

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>  | No memery to handle this request. |

Note

This function can only be called in TEST\_MODE or MANUFACTOR\_MODE.

Definition at line 156 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

4.39.2.6 ehsm\_get\_challenge()

```
ehsm_uint32_t ehsm_get_challenge (
 ehsm_get_challenge_st * req)
```

This functon is used to get challenge for debug authentication.

Parameters

|    |     |                                                     |
|----|-----|-----------------------------------------------------|
| in | req | The challenge type and buf to stored the challenge. |
|----|-----|-----------------------------------------------------|

Returns

ehsm\_uint32\_t

## Return values

|                                      |                                             |
|--------------------------------------|---------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>           | No error.                                   |
| <i>EHSM_ERR_PARAM_ERROR</i>          | Parameter error.                            |
| <i>EHSM_ERR_OUT_OF_MEM</i>           | No memory to handle this request.           |
| <i>EHSM_ERR_EHSM_LIFECYCLE_LIMIT</i> | function is limited by lifecycle            |
| <i>EHSM_ERR_WRONG_CHALLENGE_TYPE</i> | Wrong challenge type for getting challenge. |
| <i>EHSM_ERR_TRNG_WORK_ERROR</i>      | TRNG IP work err by crypto software         |

Definition at line 200 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.7 ehsm\_get\_emu\_status()

```
ehsm_uint32_t ehsm_get_emu_status (
 ehsm_uint8_t * emu,
 ehsm_uint32_t * size)
```

Request to get emu status.

## Parameters

|    |            |                                                                                          |
|----|------------|------------------------------------------------------------------------------------------|
| in | <i>emu</i> | Host address to stored the EMU status.                                                   |
|    | [in/out]   | size: Input with the buffer size in bytes. It will be updated with the output data size. |

## Returns

ehsm\_uint32\_t

## Return values

|                                  |                                   |
|----------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>       | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>      | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>       | No memory to handle this request. |
| <i>EHSM_ERR_NOT_SUPPORT</i>      | The info type is not supported.   |
| <i>EHSM_ERR_WRONG_KEY_HANDLE</i> | Given key handle is wrong         |

Definition at line 685 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.8 ehsm\_get\_random\_key()

```
ehsm_uint32_t ehsm_get_random_key (
 ehsm_fw_random_key_type_e key_type,
 ehsm_fw_random_key_slot_e key_slot)
```

This function is used to install random SOC keys.

## Parameters

|    |                 |                                       |
|----|-----------------|---------------------------------------|
| in | <i>key_type</i> | Please see ehsm_fw_random_key_type_e. |
| in | <i>key_slot</i> | Please see ehsm_fw_random_key_slot_e. |

## Returns

ehsm\_uint32\_t

## Return values

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>  | No memory to handle this request. |

## Note

This function can only be called in TEST\_MODE or MANUFACTOR\_MODE.

Definition at line 129 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.9 ehsm\_hsm\_fw\_upgrade\_finish()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_finish (
 const ehsm_uint8_t * upg_encrypted_img,
 ehsm_uint32_t upg_img_size,
 ehsm_uint8_t * storage_img,
 ehsm_uint32_t * storage_img_size,
 char * mac,
 ehsm_uint32_t * mac_size)
```

Secure upgrade service finish. Only supported by eHSM firmware.

## Parameters

|    |                          |                                                                |
|----|--------------------------|----------------------------------------------------------------|
| in | <i>upg_encrypted_img</i> | Remaining encrypted code image, may be NULL.                   |
| in | <i>upg_img_size</i>      | Remaining code image size in bytes, may be 0.                  |
|    | <i>[in/out]</i>          | storage_img: Output storage encrypted code image, may be NULL. |
|    | <i>[in/out]</i>          | storage_img_size: Output code image size, may be 0.            |

## Returns

ehsm\_uint32\_t

## Return values

|                             |                  |
|-----------------------------|------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.        |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error. |

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memery to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 288 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.10 ehsm\_hsm\_fw\_upgrade\_init()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_init (
 const char * upg_info,
 ehsm_uint32_t upg_info_size)
```

Secure upgrade service init. Only supported by eHSM firmware.

## Parameters

|    |                      |                                      |
|----|----------------------|--------------------------------------|
| in | <i>upg_info</i>      | Upgrade info header.                 |
| in | <i>upg_info_size</i> | Size of upgrade info header in byte. |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memery to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 231 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.11 ehsm\_hsm\_fw\_upgrade\_update()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_update (
 const ehsm_uint8_t * upg_encrypted_img,
```

```
ehsm_uint32_t upg_img_size,
ehsm_uint8_t * storage_img,
ehsm_uint32_t * storage_img_size)
```

Secure upgrade service update. Only supported by eHSM firmware.

#### Parameters

|    |                          |                                                                                          |
|----|--------------------------|------------------------------------------------------------------------------------------|
| in | <i>upg_encrypted_img</i> | Part of the encrypted code image.                                                        |
| in | <i>upg_img_size</i>      | Input size in bytes, should be a multiple of the hash or symmetric algorithm block size. |
|    | [in/out]                 | storage_img: Output storage encrypted code image.                                        |
|    | [in/out]                 | storage_img_size: Output code image size.                                                |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 256 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.39.2.12 ehsm\_hsm\_fw\_upgrade\_verify\_finish()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_finish (
 const ehsm_uint8_t * encrypted_img,
 ehsm_uint32_t img_size)
```

Secure upgrade verify service finish. Only supported by eHSM firmware.

#### Parameters

|    |                      |                                               |
|----|----------------------|-----------------------------------------------|
| in | <i>encrypted_img</i> | Remaining encrypted code image, may be NULL.  |
| in | <i>img_size</i>      | Remaining code image size in bytes, may be 0. |

#### Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 371 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.13 ehsm\_hsm\_fw\_upgrade\_verify\_init()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_init (
 const ehsm_uint8_t * code_info,
 ehsm_uint32_t code_info_size)
```

Secure upgrade verify service init. Only supported by eHSM firmware.

## Parameters

|    |                       |                                   |
|----|-----------------------|-----------------------------------|
| in | <i>code_info</i>      | Code info header.                 |
| in | <i>code_info_size</i> | Size of code info header in byte. |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 321 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.14 ehsm\_hsm\_fw\_upgrade\_verify\_update()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_update (
 const ehsm_uint8_t * encrypted_img,
 ehsm_uint32_t img_size)
```

Secure upgrade verify service update. Only supported by eHSM firmware.

#### Parameters

|    |                      |                                                                                                                                         |
|----|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>encrypted_img</i> | Part of the encrypted code image.                                                                                                       |
| in | <i>img_size</i>      | Input size in bytes, should be a multiple of both storage encrypt algorithm (aes or sm4) and secure boot algorithm (symmetric or hash). |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 346 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.39.2.15 ehsm\_low\_power()

```
ehsm_uint32_t ehsm_low_power (
 ehsm_power_mode_e power_mode)
```

Request to let eHSM enter low power mode.

#### Parameters

|    |                   |            |
|----|-------------------|------------|
| in | <i>power_mode</i> | power mode |
|----|-------------------|------------|

#### Returns

ehsm\_uint32\_t

#### Return values

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error.                                                           |
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                                                    |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.                                   |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | No authentication is generated before, or authentication is failed. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Common error.                                                       |

Definition at line 609 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.39.2.16 ehsm\_read\_otp\_data()

```
ehsm_uint32_t ehsm_read_otp_data (
 ehsm_uint32_t otp_addr,
 ehsm_uint8_t * buf,
 ehsm_uint32_t read_size)
```

Read data from otp area.

##### Parameters

|    |                  |                                                       |
|----|------------------|-------------------------------------------------------|
| in | <i>otp_addr</i>  | The address of OTP, where data being read from.       |
| in | <i>buf</i>       | The start address of OTP data buffer.                 |
| in | <i>read_size</i> | The size of buffer, The maximum length is 1024 bytes; |

##### Returns

ehsm\_uint32\_t

##### Return values

|                               |                                   |
|-------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>    | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>   | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>    | No memory to handle this request. |
| <i>EHSM_ERR_GENERAL_ERROR</i> | Common error.                     |

Definition at line 533 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.39.2.17 ehsm\_secure\_boot()

```
ehsm_uint32_t ehsm_secure_boot (
 ehsm_soc_image_verify_input_st * req)
```

Secure boot verification. Only supported by eHSM firmware.

##### Parameters

|    |            |                                                  |
|----|------------|--------------------------------------------------|
| in | <i>req</i> | The image info (header and code) to be verified. |
|----|------------|--------------------------------------------------|

##### Returns

ehsm\_uint32\_t



## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 506 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.18 ehsm\_set\_uart\_baudrate()

```
ehsm_uint32_t ehsm_set_uart_baudrate (
 ehsm_uint32_t baudrate)
```

Set uart baudrate.

## Parameters

|    |                 |                                                     |
|----|-----------------|-----------------------------------------------------|
| in | <i>baudrate</i> | baudrate to be set, refers to ehsm_uart_baudrate_e. |
|----|-----------------|-----------------------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>  | No memory to handle this request. |

Definition at line 623 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.39.2.19 ehsm\_sm9\_cipher()

```
ehsm_uint32_t ehsm_sm9_cipher (
 ehsm_uint8_t enc_type,
 ehsm_uint8_t dir,
 ehsm_uint8_t padding,
 ehsm_uint32_t key2_size,
 ehsm_uint8_t hid,
 ehsm_uint32_t key_handle,
 const ehsm_uint8_t * id_addr,
```

```

ehsm_uint32_t id_size,
const ehsm_uint8_t * fp12g,
ehsm_uint8_t const * kgc_pubkey,
const ehsm_uint8_t * input,
ehsm_uint32_t input_size,
ehsm_uint8_t * output,
ehsm_uint32_t * output_size)

```

SM9 encryption and decryption.

#### Parameters

|     |                    |                                                                                               |
|-----|--------------------|-----------------------------------------------------------------------------------------------|
| in  | <i>enc_type</i>    | type of encryption (SM9_ENC_KDF_STREAM_CIPHER or SM9_ENC_KDF_BLOCK_CIPHER)                    |
| in  | <i>dir</i>         | encryption or decryption                                                                      |
| in  | <i>padding</i>     | type of padding(SKE_NO_PADDING or SKE_PKCS_5_7_PADDING)                                       |
| in  | <i>input</i>       | input address of data                                                                         |
| in  | <i>input_size</i>  | the size of input data                                                                        |
| out | <i>output</i>      | cipher text of plain text                                                                     |
| out | <i>output_size</i> | the size of cipher, only valid when in encryption mode                                        |
| in  | <i>id_addr</i>     | the address of identity of user B, user B is the cipher receiver                              |
| in  | <i>id_size</i>     | the size of identity of user B                                                                |
| in  | <i>fp12g</i>       | the value of e(Ppub_e, P2), if set to null, it will be calculated within the function         |
| in  | <i>kgc_pubkey</i>  | KGC's system encryption master public key                                                     |
| in  | <i>key2_size</i>   | bytes length of the key K2 in MAC function                                                    |
| in  | <i>hid</i>         | user private generation function identity, published by KGC, default value is 0x03, one byte. |
| in  | <i>key_handle</i>  | key handle in sm9                                                                             |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                   |                                    |
|-----------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i>    | PKE IP work err by crypto software |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i> | Wrong data length                  |

#### 4.39.2.20 ehsm\_sm9\_exchg\_key()

```

ehsm_uint32_t ehsm_sm9_exchg_key (
 ehsm_uint8_t role,
 ehsm_key_mem_type_e storage_type,
 ehsm_uint32_t shared_key_size,
 ehsm_uint32_t user_tmp_key_handle,
 ehsm_uint32_t user_priv_key_handle,
 const ehsm_uint8_t * peer_tmp_pub,
 const ehsm_uint8_t * peer_id,

```

```

ehsm_uint32_t peer_id_size,
const ehsm_uint8_t * self_id,
ehsm_uint32_t self_id_size,
const ehsm_uint8_t * fp12g,
ehsm_uint8_t kgc_pub_key[64],
ehsm_uint8_t * s1_s2,
ehsm_uint32_t * s1_s2_size,
ehsm_uint8_t * sa_sb,
ehsm_uint32_t * sa_sb_size,
ehsm_uint32_t * key_handle)

```

SM9 Key exchange.

#### Parameters

|     |                             |                                                                             |
|-----|-----------------------------|-----------------------------------------------------------------------------|
| in  | <i>role</i>                 | Local user's role(SM9_Role_Sponsor or SM9_Role_Responder)                   |
| in  | <i>key_size</i>             | The output key bytes                                                        |
| in  | <i>user_tmp_key_handle</i>  | Key handle of local user's temporary key.                                   |
| in  | <i>user_priv_key_handle</i> | Key handle of local user's private key.                                     |
| in  | <i>peer_tmp_pub</i>         | Peer user's temporary public key.                                           |
| in  | <i>peer_id</i>              | Peer user's ID.                                                             |
| in  | <i>ida_size</i>             | Size of peer user's ID.                                                     |
| in  | <i>self_id</i>              | Self user ID.                                                               |
| in  | <i>idb_size</i>             | Size of self user ID.                                                       |
| in  | <i>s1_s2</i>                | Sponsor's S1, or responder's S2                                             |
| in  | <i>s1_s2_size</i>           | Size of s1_s2                                                               |
| in  | <i>sa_sb</i>                | Sponsor's SA, or responder's SB                                             |
| in  | <i>sa_sb_size</i>           | Size of sa_sb                                                               |
| in  | <i>fp12g</i>                | The value of e(P1, pub_key), if set to NULL, it will be calculated in eHSM. |
| in  | <i>pub_key</i>              | KGC's encryption master public key, pub_key size is 64 byte.                |
| out | <i>key_handle</i>           | Returned key handle                                                         |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

#### 4.39.2.21 ehsm\_sm9\_exckey\_gen\_tmpkey()

```

ehsm_uint32_t ehsm_sm9_exckey_gen_tmpkey (
 const ehsm_uint8_t * peer_id,
 ehsm_uint32_t peer_id_size,
 ehsm_key_mem_type_e type,

```

```
ehsm_uint8_t hid,
const ehsm_uint8_t kgc_pub_key[64],
ehsm_uint32_t * key_handle)
```

generate user exchange temporary key pair.

#### Parameters

|     |                   |                                                                                                                                                                                                                              |
|-----|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>user_id</i>    | User ID                                                                                                                                                                                                                      |
| in  | <i>id_size</i>    | Size of user ID                                                                                                                                                                                                              |
| in  | <i>type</i>       | Memory_target non-volatile or RAM                                                                                                                                                                                            |
| in  | <i>hid</i>        | User private key generation function identity, published by KGC, default value is 0x01 for sign user private key, 0x02 for exchange user private key, 0x03 for encryption user private key, 0x02 for exchange user temp key. |
| in  | <i>pub_key</i>    | KGC's encryption master public key, pub_key size is 64 byte.                                                                                                                                                                 |
| out | <i>key_handle</i> | Returned key handle of user private key.                                                                                                                                                                                     |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

#### 4.39.2.22 ehsm\_sm9\_export\_key()

```
ehsm_uint32_t ehsm_sm9_export_key (
 ehsm_uint32_t key_handle,
 ehsm_uint8_t * key_blob,
 ehsm_uint32_t * key_blob_size,
 ehsm_uint8_t * key_auth_value,
 ehsm_uint32_t * key_auth_size)
```

SM9 key export.

#### Parameters

|    |                       |                                                                                     |
|----|-----------------------|-------------------------------------------------------------------------------------|
| in | <i>key_handle</i>     | Key handle of the key to be exported                                                |
| in | <i>key_blob</i>       | Address of the exported key blob                                                    |
|    | <i>[in/out]</i>       | key_blob_size: In for Size of buffer size, out for the real key blob size           |
| in | <i>key_auth_value</i> | Address of the authorization code for the exported key.                             |
|    | <i>[in/out]</i>       | key_auth_size: In for Size of buffer size, out for the real authorization code size |

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.23 ehsm\_sm9\_gen\_mastpubkey\_from\_mastprivkey()

```
ehsm_uint32_t ehsm_sm9_gen_mastpubkey_from_mastprivkey (
 ehsm_sm9_master_key_type_e key_type,
 ehsm_uint8_t pub_key[128],
 ehsm_uint32_t * pub_key_size)
```

Generate KGC's master public key from master private key.

## Parameters

|     |                     |                                |
|-----|---------------------|--------------------------------|
| in  | <i>key_type</i>     | Masker key type                |
| out | <i>pub_key</i>      | Buffer for returned public key |
| out | <i>pub_key_size</i> | Size of returned public key    |

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.24 ehsm\_sm9\_gen\_tmppubkey\_from\_tmpprivkey()

```
ehsm_uint32_t ehsm_sm9_gen_tmppubkey_from_tmpprivkey (
 ehsm_uint32_t key_handle,
 const ehsm_uint8_t * user_id,
 ehsm_uint32_t id_size,
 ehsm_uint8_t hid,
 ehsm_uint8_t pub_key[64])
```

Generate user's temporary public key from private key for SM9 key-exchange system.

## Parameters

|    |                   |                                                                                                   |
|----|-------------------|---------------------------------------------------------------------------------------------------|
| in | <i>key_handle</i> | Key handle of local's temporary private key                                                       |
| in | <i>user_id</i>    | User ID                                                                                           |
| in | <i>id_size</i>    | Size of User ID                                                                                   |
| in | <i>hid</i>        | user private key generation function identity, published by KGC, default value is 0x02, one byte. |
| in | <i>pub_key</i>    | Buffer for returned public key, pub_key size is 64 byte, so buffer must be larger than 64 bytes.  |

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.25 ehsm\_sm9\_generate\_master\_key()

```
ehsm_uint32_t ehsm_sm9_generate_master_key (
 ehsm_sm9_master_key_type_e key_type)
```

Generate KGC's master key pair.

## Parameters

|    |                 |                                                                                       |
|----|-----------------|---------------------------------------------------------------------------------------|
| in | <i>key_type</i> | Master key type of SM9, sign master key, encryption master key or exchange master key |
|----|-----------------|---------------------------------------------------------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.26 ehsm\_sm9\_generate\_priv\_key()

```
ehsm_uint32_t ehsm_sm9_generate_priv_key (
```

```

ehsm_sm9_user_privkey_type_e key_type,
ehsm_uint8_t * user_id,
ehsm_uint32_t id_size,
ehsm_key_mem_type_e type,
ehsm_uint8_t hid,
ehsm_uint32_t * key_handle)

```

Generate user's private key.

#### Parameters

|     |                   |                                                                                                                                                                                                                              |
|-----|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in  | <i>key_type</i>   | User private key type, sign user private key, encryption user private key, exchange user private key or exchange user temp key                                                                                               |
| in  | <i>user_id</i>    | User ID                                                                                                                                                                                                                      |
| in  | <i>id_size</i>    | Size of user ID                                                                                                                                                                                                              |
| in  | <i>type</i>       | Memory_target non-volatile or RAM                                                                                                                                                                                            |
| in  | <i>hid</i>        | User private key generation function identity, published by KGC, default value is 0x01 for sign user private key, 0x02 for exchange user private key, 0x03 for encryption user private key, 0x02 for exchange user temp key. |
| out | <i>key_handle</i> | Returned key handle of user private key.                                                                                                                                                                                     |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

#### 4.39.2.27 ehsm\_sm9\_import\_key()

```

ehsm_uint32_t ehsm_sm9_import_key (
 ehsm_uint32_t * key_handle,
 const ehsm_uint8_t * key_blob,
 ehsm_uint32_t key_blob_size,
 ehsm_key_mem_type_e type,
 ehsm_uint8_t * key_auth_value,
 ehsm_uint32_t key_auth_size,
 ehsm_uint8_t key_is_plain)

```

SM9 key import.

#### Parameters

|    |                      |                                        |
|----|----------------------|----------------------------------------|
| in | <i>key_blob</i>      | Address of the key blob to be imported |
| in | <i>key_blob_size</i> | Size of key_blob                       |
| in | <i>key_handle</i>    | Returned key handle                    |



## Parameters

|    |                       |                                   |
|----|-----------------------|-----------------------------------|
| in | <i>type</i>           | type of imported key              |
| in | <i>key_auth_value</i> | Address of the authorization code |
| in | <i>key_auth_size</i>  | Aize of the authorization code    |
| in | <i>key_is_plain</i>   | Import key is plain               |

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memery to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.28 ehsm\_sm9\_remove\_key()

```
ehsm_int32_t ehsm_sm9_remove_key (
 ehsm_uint32_t key_handle)
```

Remove a SM9 key.

## Parameters

|    |                   |                           |
|----|-------------------|---------------------------|
| in | <i>key_handle</i> | Key handle to be removed. |
|----|-------------------|---------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                               |                                   |
|-------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>    | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>   | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>    | No memery to handle this request. |
| <i>EHSM_ERR_GENERAL_ERROR</i> | Common error.                     |

## 4.39.2.29 ehsm\_sm9\_sign()

```
ehsm_uint32_t ehsm_sm9_sign (
 ehsm_uint8_t dir,
```

```

ehsm_uint8_t hid,
ehsm_uint32_t key_handle,
const ehsm_uint8_t * id_addr,
ehsm_uint32_t id_size,
const ehsm_uint8_t * fp12g,
const ehsm_uint8_t * kgc_pubkey,
const ehsm_uint8_t * input,
ehsm_uint32_t input_size,
ehsm_uint8_t * signature)

```

SM9 signification and verification.

#### Parameters

|         |                     |                                                                                               |
|---------|---------------------|-----------------------------------------------------------------------------------------------|
| in      | <i>dir</i>          | encryption or decryption                                                                      |
| in      | <i>key_handle</i>   | key id                                                                                        |
| in      | <i>input</i>        | input address of msg                                                                          |
| in      | <i>input_size</i>   | the size of input msg                                                                         |
| in, out | <i>signature</i>    | address of signature to be generated or verified                                              |
| in      | <i>id_addr</i>      | the address of identity of user B, user B is the cipher receiver                              |
| in      | <i>id_size</i>      | the size of identity of user B                                                                |
| in      | <i>fp12g</i>        | the value of e(Ppub_e, P2), if set to null, it will be calculated within the function         |
| in      | <i>kgc_pubkey</i>   | KGC's system encryption master public key                                                     |
| in      | <i>hid</i>          | user private generation function identity, published by KGC, default value is 0x03, one byte. |
| in      | <i>auth_code</i>    | authrization code                                                                             |
| in      | <i>auth_code_sz</i> | size of authrization code                                                                     |

#### Returns

ehsm\_uint32\_t

#### Return values

|                                   |                                    |
|-----------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memery to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i>    | PKE IP work err by crypto software |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i> | Wrong data length                  |

#### 4.39.2.30 ehsm\_sm9\_unwrap\_key()

```

ehsm_uint32_t ehsm_sm9_unwrap_key (
 ehsm_uint32_t user_priv_key_handle,
 const ehsm_uint8_t * cipher_addr,
 ehsm_uint32_t cipher_size,
 const ehsm_uint8_t * user_id,
 ehsm_uint32_t id_size,
 ehsm_uint8_t * key_addr,
 ehsm_uint32_t * key_size)

```

SM9 key decapsulation.

## Parameters

|    |                             |                                |
|----|-----------------------------|--------------------------------|
| in | <i>user_priv_key_handle</i> | Key handle of user private key |
| in | <i>cipher_addr</i>          | Address of the wrapped key.    |
| in | <i>cipher_size</i>          | Size of the wrapped key        |
| in | <i>user_id</i>              | User ID                        |
| in | <i>id_size</i>              | Size of User ID                |
| in | <i>key_addr</i>             | Address of the unwrapped key.  |
| in | <i>key_size</i>             | Size of the unwrapped key.     |

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.31 ehsm\_sm9\_wrap\_key()

```
ehsm_uint32_t ehsm_sm9_wrap_key (
 const ehsm_uint8_t * user_id,
 ehsm_uint32_t id_size,
 ehsm_uint8_t * key_addr,
 ehsm_uint32_t * key_size,
 const ehsm_uint8_t * fp12g,
 const ehsm_uint8_t pub_key[64],
 ehsm_uint8_t hid)
```

SM9 key encapsulation.

## Parameters

|    |                 |                                                                                                                               |
|----|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| in | <i>user_id</i>  | User ID                                                                                                                       |
| in | <i>id_size</i>  | Size of user ID                                                                                                               |
| in | <i>key_addr</i> | Address of the generated key and its cipher. The buffer should be key_size + 64                                               |
|    | [in/out]        | key_size: Size of generated key.                                                                                              |
| in | <i>fp12g</i>    | The value of e(P1, pub_key), if set to NULL, it will be calculated in eHSM.                                                   |
| in | <i>pub_key</i>  | KGC's encryption master public key, pub_key size is 64 bytes.                                                                 |
| in | <i>hid</i>      | User private key generation function identity, published by KGC, default value is 0x03, one byte, ingnroed in signature mode. |

## Returns

ehsm\_uint32\_t

## Return values

|                                |                                    |
|--------------------------------|------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>     | No error.                          |
| <i>EHSM_ERR_PARAM_ERROR</i>    | Parameter error.                   |
| <i>EHSM_ERR_OUT_OF_MEM</i>     | No memory to handle this request.  |
| <i>EHSM_ERR_GENERAL_ERROR</i>  | Common error.                      |
| <i>EHSM_ERR_PKE_WORK_ERROR</i> | PKE IP work err by crypto software |

## 4.39.2.32 ehsm\_write\_otp\_data()

```
ehsm_uint32_t ehsm_write_otp_data (
 ehsm_uint32_t otp_addr,
 ehsm_uint8_t * buf,
 ehsm_uint32_t write_size)
```

Write data to otp area.

## Parameters

|    |                   |                                                       |
|----|-------------------|-------------------------------------------------------|
| in | <i>otp_addr</i>   | The address of OTP, where data being write from.      |
| in | <i>buf</i>        | The start address of OTP data buffer.                 |
| in | <i>write_size</i> | The size of buffer, The maximum length is 1024 bytes; |

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                   |
|--------------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>           | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>          | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>           | No memory to handle this request. |
| <i>EHSM_ERR_EHSM_LIFECYCLE_LIMIT</i> | function is limited by lifecycle  |
| <i>EHSM_ERR_GENERAL_ERROR</i>        | Common error.                     |

Definition at line 554 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.40 eHSM\_If\_Ext\_Sm9\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include <string.h>
#include "eHSM_If_Ext_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
```

```
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Mailbox_CmdId_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
```

## Macros

- `#define EHSM_CRYPTO_V_SM9_MAX_ID_SIZE (1024U)`

### 4.40.1 Macro Definition Documentation

#### 4.40.1.1 EHSM\_CRYPTO\_V\_SM9\_MAX\_ID\_SIZE

```
#define EHSM_CRYPTO_V_SM9_MAX_ID_SIZE (1024U)
```

Definition at line 26 of file eHSM\_If\_Ext\_Sm9\_Ip.c.

## 4.41 eHSM\_If\_Ext\_SysMgr\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include <string.h>
#include "eHSM_If_Ext_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Mailbox_CmdId_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_She_ErrCode_Ip.h"
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_Mailbox_Ip.h"
```

## Macros

- `#define EHSM_IMAGE_VERIFY_TYPE_FW_UPGRADE 0x01UL`
- `#define EHSM_IMAGE_VERIFY_TYPE_SOC_BOOT 0x02UL`
- `#define EHSM_IMAGE_VERIFY_TYPE_SOC_UPGRADE 0x04UL`

## Functions

- `ehsm_uint32_t ehsm_get_random_key` (`ehsm_fw_random_key_type_e` key\_type, `ehsm_fw_random_key_slot_e` key\_slot)
 

*This function is used to install random SOC keys.*
- `ehsm_uint32_t ehsm_encrypt_key` (`ehsm_fw_encrypt_key_type_e` key\_type, `ehsm_fw_encrypt_key_slot_e` key\_slot, `const ehsm_uint8_t *key`, `ehsm_uint32_t` key\_len)
 

*This function is used to install SOC keys such as upgrade encrypt key.*
- `ehsm_uint32_t ehsm_get_challenge` (`ehsm_get_challenge_st` \*req)
 

*This function is used to get challenge for debug authentication.*
- `ehsm_uint32_t ehsm_debug_auth` (`ehsm_debug_auth_st` \*req)
 

*This function is used for debug authentication.*
- `ehsm_uint32_t ehsm_hsm_fw_upgrade_init` (`const char *upg_info`, `ehsm_uint32_t` upg\_info\_size)
 

*Secure upgrade service init. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_hsm_fw_upgrade_update` (`const ehsm_uint8_t *upg_encrypted_img`, `ehsm_uint32_t` upg\_img\_size, `ehsm_uint8_t *storage_img`, `ehsm_uint32_t` \*storage\_img\_size)
 

*Secure upgrade service update. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_hsm_fw_upgrade_finish` (`const ehsm_uint8_t *upg_encrypted_img`, `ehsm_uint32_t` upg\_img\_size, `ehsm_uint8_t *storage_img`, `ehsm_uint32_t` \*storage\_img\_size, `char *mac`, `ehsm_uint32_t` \*mac\_size)
 

*Secure upgrade service finish. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_init` (`const ehsm_uint8_t *code_info`, `ehsm_uint32_t` code\_info\_size)
 

*Secure upgrade verify service init. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_update` (`const ehsm_uint8_t *encrypted_img`, `ehsm_uint32_t` img\_size)
 

*Secure upgrade verify service update. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_finish` (`const ehsm_uint8_t *encrypted_img`, `ehsm_uint32_t` img\_size)
 

*Secure upgrade verify service finish. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_secure_boot` (`ehsm_soc_image_verify_input_st` \*req)
 

*Secure boot verification. Only supported by eHSM firmware.*
- `ehsm_uint32_t ehsm_read_otp_data` (`ehsm_uint32_t` otp\_addr, `ehsm_uint8_t` \*buf, `ehsm_uint32_t` read\_size)
 

*Read data from otp area.*
- `ehsm_uint32_t ehsm_write_otp_data` (`ehsm_uint32_t` otp\_addr, `ehsm_uint8_t` \*buf, `ehsm_uint32_t` write\_size)
 

*Write data to otp area.*
- `ehsm_uint32_t ehsm_self_test` (`ehsm_uint32_t` flag)
 

*The self test service.*
- `ehsm_uint32_t ehsm_low_power` (`ehsm_power_mode_e` power\_mode)
 

*Request to let eHSM enter low power mode.*
- `ehsm_uint32_t ehsm_set_uart_baudrate` (`ehsm_uint32_t` baudrate)
 

*Set uart baudrate.*
- `ehsm_uint32_t ehsm_change_lifecycle` (`ehsm_lifecycle_e` aim)
 

*Request to change lifecycle.*
- `ehsm_uint32_t ehsm_change_controlfield` (`ehsm_control_field_type_e` type, `ehsm_uint8_t` \*value, `ehsm_uint16_t` size)
 

*Request to change control field.*
- `ehsm_uint32_t ehsm_get_emu_status` (`ehsm_uint8_t` \*emu, `ehsm_uint32_t` \*size)
 

*Request to get emu status.*
- `ehsm_uint32_t ehsm_close_debug` (`ehsm_challenge_type_e` type)
 

*This function is used to close authentication result(the previous authentication results will be cleared).*

## Variables

- `ehsm_uint32_t g_ctx` = 0x1FFEF000

### 4.41.1 Macro Definition Documentation

#### 4.41.1.1 EHSM\_IMAGE\_VERIFY\_TYPE\_FW\_UPGRADE

```
#define EHSM_IMAGE_VERIFY_TYPE_FW_UPGRADE 0x01UL
```

Definition at line 25 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.41.1.2 EHSM\_IMAGE\_VERIFY\_TYPE\_SOC\_BOOT

```
#define EHSM_IMAGE_VERIFY_TYPE_SOC_BOOT 0x02UL
```

Definition at line 26 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

#### 4.41.1.3 EHSM\_IMAGE\_VERIFY\_TYPE\_SOC\_UPGRADE

```
#define EHSM_IMAGE_VERIFY_TYPE_SOC_UPGRADE 0x04UL
```

Definition at line 27 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

### 4.41.2 Function Documentation

#### 4.41.2.1 ehsm\_change\_controlfield()

```
ehsm_uint32_t ehsm_change_controlfield (
 ehsm_control_field_type_e type,
 ehsm_uint8_t * value,
 ehsm_uint16_t size)
```

Request to change control field.

##### Parameters

|    |              |                               |
|----|--------------|-------------------------------|
| in | <i>type</i>  | The control field type.       |
| in | <i>value</i> | The control field value.      |
| in | <i>size</i>  | The control field value size. |

##### Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error.                                                           |
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                                                    |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.                                   |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | No authentication is generated before, or authentication is failed. |

Definition at line 658 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.2 ehsm\_change\_lifecycle()

```
ehsm_uint32_t ehsm_change_lifecycle (
 ehsm_lifecycle_e aim)
```

Request to change lifecycle.

## Parameters

|    |     |                                 |
|----|-----|---------------------------------|
| in | aim | The lifecycle to be changed to. |
|----|-----|---------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error.                                                           |
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                                                    |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.                                   |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | No authentication is generated before, or authentication is failed. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Common error.                                                       |

## Note

Partial lifecycle switching (user -> debug or user -> destroy) requires authorization with user auth key. So the relate authorization command please invoke firstly.

Definition at line 638 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.3 ehsm\_close\_debug()

```
ehsm_uint32_t ehsm_close_debug (
 ehsm_challenge_type_e type)
```

This function is used to close authentication result(the previous authentication results will be cleared).



## Parameters

|           |             |                               |
|-----------|-------------|-------------------------------|
| <i>in</i> | <i>type</i> | which debug type to be closed |
|-----------|-------------|-------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                               |                                             |
|-------------------------------|---------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>    | The function has been successfully executed |
| <i>EHSM_ERR_GENERAL_ERROR</i> | Common error.                               |

## Note

Definition at line 710 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.4 ehsm\_debug\_auth()

```
ehsm_uint32_t ehsm_debug_auth (
 ehsm_debug_auth_st * req)
```

This function is used for debug authentication.

## Parameters

|           |            |                                |
|-----------|------------|--------------------------------|
| <i>in</i> | <i>req</i> | Info for debug authentication. |
|-----------|------------|--------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                                  |
|--------------------------------------|--------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>           | No error.                                        |
| <i>EHSM_ERR_PARAM_ERROR</i>          | Parameter error.                                 |
| <i>EHSM_ERR_OUT_OF_MEM</i>           | No memory to handle this request.                |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>    | Wrong data length                                |
| <i>EHSM_ERR_WRONG_CHALLENGE_TYPE</i> | Wrong challenge type for getting challenge.      |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>      | Given algorithm or algorithm mode not available. |
| <i>EHSM_ERR_EHSM_LIFECYCLE_LIMIT</i> | function is limited by lifecycle                 |
| <i>EHSM_ERR_WRONG_KEY_TYPE</i>       | Wrong key type.                                  |
| <i>EHSM_ERR_GENERAL_ERROR</i>        | Common error.                                    |
| <i>EHSM_ERR_HASH_WORK_ERROR</i>      | HASH IP work err by crypto software              |
| <i>EHSM_ERR_PKE_WORK_ERROR</i>       | PKE IP work err by crypto software               |

Note

This function should be called after ehsm\_get\_challenge

Definition at line 216 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

4.41.2.5 ehsm\_encrypt\_key()

```
ehsm_uint32_t ehsm_encrypt_key (
 ehsm_fw_encrypt_key_type_e key_type,
 ehsm_fw_encrypt_key_slot_e key_slot,
 const ehsm_uint8_t * key,
 ehsm_uint32_t key_len)
```

This functon is used to install SOC keys such as upgrade encrypt key.

Parameters

|    |          |                                                    |
|----|----------|----------------------------------------------------|
| in | key_type | Please see ehsm_fw_encrypt_key_type_e.             |
| in | key_slot | Please see ehsm_fw_encrypt_key_slot_e.             |
| in | key      | The raw key or hash of public key to be encrypted. |
| in | key_len  | The size of key in bytes.                          |

Returns

ehsm\_uint32\_t

Return values

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>  | No memery to handle this request. |

Note

This function can only be called in TEST\_MODE or MANUFACTOR\_MODE.

Definition at line 156 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

4.41.2.6 ehsm\_get\_challenge()

```
ehsm_uint32_t ehsm_get_challenge (
 ehsm_get_challenge_st * req)
```

This functon is used to get challenge for debug authentication.

## Parameters

|    |     |                                                     |
|----|-----|-----------------------------------------------------|
| in | req | The challenge type and buf to stored the challenge. |
|----|-----|-----------------------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                             |
|--------------------------------------|---------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>           | No error.                                   |
| <i>EHSM_ERR_PARAM_ERROR</i>          | Parameter error.                            |
| <i>EHSM_ERR_OUT_OF_MEM</i>           | No memory to handle this request.           |
| <i>EHSM_ERR_EHSM_LIFECYCLE_LIMIT</i> | function is limited by lifecycle            |
| <i>EHSM_ERR_WRONG_CHALLENGE_TYPE</i> | Wrong challenge type for getting challenge. |
| <i>EHSM_ERR_TRNG_WORK_ERROR</i>      | TRNG IP work err by crypto software         |

Definition at line 200 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.7 ehsm\_get\_emu\_status()

```
ehsm_uint32_t ehsm_get_emu_status (
 ehsm_uint8_t * emu,
 ehsm_uint32_t * size)
```

Request to get emu status.

## Parameters

|    |          |                                                                                          |
|----|----------|------------------------------------------------------------------------------------------|
| in | emu      | Host address to stored the EMU status.                                                   |
|    | [in/out] | size: Input with the buffer size in bytes. It will be updated with the output data size. |

## Returns

ehsm\_uint32\_t

## Return values

|                                  |                                   |
|----------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>       | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>      | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>       | No memory to handle this request. |
| <i>EHSM_ERR_NOT_SUPPORT</i>      | The info type is not supported.   |
| <i>EHSM_ERR_WRONG_KEY_HANDLE</i> | Given key handle is wrong         |

Definition at line 685 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.8 ehsm\_get\_random\_key()

```
ehsm_uint32_t ehsm_get_random_key (
 ehsm_fw_random_key_type_e key_type,
 ehsm_fw_random_key_slot_e key_slot)
```

This function is used to install random SOC keys.

## Parameters

|    |                 |                                       |
|----|-----------------|---------------------------------------|
| in | <i>key_type</i> | Please see ehsm_fw_random_key_type_e. |
| in | <i>key_slot</i> | Please see ehsm_fw_random_key_slot_e. |

## Returns

ehsm\_uint32\_t

## Return values

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>  | No memory to handle this request. |

## Note

This function can only be called in TEST\_MODE or MANUFACTOR\_MODE.

Definition at line 129 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.9 ehsm\_hsm\_fw\_upgrade\_finish()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_finish (
 const ehsm_uint8_t * upg_encrypted_img,
 ehsm_uint32_t upg_img_size,
 ehsm_uint8_t * storage_img,
 ehsm_uint32_t * storage_img_size,
 char * mac,
 ehsm_uint32_t * mac_size)
```

Secure upgrade service finish. Only supported by eHSM firmware.

## Parameters

|    |                          |                                                                |
|----|--------------------------|----------------------------------------------------------------|
| in | <i>upg_encrypted_img</i> | Remaining encrypted code image, may be NULL.                   |
| in | <i>upg_img_size</i>      | Remaining code image size in bytes, may be 0.                  |
|    | <i>[in/out]</i>          | storage_img: Output storage encrypted code image, may be NULL. |
|    | <i>[in/out]</i>          | storage_img_size: Output code image size, may be 0.            |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 288 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.10 ehsm\_hsm\_fw\_upgrade\_init()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_init (
 const char * upg_info,
 ehsm_uint32_t upg_info_size)
```

Secure upgrade service init. Only supported by eHSM firmware.

## Parameters

|    |                      |                                      |
|----|----------------------|--------------------------------------|
| in | <i>upg_info</i>      | Upgrade info header.                 |
| in | <i>upg_info_size</i> | Size of upgrade info header in byte. |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 231 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.11 ehsm\_hsm\_fw\_upgrade\_update()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_update (
 const ehsm_uint8_t * upg_encrypted_img,
 ehsm_uint32_t upg_img_size,
 ehsm_uint8_t * storage_img,
 ehsm_uint32_t * storage_img_size)
```

Secure upgrade service update. Only supported by eHSM firmware.

## Parameters

|    |                          |                                                                                          |
|----|--------------------------|------------------------------------------------------------------------------------------|
| in | <i>upg_encrypted_img</i> | Part of the encrypted code image.                                                        |
| in | <i>upg_img_size</i>      | Input size in bytes, should be a multiple of the hash or symmetric algorithm block size. |
|    | [in/out]                 | storage_img: Output storage encrypted code image.                                        |
|    | [in/out]                 | storage_img_size: Output code image size.                                                |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 256 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.12 ehsm\_hsm\_fw\_upgrade\_verify\_finish()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_finish (
 const ehsm_uint8_t * encrypted_img,
 ehsm_uint32_t img_size)
```

Secure upgrade verify service finish. Only supported by eHSM firmware.

## Parameters

|    |                      |                                               |
|----|----------------------|-----------------------------------------------|
| in | <i>encrypted_img</i> | Remaining encrypted code image, may be NULL.  |
| in | <i>img_size</i>      | Remaining code image size in bytes, may be 0. |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 371 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.13 ehsm\_hsm\_fw\_upgrade\_verify\_init()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_init (
 const ehsm_uint8_t * code_info,
 ehsm_uint32_t code_info_size)
```

Secure upgrade verify service init. Only supported by eHSM firmware.

## Parameters

|    |                       |                                   |
|----|-----------------------|-----------------------------------|
| in | <i>code_info</i>      | Code info header.                 |
| in | <i>code_info_size</i> | Size of code info header in byte. |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 321 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.14 ehsm\_hsm\_fw\_upgrade\_verify\_update()

```
ehsm_uint32_t ehsm_hsm_fw_upgrade_verify_update (
 const ehsm_uint8_t * encrypted_img,
 ehsm_uint32_t img_size)
```

Secure upgrade verify service update. Only supported by eHSM firmware.

## Parameters

|    |                      |                                                                                                                                         |
|----|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>encrypted_img</i> | Part of the encrypted code image.                                                                                                       |
| in | <i>img_size</i>      | Input size in bytes, should be a multiple of both storage encrypt algorithm (aes or sm4) and secure boot algorithm (symmetric or hash). |

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 346 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.15 ehsm\_low\_power()

```
ehsm_uint32_t ehsm_low_power (
 ehsm_power_mode_e power_mode)
```

Request to let eHSM enter low power mode.

## Parameters

|    |                   |            |
|----|-------------------|------------|
| in | <i>power_mode</i> | power mode |
|----|-------------------|------------|

## Returns

ehsm\_uint32\_t

## Return values

|                            |           |
|----------------------------|-----------|
| <i>EHSM_ERR_SW_SUCCESS</i> | No error. |
|----------------------------|-----------|



## Return values

|                                   |                                                                     |
|-----------------------------------|---------------------------------------------------------------------|
| <i>EHSM_ERR_PARAM_ERROR</i>       | Parameter error.                                                    |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.                                   |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | No authentication is generated before, or authentication is failed. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Common error.                                                       |

Definition at line 609 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.16 ehsm\_read\_otp\_data()

```
ehsm_uint32_t ehsm_read_otp_data (
 ehsm_uint32_t otp_addr,
 ehsm_uint8_t * buf,
 ehsm_uint32_t read_size)
```

Read data from otp area.

## Parameters

|    |                  |                                                       |
|----|------------------|-------------------------------------------------------|
| in | <i>otp_addr</i>  | The address of OTP, where data being read from.       |
| in | <i>buf</i>       | The start address of OTP data buffer.                 |
| in | <i>read_size</i> | The size of buffer, The maximum length is 1024 bytes; |

## Returns

ehsm\_uint32\_t

## Return values

|                               |                                   |
|-------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>    | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>   | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>    | No memory to handle this request. |
| <i>EHSM_ERR_GENERAL_ERROR</i> | Common error.                     |

Definition at line 533 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.17 ehsm\_secure\_boot()

```
ehsm_uint32_t ehsm_secure_boot (
 ehsm_soc_image_verify_input_st * req)
```

Secure boot verification. Only supported by eHSM firmware.

## Parameters

|    |            |                                                  |
|----|------------|--------------------------------------------------|
| in | <i>req</i> | The image info (header and code) to be verified. |
|----|------------|--------------------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>            | No error.                                       |
| <i>EHSM_ERR_PARAM_ERROR</i>           | Parameter error.                                |
| <i>EHSM_ERR_OUT_OF_MEM</i>            | No memory to handle this request.               |
| <i>EHSM_ERR_WRONG_DATA_LENGTH</i>     | Wrong data length                               |
| <i>EHSM_ERR_WRONG_VERSION_COUNTER</i> | Wrong version counter                           |
| <i>EHSM_ERR_WRONG_ALGORITHM</i>       | Given algorithm or algorithm mode not available |
| <i>EHSM_ERR_WRONG_CONTEXT</i>         | Given context was wrong                         |
| <i>EHSM_ERR_DATA_CHECK_ERROR</i>      | Two parts of data is not the same               |

Definition at line 506 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.18 ehsm\_self\_test()

```
ehsm_uint32_t ehsm_self_test (
 ehsm_uint32_t flag)
```

The self test service.

## Parameters

|    |      |                                      |
|----|------|--------------------------------------|
| in | flag | The flag of algorithms to be tested. |
|----|------|--------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                                   |                                                 |
|-----------------------------------|-------------------------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>        | No error                                        |
| <i>EHSM_ERR_OUT_OF_MEM</i>        | No memory to handle this request.               |
| <i>EHSM_ERR_DEBUG_AUTH_FAILED</i> | This function is called with no authentication. |
| <i>EHSM_ERR_GENERAL_ERROR</i>     | Some algorithm self test failed.                |

Definition at line 594 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.19 ehsm\_set\_uart\_baudrate()

```
ehsm_uint32_t ehsm_set_uart_baudrate (
 ehsm_uint32_t baudrate)
```

Set uart baudrate.

## Parameters

|    |                 |                                                     |
|----|-----------------|-----------------------------------------------------|
| in | <i>baudrate</i> | baudrate to be set, refers to ehsm_uart_baudrate_e. |
|----|-----------------|-----------------------------------------------------|

## Returns

ehsm\_uint32\_t

## Return values

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>  | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i> | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>  | No memory to handle this request. |

Definition at line 623 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.41.2.20 ehsm\_write\_otp\_data()

```
ehsm_uint32_t ehsm_write_otp_data (
 ehsm_uint32_t otp_addr,
 ehsm_uint8_t * buf,
 ehsm_uint32_t write_size)
```

Write data to otp area.

## Parameters

|    |                   |                                                       |
|----|-------------------|-------------------------------------------------------|
| in | <i>otp_addr</i>   | The address of OTP, where data being write from.      |
| in | <i>buf</i>        | The start address of OTP data buffer.                 |
| in | <i>write_size</i> | The size of buffer, The maximum length is 1024 bytes; |

## Returns

ehsm\_uint32\_t

## Return values

|                                      |                                   |
|--------------------------------------|-----------------------------------|
| <i>EHSM_ERR_SW_SUCCESS</i>           | No error.                         |
| <i>EHSM_ERR_PARAM_ERROR</i>          | Parameter error.                  |
| <i>EHSM_ERR_OUT_OF_MEM</i>           | No memory to handle this request. |
| <i>EHSM_ERR_EHSM_LIFECYCLE_LIMIT</i> | function is limited by lifecycle  |
| <i>EHSM_ERR_GENERAL_ERROR</i>        | Common error.                     |

Definition at line 554 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

### 4.41.3 Variable Documentation

#### 4.41.3.1 g\_ctx

```
ehsm_uint32_t g_ctx = 0x1FFEF000
```

Definition at line 48 of file eHSM\_If\_Ext\_SysMgr\_Ip.c.

## 4.42 eHSM\_If\_Ext\_Types\_Ip.h File Reference

```
#include "eHSM_Types_Ip.h"
```

### Macros

- #define CONFIG\_SM9\_SIGN\_DEFAULT\_HID\_VALUE 0x1
- #define CONFIG\_SM9\_EXCHG\_DEFAULT\_HID\_VALUE 0x2
- #define CONFIG\_SM9\_ENC\_DEFAULT\_HID\_VALUE 0x3
- #define SM2\_PUBLIC\_KEY\_SIZE 65U
- #define SM2\_S1\_S2\_SIZE 32U

### Enumerations

- enum ehsm\_gen\_key\_alg\_e {  
 EHSM\_ALG\_RANDOM = 0U, EHSM\_ALG\_DES = 1, EHSM\_ALG\_TDES\_128 = 2, EHSM\_ALG\_TDES\_192 = 3,  
 EHSM\_ALG\_AES\_128 = 4, EHSM\_ALG\_AES\_192 = 5, EHSM\_ALG\_AES\_256 = 6, EHSM\_ALG\_SM4 = 7,  
 EHSM\_ALG\_SM2 = 8, EHSM\_ALG\_DH = 17, EHSM\_ALG\_BRAINPOOLP160R1 = 18, EHSM\_ALG\_BRAINPOOLP192R1 = 19,  
 EHSM\_ALG\_BRAINPOOLP224R1 = 20, EHSM\_ALG\_BRAINPOOLP256R1 = 21, EHSM\_ALG\_BRAINPOOLP320R1 = 22, EHSM\_ALG\_BRAINPOOLP384R1 = 23,  
 EHSM\_ALG\_BRAINPOOLP512R1 = 24, EHSM\_ALG\_SECP192R1 = 25, EHSM\_ALG\_SECP224R1 = 26, EHSM\_ALG\_SECP256R1 = 27,  
 EHSM\_ALG\_SECP384R1 = 28, EHSM\_ALG\_SECP521R1 = 29, EHSM\_ALG\_ED25519 = 30, EHSM\_ALG\_END }

### 4.42.1 Macro Definition Documentation

#### 4.42.1.1 CONFIG\_SM9\_ENC\_DEFAULT\_HID\_VALUE

```
#define CONFIG_SM9_ENC_DEFAULT_HID_VALUE 0x3
```

Definition at line 23 of file eHSM\_If\_Ext\_Types\_Ip.h.

#### 4.42.1.2 CONFIG\_SM9\_EXCHG\_DEFAULT\_HID\_VALUE

```
#define CONFIG_SM9_EXCHG_DEFAULT_HID_VALUE 0x2
```

Definition at line 22 of file eHSM\_If\_Ext\_Types\_Ip.h.

#### 4.42.1.3 CONFIG\_SM9\_SIGN\_DEFAULT\_HID\_VALUE

```
#define CONFIG_SM9_SIGN_DEFAULT_HID_VALUE 0x1
```

Definition at line 21 of file eHSM\_If\_Ext\_Types\_Ip.h.

#### 4.42.1.4 SM2\_PUBLIC\_KEY\_SIZE

```
#define SM2_PUBLIC_KEY_SIZE 65U
```

Definition at line 25 of file eHSM\_If\_Ext\_Types\_Ip.h.

#### 4.42.1.5 SM2\_S1\_S2\_SIZE

```
#define SM2_S1_S2_SIZE 32U
```

Definition at line 26 of file eHSM\_If\_Ext\_Types\_Ip.h.

### 4.42.2 Enumeration Type Documentation

#### 4.42.2.1 ehsm\_gen\_key\_alg\_e

```
enum ehsm_gen_key_alg_e
```

##### Enumerator

|                   |  |
|-------------------|--|
| EHSM_ALG_RANDOM   |  |
| EHSM_ALG_DES      |  |
| EHSM_ALG_TDES_128 |  |
| EHSM_ALG_TDES_192 |  |
| EHSM_ALG_AES_128  |  |
| EHSM_ALG_AES_192  |  |
| EHSM_ALG_AES_256  |  |
| EHSM_ALG_SM4      |  |
| EHSM_ALG_SM2      |  |
| EHSM_ALG_DH       |  |

## Enumerator

|                          |  |
|--------------------------|--|
| EHSM_ALG_BRAINPOOLP160R1 |  |
| EHSM_ALG_BRAINPOOLP192R1 |  |
| EHSM_ALG_BRAINPOOLP224R1 |  |
| EHSM_ALG_BRAINPOOLP256R1 |  |
| EHSM_ALG_BRAINPOOLP320R1 |  |
| EHSM_ALG_BRAINPOOLP384R1 |  |
| EHSM_ALG_BRAINPOOLP512R1 |  |
| EHSM_ALG_SECP192R1       |  |
| EHSM_ALG_SECP224R1       |  |
| EHSM_ALG_SECP256R1       |  |
| EHSM_ALG_SECP384R1       |  |
| EHSM_ALG_SECP521R1       |  |
| EHSM_ALG_ED25519         |  |
| EHSM_ALG_END             |  |

Definition at line 32 of file eHSM\_If\_Ext\_Types\_Ip.h.

## 4.43 eHSM\_If\_She\_ErrCode\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_If_She_Types_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_She_ErrCode_Ip.h"
#include "eHSM_Debug_Ip.h"
```

### Functions

- [ehsm\\_int32\\_t ehsm\\_she\\_convert\\_ret\\_code \(ehsm\\_int32\\_t ret\)](#)  
*convert a eHSM error code into SHE error code*

#### 4.43.1 Function Documentation

##### 4.43.1.1 ehsm\_she\_convert\_ret\_code()

```
ehsm_int32_t ehsm_she_convert_ret_code (
 ehsm_int32_t ret)
```

convert a eHSM error code into SHE error code

#### Parameters

|    |     |                                                                                |
|----|-----|--------------------------------------------------------------------------------|
| in | ret | error code of eHSM, refers to error code in <a href="#">eHSM_Err_Code_Ip.h</a> |
|----|-----|--------------------------------------------------------------------------------|

**Returns**

SHE error code, refers to error code in [eHSM\\_If\\_She\\_Types\\_Ip.h](#)

Definition at line 52 of file eHSM\_If\_She\_ErrCode\_Ip.c.

**4.44 eHSM\_If\_She\_ErrCode\_Ip.h File Reference**

```
#include "eHSM_Types_Ip.h"
```

**Functions**

- [ehsm\\_int32\\_t ehsm\\_she\\_convert\\_ret\\_code \(ehsm\\_int32\\_t ret\)](#)  
*convert a eHSM error code into SHE error code*

**4.44.1 Function Documentation****4.44.1.1 ehsm\_she\_convert\_ret\_code()**

```
ehsm_int32_t ehsm_she_convert_ret_code (
 ehsm_int32_t ret)
```

convert a eHSM error code into SHE error code

**Parameters**

|           |            |                                                                                |
|-----------|------------|--------------------------------------------------------------------------------|
| <b>in</b> | <b>ret</b> | error code of eHSM, refers to error code in <a href="#">eHSM_Err_Code_Ip.h</a> |
|-----------|------------|--------------------------------------------------------------------------------|

**Returns**

SHE error code, refers to error code in [eHSM\\_If\\_She\\_Types\\_Ip.h](#)

Definition at line 52 of file eHSM\_If\_She\_ErrCode\_Ip.c.

**4.45 eHSM\_If\_She\_Ip.c File Reference**

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include <string.h>
#include "eHSM_If_She_Ip.h"
#include "eHSM_If_She_Types_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Err_Code_Ip.h"
```

```
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Mailbox_CmdId_Ip.h"
#include "eHSM_If_Ext_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_If_She_ErrCode_Ip.h"
#include "eHSM_Dspt_Ip.h"
#include "eHSM_Exclusive_Area.h"
```

## Macros

- #define `EHSM_SOC_BOOT_STATUS_OK` (0x01U)
- #define `EHSM_SOC_BOOT_STATUS_FAIL` (0x02U)
- #define `CONFIG_EHSM_KMGR_V_SHE_K_MIN` (0x04U)
- #define `CONFIG_EHSM_KMGR_V_SHE_BOOT_MAC_K` (0x02U)

## Functions

- `ehsm_uint32_t she_crypto_ecb_extend` (`ehsm_uint32_t` key\_id, const `ehsm_uint8_t` \*in, `ehsm_uint8_t` \*out, `ehsm_uint32_t` size, `ehsm_uint32_t` direction)  
*SHE encryption or decryption in ECB mode. No padding is supported.*
- `ehsm_uint32_t she_crypto_ecb` (`ehsm_uint32_t` key\_id, const `ehsm_uint8_t` \*in, `ehsm_uint8_t` \*out, `ehsm_uint32_t` direction)  
*SHE encryption or decryption in ECB mode. No padding is supported, only support 128 bit input text.*
- `ehsm_uint32_t she_crypto_cbc` (`ehsm_uint32_t` key\_id, const `ehsm_uint8_t` \*iv, const `ehsm_uint8_t` \*in, `ehsm_uint8_t` \*out, `ehsm_uint32_t` size, `ehsm_uint32_t` direction)  
*SHE encryption or decryption in CBC mode. No padding is supported.*
- `ehsm_uint32_t she_generate_mac` (`ehsm_uint32_t` key\_id, const `ehsm_uint8_t` \*msg, `ehsm_uint32_t` size, `ehsm_uint8_t` \*mac)  
*SHE mac generation. No padding is supported.*
- `ehsm_uint32_t she_verify_mac` (`ehsm_uint32_t` key\_id, const `ehsm_uint8_t` \*msg, `ehsm_uint32_t` size, const `ehsm_uint8_t` \*mac, `ehsm_uint32_t` mac\_size, `ehsm_uint32_t` \*vrf\_status)  
*SHE mac verification. No padding is supported.*
- `ehsm_uint32_t she_load_key` (const `ehsm_uint8_t` \*m1, const `ehsm_uint8_t` \*m2, const `ehsm_uint8_t` \*m3, `ehsm_uint8_t` \*m4, `ehsm_uint8_t` \*m5)  
*she load key operation*
- `ehsm_uint32_t she_load_key_extend` (const `ehsm_uint8_t` \*m1, const `ehsm_uint8_t` \*m2, const `ehsm_uint8_t` \*m3, `ehsm_uint8_t` \*m4, `ehsm_uint8_t` \*m5)  
*she load key operation*
- `ehsm_uint32_t she_load_plain_key` (const `ehsm_uint8_t` \*key)  
*SHE load plain key operation.*
- `ehsm_uint32_t she_export_ram_key` (`ehsm_uint8_t` \*m1, `ehsm_uint8_t` \*m2, `ehsm_uint8_t` \*m3, `ehsm_uint8_t` \*m4, `ehsm_uint8_t` \*m5)  
*she export ram key operation*
- `ehsm_uint32_t she_init_rng` (void)  
*Initializes the random number generator.*
- `ehsm_uint32_t she_extend_seed` (void)  
*Reseed the random number generator.*
- `ehsm_uint32_t she_rnd` (`ehsm_uint8_t` \*random\_data\_addr)  
*Generate 128 bit random number.*
- `ehsm_uint32_t she_secure_boot` (`ehsm_uint32_t` size, const `ehsm_uint8_t` \*data)  
*This function is called for the SOC secure boot in parallel.*
- `ehsm_uint32_t she_get_status` (`ehsm_uint8_t` \*sreg)



*This function returns the eHSM status.*

- `ehsm_uint32_t she_get_id` (const `ehsm_uint8_t` \*challenge, `ehsm_uint8_t` \*id, `ehsm_uint8_t` \*sreg, `ehsm_uint8_t` \*mac)

*The function returns the identity (UID) and the value of the status register protected by a MAC over a challenge and the data.*

- `ehsm_uint32_t ehsm_she_cancel` (void)

*Job cancel.*

- `ehsm_uint32_t she_debug` (`ehsm_uint8_t` \*challenge, const `ehsm_uint8_t` \*auth)

*Debug authentication.*

- `ehsm_uint32_t she_boot_ok` (void)

*This function is called to enable boot success status.*

- `ehsm_uint32_t she_boot_failure` (void)

*This function is called to enable boot failure status.*

## 4.45.1 Macro Definition Documentation

### 4.45.1.1 CONFIG\_EHSM\_KMGR\_V\_SHE\_BOOT\_MAC\_K

```
#define CONFIG_EHSM_KMGR_V_SHE_BOOT_MAC_K (0x02U)
```

Definition at line 36 of file `eHSM_If_She_Ip.c`.

### 4.45.1.2 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_MIN

```
#define CONFIG_EHSM_KMGR_V_SHE_K_MIN (0x04U)
```

Definition at line 35 of file `eHSM_If_She_Ip.c`.

### 4.45.1.3 EHSM\_SOC\_BOOT\_STATUS\_FAIL

```
#define EHSM_SOC_BOOT_STATUS_FAIL (0x02U)
```

Definition at line 34 of file `eHSM_If_She_Ip.c`.

### 4.45.1.4 EHSM\_SOC\_BOOT\_STATUS\_OK

```
#define EHSM_SOC_BOOT_STATUS_OK (0x01U)
```

Definition at line 33 of file `eHSM_If_She_Ip.c`.

## 4.45.2 Function Documentation

### 4.45.2.1 ehsm\_she\_cancel()

```
ehsm_uint32_t ehsm_she_cancel (
 void)
```

Job cancel.

Returns

`ehsm_uint32_t` `ERC_NO_ERROR` for success, other values for error.

## Return values

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>      | No error has occurred and the command will be executed.                                              |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE. |

## Note

Definition at line 546 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.2 she\_boot\_failure()

```
ehsm_uint32_t she_boot_failure (
 void)
```

This function is called to enable boot failure status.

## Returns

ehsm\_uint32\_t Result of enable boot failure status.

## Return values

|                           |                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | Operation is successful.                                                                                                                                                                                                                                                                                                                  |
| <i>ERC_NO_SECURE_BOOT</i> | 1) If sequential boot is enabled, 2) or parallel boot is enabled, but she_secure_boot is not called, 3) or she_secure_boot is called but failed, 4) or she_secure_boot is successful, but she_boot_ok has been called successfully before, 5) or she_boot_ok is not successful, but she_boot_failure has been called successfully before. |
| <i>ERC_BUSY</i>           | The request can not be handled.                                                                                                                                                                                                                                                                                                           |

## Note

This function must be called after she\_secure\_boot.

Definition at line 600 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.3 she\_boot\_ok()

```
ehsm_uint32_t she_boot_ok (
 void)
```

This function is called to enable boot success status.

## Returns

ehsm\_uint32\_t Result of enable boot success status.

## Return values

|                           |                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | Operation is successful.                                                                                                                                                                                                                                                                                                                  |
| <i>ERC_NO_SECURE_BOOT</i> | 1) If sequential boot is enabled, 2) or parallel boot is enabled, but she_secure_boot is not called, 3) or she_secure_boot is called but failed, 4) or she_secure_boot is successful, but she_boot_ok has been called successfully before, 5) or she_boot_ok is not successful, but she_boot_failure has been called successfully before. |
| <i>ERC_BUSY</i>           | The request can not be handled.                                                                                                                                                                                                                                                                                                           |

## Note

This function must be called after she\_secure\_boot.

Definition at line 584 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.4 she\_crypto\_cbc()

```
ehsm_uint32_t she_crypto_cbc (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * iv,
 const ehsm_uint8_t * in,
 ehsm_uint8_t * out,
 ehsm_uint32_t size,
 ehsm_uint32_t direction)
```

SHE encryption or decryption in CBC mode. No padding is supported.

## Parameters

|    |                  |                                                               |
|----|------------------|---------------------------------------------------------------|
| in | <i>key_id</i>    | SHE key index, only [4, 54] are valid.                        |
| in | <i>iv</i>        | Iv for CBC mode, size should be 16.                           |
| in | <i>in</i>        | Plaintext for encryption (or ciphertext for decryption).      |
|    | <i>[in/out]</i>  | out: Ciphertext for encryption (or plaintext for decryption). |
| in | <i>size</i>      | Size of input data in bytes, should be a multiple of 16.      |
| in | <i>direction</i> | Encryption or decryption.                                     |

## Returns

0 for success, a negative value for error.

Definition at line 179 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.5 she\_crypto\_ecb()

```
ehsm_uint32_t she_crypto_ecb (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * in,
 ehsm_uint8_t * out,
 ehsm_uint32_t direction)
```

SHE encryption or decryption in ECB mode. No padding is supported, only support 128 bit input text.

## Parameters

|    |                  |                                                               |
|----|------------------|---------------------------------------------------------------|
| in | <i>key_id</i>    | SHE key index, only [4, 54] are valid.                        |
| in | <i>in</i>        | Plaintext for encryption (or ciphertext for decryption).      |
|    | <i>[in/out]</i>  | out: Ciphertext for encryption (or plaintext for decryption). |
| in | <i>direction</i> | Encryption or decryption.                                     |

## Returns

0 for success, a negative value for error.

Definition at line 170 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.6 she\_crypto\_ecb\_extend()

```
ehsm_uint32_t she_crypto_ecb_extend (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * in,
 ehsm_uint8_t * out,
 ehsm_uint32_t size,
 ehsm_uint32_t direction)
```

SHE encryption or decryption in ECB mode. No padding is supported.

## Parameters

|    |                  |                                                               |
|----|------------------|---------------------------------------------------------------|
| in | <i>key_id</i>    | SHE key index, only [4, 54] are valid.                        |
| in | <i>in</i>        | Plaintext for encryption (or ciphertext for decryption).      |
|    | <i>[in/out]</i>  | out: Ciphertext for encryption (or plaintext for decryption). |
| in | <i>size</i>      | Size of input data in bytes, should be a multiple of 16.      |
| in | <i>direction</i> | Encryption or decryption.                                     |

## Returns

0 for success, a negative value for error.

Definition at line 128 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.7 she\_debug()

```
ehsm_uint32_t she_debug (
 ehsm_uint8_t * challenge,
 const ehsm_uint8_t * auth)
```

Debug authentication.

## Parameters

|    |                  |                              |
|----|------------------|------------------------------|
| in | <i>challenge</i> | The 128 bits challenge.      |
| in | <i>auth</i>      | The 128 bits authentication. |

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for successful, other values for failure.

## Return values

|                                |                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>            | No error has occurred and the command will be executed.                                                                                                                                  |
| <i>ERC_KEY_WRITE_PROTECTED</i> | This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected. |
| <i>ERC_NO_DEBUGGING</i>        | The error code is returned if internal debugging is not possible because the authentication with the challenge response protocol did not succeed.                                        |
| <i>ERC_MEMORY_FAILURE</i>      | This error code can be returned if the underlying memory technology is able to detect physical errors.                                                                                   |
| <i>ERC_GENERAL_ERROR</i>       | This error code is returned if an error not covered by the error codes above is detected inside SHE.                                                                                     |

## Note

Definition at line 552 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.8 she\_export\_ram\_key()

```
ehsm_uint32_t she_export_ram_key (
 ehsm_uint8_t * m1,
 ehsm_uint8_t * m2,
 ehsm_uint8_t * m3,
 ehsm_uint8_t * m4,
 ehsm_uint8_t * m5)
```

she export ram key operation

## Parameters

|     |           |                                     |
|-----|-----------|-------------------------------------|
| out | <i>m1</i> | m1 in she spec. should be 16 bytes. |
| out | <i>m2</i> | m2 in she spec. should be 32 bytes. |
| out | <i>m3</i> | m3 in she spec. should be 16 bytes. |
| out | <i>m4</i> | m4 in she spec. should be 32 bytes. |
| out | <i>m5</i> | m5 in she spec. should be 16 bytes. |

**Returns**

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

**Return values**

|                           |                                                                                                                                                  |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | No error has occurred and the command will be executed.                                                                                          |
| <i>ERC_KEY_INVALID</i>    | This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation. |
| <i>ERC_MEMORY_FAILURE</i> | This error code can be returned if the underlying memory technology is able to detect physical errors.                                           |
| <i>ERC_KEY_EMPTY</i>      | This error code is returned by SHE if the application attempts to use a key that has not been initialized yet.                                   |
| <i>ERC_GENERAL_ERROR</i>  | This error code is returned if an error not covered by the error codes above is detected inside SHE.                                             |

**Note**

Definition at line 372 of file eHSM\_If\_She\_Ip.c.

**4.45.2.9 she\_extend\_seed()**

```
ehsm_uint32_t she_extend_seed (
 void)
```

Reseed the random number generator.

**Returns**

ehsm\_uint32\_t

**Return values**

|                     |                      |
|---------------------|----------------------|
| <i>ERC_NO_ERROR</i> | Successfully execute |
|---------------------|----------------------|

**Note**

Since the rng will be reseeded by hardware, this function will do nothing

Definition at line 412 of file eHSM\_If\_She\_Ip.c.

**4.45.2.10 she\_generate\_mac()**

```
ehsm_uint32_t she_generate_mac (
 ehsm_uint32_t key_id,
```

```
const ehsm_uint8_t * msg,
 ehsm_uint32_t size,
 ehsm_uint8_t * mac)
```

SHE mac generation. No padding is supported.

## Parameters

|    |                     |                                                             |
|----|---------------------|-------------------------------------------------------------|
| in | <i>key↔<br/>_id</i> | SHE key index, only [4, 54] are valid.                      |
| in | <i>msg</i>          | Message.                                                    |
| in | <i>size</i>         | Size of message in bytes, should be a multiple of 16.       |
|    | <i>[in/out]</i>     | mac: The generated mac, size should not be smaller than 16. |

## Returns

0 for success, a negative value for error.

Definition at line 225 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.11 she\_get\_id()

```
ehsm_uint32_t she_get_id (
 const ehsm_uint8_t * challenge,
 ehsm_uint8_t * id,
 ehsm_uint8_t * sreg,
 ehsm_uint8_t * mac)
```

The function returns the identity (UID) and the value of the status register protected by a MAC over a challenge and the data.

## Parameters

|    |                  |                                                           |
|----|------------------|-----------------------------------------------------------|
| in | <i>challenge</i> | The 128 bits challenge.                                   |
| in | <i>id</i>        | The output buffer to store UID, should be 120 bits.       |
| in | <i>sreg</i>      | The output buffer to store status info, should be 8 bits. |
| in | <i>mac</i>       | The output buffer to store MAC, should be 128 bits.       |

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                              |                                                                                                        |
|------------------------------|--------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>          | No error has occurred and the command will be executed.                                                |
| <i>ERC_KEY_NOT_AVAILABLE</i> | This error code is returned if a key is locked due to failed boot measurement or an active debugger.   |
| <i>ERC_MEMORY_FAILURE</i>    | This error code can be returned if the underlying memory technology is able to detect physical errors. |
| <i>ERC_GENERAL_ERROR</i>     | This error code is returned if an error not covered by the error codes above is detected inside SHE.   |

## Note



Definition at line 505 of file eHSM\_If\_She\_Ip.c.

4.45.2.12 she\_get\_status()

```
ehsm_uint32_t she_get_status (
 ehsm_uint8_t * sreg)
```

This function returns the eHSM status.

Parameters

|    |      |                                                          |
|----|------|----------------------------------------------------------|
| in | sreg | Buffer to store the status infomation, should be 1 byte. |
|----|------|----------------------------------------------------------|

Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

Return values

|                   |                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------|
| ERC_NO_ERROR      | No error has occurred and the command will be executed.                                              |
| ERC_GENERAL_ERROR | This error code is returned if an error not covered by the error codes above is detected inside SHE. |

Note

Definition at line 488 of file eHSM\_If\_She\_Ip.c.

4.45.2.13 she\_init\_rng()

```
ehsm_uint32_t she_init_rng (
 void)
```

Initializes the random number generator.

Returns

ehsm\_uint32\_t

Return values

|              |                      |
|--------------|----------------------|
| ERC_NO_ERROR | Successfully execute |
|--------------|----------------------|

**Note**

Since the rng will be initialized by hardware, this function will do nothing

Definition at line 407 of file eHSM\_If\_She\_Ip.c.

**4.45.2.14 she\_load\_key()**

```
ehsm_uint32_t she_load_key (
 const ehsm_uint8_t * m1,
 const ehsm_uint8_t * m2,
 const ehsm_uint8_t * m3,
 ehsm_uint8_t * m4,
 ehsm_uint8_t * m5)
```

she load key operation

**Parameters**

|     |           |                                     |
|-----|-----------|-------------------------------------|
| in  | <i>m1</i> | m1 in she spec. should be 16 bytes. |
| in  | <i>m2</i> | m2 in she spec. should be 32 bytes. |
| in  | <i>m3</i> | m3 in she spec. should be 16 bytes. |
| out | <i>m4</i> | m4 in she spec. should be 32 bytes. |
| out | <i>m5</i> | m5 in she spec. should be 16 bytes. |

**Returns**

ehsm\_int32\_t ERC\_NO\_ERROR for success, other values for error.

**Return values**

|                                |                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>            | No error has occurred and the command will be executed.                                                                                                                                  |
| <i>ERC_KEY_NOT_AVAILABLE</i>   | This error code is returned if a key is locked due to failed boot measurement or an active debugger.                                                                                     |
| <i>ERC_KEY_INVALID</i>         | This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation.                                         |
| <i>ERC_KEY_WRITE_PROTECTED</i> | This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected. |
| <i>ERC_KEY_UPDATE_ERROR</i>    | This error is returned when a key update did not succeed due to errors in verification of the messages.                                                                                  |
| <i>ERC_MEMORY_FAILURE</i>      | This error code can be returned if the underlying memory technology is able to detect physical errors.                                                                                   |
| <i>ERC_KEY_EMPTY</i>           | This error code is returned by SHE if the application attempts to use a key that has not been initialized yet.                                                                           |
| <i>ERC_GENERAL_ERROR</i>       | This error code is returned if an error not covered by the error codes above is detected inside SHE.                                                                                     |

**Note**

Definition at line 313 of file eHSM\_If\_She\_Ip.c.

#### 4.45.2.15 she\_load\_key\_extend()

```
ehsm_uint32_t she_load_key_extend (
 const ehsm_uint8_t * m1,
 const ehsm_uint8_t * m2,
 const ehsm_uint8_t * m3,
 ehsm_uint8_t * m4,
 ehsm_uint8_t * m5)
```

she load key operation

##### Parameters

|     |           |                                     |
|-----|-----------|-------------------------------------|
| in  | <i>m1</i> | m1 in she spec. should be 16 bytes. |
| in  | <i>m2</i> | m2 in she spec. should be 32 bytes. |
| in  | <i>m3</i> | m3 in she spec. should be 16 bytes. |
| out | <i>m4</i> | m4 in she spec. should be 32 bytes. |
| out | <i>m5</i> | m5 in she spec. should be 16 bytes. |

##### Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

##### Return values

|                                |                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>            | No error has occurred and the command will be executed.                                                                                                                                  |
| <i>ERC_KEY_NOT_AVAILABLE</i>   | This error code is returned if a key is locked due to failed boot measurement or an active debugger.                                                                                     |
| <i>ERC_KEY_INVALID</i>         | This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation.                                         |
| <i>ERC_KEY_WRITE_PROTECTED</i> | This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected. |
| <i>ERC_KEY_UPDATE_ERROR</i>    | This error is returned when a key update did not succeed due to errors in verification of the messages.                                                                                  |
| <i>ERC_MEMORY_FAILURE</i>      | This error code can be returned if the underlying memory technology is able to detect physical errors.                                                                                   |
| <i>ERC_KEY_EMPTY</i>           | This error code is returned by SHE if the application attempts to use a key that has not been initialized yet.                                                                           |
| <i>ERC_GENERAL_ERROR</i>       | This error code is returned if an error not covered by the error codes above is detected inside SHE.                                                                                     |

##### Note

Definition at line 330 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.16 she\_load\_plain\_key()

```
ehsm_uint32_t she_load_plain_key (
 const ehsm_uint8_t * key)
```

SHE load plain key operation.

## Parameters

|    |     |                           |
|----|-----|---------------------------|
| in | key | 16 bytes key in raw data. |
|----|-----|---------------------------|

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>      | No error has occurred and the command will be executed.                                              |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE. |

## Note

Definition at line 347 of file eHSM\_If\_She\_Ip.c.

## 4.45.2.17 she\_rnd()

```
ehsm_uint32_t she_rnd (
 ehsm_uint8_t * random_data_addr)
```

Generate 128 bit random number.

## Parameters

|    |                  |  |
|----|------------------|--|
| in | random_data_addr |  |
|----|------------------|--|

## Returns

ehsm\_uint32\_t

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | No error has occurred and the command will be executed     |
| <i>ERC_SEQUENCE_ERROR</i> | The sequence of commands or subcommands is out of sequence |
| <i>ERC_RNG_SEED</i>       | The seed has not been initialized before                   |
| <i>ERC_GENERAL_ERROR</i>  | Other error code that is not include in SHE spec           |

Note

Definition at line 417 of file eHSM\_If\_She\_Ip.c.

4.45.2.18 she\_secure\_boot()

```
ehsm_uint32_t she_secure_boot (
 ehsm_uint32_t size,
 const ehsm_uint8_t * data)
```

This functon is called for the SOC secure boot in parallel.

Parameters

|    |      |                                                                         |
|----|------|-------------------------------------------------------------------------|
| in | size | The SOC image size for secure boot. Should be (1024, 255 * 1024] bytes. |
| in | data | The SOC image data.                                                     |

Returns

ehsm\_uint32\_t Secure boot result.

Return values

|                    |                                                               |
|--------------------|---------------------------------------------------------------|
| ERC_NO_ERROR       | Secure boot is successful.                                    |
| ERC_GENERAL_ERROR  | Secure boot failed since input is illegal.                    |
| ERC_NO_SECURE_BOOT | Secure boot verification failed or secure boot has been done. |
| ERC_BUSY           | The request can not be handled.                               |

Note

This function can be called only once every power on. And can be called only parallel boot is enabled.

Definition at line 447 of file eHSM\_If\_She\_Ip.c.

4.45.2.19 she\_verify\_mac()

```
ehsm_uint32_t she_verify_mac (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * msg,
 ehsm_uint32_t size,
 const ehsm_uint8_t * mac,
 ehsm_uint32_t mac_size,
 ehsm_uint32_t * vrf_status)
```

SHE mac verification. No padding is supported.

## Parameters

|    |                 |                                                          |
|----|-----------------|----------------------------------------------------------|
| in | <i>key_id</i>   | SHE key index, only [4, 54] are valid.                   |
| in | <i>msg</i>      | Message.                                                 |
| in | <i>size</i>     | Size of message in bytes, should be a multiple of 16.    |
| in | <i>mac</i>      | The mac to be verified.                                  |
| in | <i>mac_size</i> | The mac size in byte to be verified. Belongs to [1, 16]. |
|    | <i>[in/out]</i> | vrf_status: 1 for mac matching, 0 for not matching.      |

## Returns

0 for success, the result is written to vrf\_status. A negative value for error.

Definition at line 268 of file eHSM\_If\_She\_Ip.c.

## 4.46 eHSM\_If\_She\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Types_Ip.h"
```

## Macros

- #define SM4\_CTRDRBG 0
- #define AES\_CTRDRBG 1

## Functions

- `ehsm_uint32_t she_crypto_ecb_extend (ehsm_uint32_t key_id, const ehsm_uint8_t *in, ehsm_uint8_t *out, ehsm_uint32_t size, ehsm_uint32_t direction)`  
*SHE encryption or decryption in ECB mode. No padding is supported.*
- `ehsm_uint32_t she_crypto_ecb (ehsm_uint32_t key_id, const ehsm_uint8_t *in, ehsm_uint8_t *out, ehsm_uint32_t size, ehsm_uint32_t direction)`  
*SHE encryption or decryption in ECB mode. No padding is supported, only support 128 bit input text.*
- `ehsm_uint32_t she_crypto_cbc (ehsm_uint32_t key_id, const ehsm_uint8_t *iv, const ehsm_uint8_t *in, ehsm_uint8_t *out, ehsm_uint32_t size, ehsm_uint32_t direction)`  
*SHE encryption or decryption in CBC mode. No padding is supported.*
- `ehsm_uint32_t she_generate_mac (ehsm_uint32_t key_id, const ehsm_uint8_t *msg, ehsm_uint32_t size, ehsm_uint8_t *mac)`  
*SHE mac generation. No padding is supported.*
- `ehsm_uint32_t she_verify_mac (ehsm_uint32_t key_id, const ehsm_uint8_t *msg, ehsm_uint32_t size, const ehsm_uint8_t *mac, ehsm_uint32_t mac_size, ehsm_uint32_t *vrf_status)`  
*SHE mac verification. No padding is supported.*
- `ehsm_uint32_t she_load_key (const ehsm_uint8_t *m1, const ehsm_uint8_t *m2, const ehsm_uint8_t *m3, ehsm_uint8_t *m4, ehsm_uint8_t *m5)`  
*she load key operation*
- `ehsm_uint32_t she_load_key_extend (const ehsm_uint8_t *m1, const ehsm_uint8_t *m2, const ehsm_uint8_t *m3, ehsm_uint8_t *m4, ehsm_uint8_t *m5)`  
*she load key operation*

- `ehsm_uint32_t she_load_plain_key` (const `ehsm_uint8_t` \*key)  
*SHE load plain key operation.*
- `ehsm_uint32_t she_export_ram_key` (`ehsm_uint8_t` \*m1, `ehsm_uint8_t` \*m2, `ehsm_uint8_t` \*m3, `ehsm_uint8_t` \*m4, `ehsm_uint8_t` \*m5)  
*she export ram key operation*
- `ehsm_uint32_t she_rnd` (`ehsm_uint8_t` \*random\_data\_addr)  
*Generate 128 bit random number.*
- `ehsm_uint32_t she_init_rng` (void)  
*Initializes the random number generator.*
- `ehsm_uint32_t she_extend_seed` (void)  
*Reseed the random number generator.*
- `ehsm_uint32_t she_secure_boot` (`ehsm_uint32_t` size, const `ehsm_uint8_t` \*data)  
*This function is called for the SOC secure boot in parallel.*
- `ehsm_uint32_t she_boot_ok` (void)  
*This function is called to enable boot success status.*
- `ehsm_uint32_t she_boot_failure` (void)  
*This function is called to enable boot failure status.*
- `ehsm_uint32_t she_get_status` (`ehsm_uint8_t` \*sreg)  
*This function returns the eHSM status.*
- `ehsm_uint32_t she_get_id` (const `ehsm_uint8_t` \*challenge, `ehsm_uint8_t` \*id, `ehsm_uint8_t` \*sreg, `ehsm_uint8_t` \*mac)  
*The function returns the identity (UID) and the value of the status register protected by a MAC over a challenge and the data.*
- `ehsm_uint32_t ehsm_she_cancel` (void)  
*Job cancel.*
- `ehsm_uint32_t she_debug` (`ehsm_uint8_t` \*challenge, const `ehsm_uint8_t` \*auth)  
*Debug authentication.*

## 4.46.1 Macro Definition Documentation

### 4.46.1.1 AES\_CTRDRBG

```
#define AES_CTRDRBG 1
```

Definition at line 21 of file eHSM\_If\_She\_Ip.h.

### 4.46.1.2 SM4\_CTRDRBG

```
#define SM4_CTRDRBG 0
```

Definition at line 20 of file eHSM\_If\_She\_Ip.h.

## 4.46.2 Function Documentation

## 4.46.2.1 ehsm\_she\_cancel()

```
ehsm_uint32_t ehsm_she_cancel (
 void)
```

Job cancel.

**Returns**

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

**Return values**

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>      | No error has occurred and the command will be executed.                                              |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE. |

**Note**

Definition at line 546 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.2 she\_boot\_failure()

```
ehsm_uint32_t she_boot_failure (
 void)
```

This function is called to enable boot failure status.

**Returns**

ehsm\_uint32\_t Result of enable boot failure status.

**Return values**

|                           |                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | Operation is successful.                                                                                                                                                                                                                                                                                                                  |
| <i>ERC_NO_SECURE_BOOT</i> | 1) If sequential boot is enabled, 2) or parallel boot is enabled, but she_secure_boot is not called, 3) or she_secure_boot is called but failed, 4) or she_secure_boot is successful, but she_boot_ok has been called successfully before, 5) or she_boot_ok is not successful, but she_boot_failure has been called successfully before. |
| <i>ERC_BUSY</i>           | The request can not be handled.                                                                                                                                                                                                                                                                                                           |

**Note**

This function must be called after she\_secure\_boot.

Definition at line 600 of file eHSM\_If\_She\_Ip.c.



## 4.46.2.3 she\_boot\_ok()

```
ehsm_uint32_t she_boot_ok (
 void)
```

This function is called to enable boot success status.

## Returns

ehsm\_uint32\_t Result of enable boot success status.

## Return values

|                           |                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | Operation is successful.                                                                                                                                                                                                                                                                                                                  |
| <i>ERC_NO_SECURE_BOOT</i> | 1) If sequential boot is enabled, 2) or parallel boot is enabled, but she_secure_boot is not called, 3) or she_secure_boot is called but failed, 4) or she_secure_boot is successful, but she_boot_ok has been called successfully before, 5) or she_boot_ok is not successful, but she_boot_failure has been called successfully before. |
| <i>ERC_BUSY</i>           | The request can not be handled.                                                                                                                                                                                                                                                                                                           |

## Note

This function must be called after she\_secure\_boot.

Definition at line 584 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.4 she\_crypto\_cbc()

```
ehsm_uint32_t she_crypto_cbc (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * iv,
 const ehsm_uint8_t * in,
 ehsm_uint8_t * out,
 ehsm_uint32_t size,
 ehsm_uint32_t direction)
```

SHE encryption or decryption in CBC mode. No padding is supported.

## Parameters

|    |                  |                                                               |
|----|------------------|---------------------------------------------------------------|
| in | <i>key_id</i>    | SHE key index, only [4, 54] are valid.                        |
| in | <i>iv</i>        | Iv for CBC mode, size should be 16.                           |
| in | <i>in</i>        | Plaintext for encryption (or ciphertext for decryption).      |
|    | <i>[in/out]</i>  | out: Ciphertext for encryption (or plaintext for decryption). |
| in | <i>size</i>      | Size of input data in bytes, should be a multiple of 16.      |
| in | <i>direction</i> | Encryption or decryption.                                     |

## Returns

0 for success, a negative value for error.

Definition at line 179 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.5 she\_crypto\_ecb()

```
ehsm_uint32_t she_crypto_ecb (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * in,
 ehsm_uint8_t * out,
 ehsm_uint32_t direction)
```

SHE encryption or decryption in ECB mode. No padding is supported, only support 128 bit input text.

##### Parameters

|    |                  |                                                               |
|----|------------------|---------------------------------------------------------------|
| in | <i>key_id</i>    | SHE key index, only [4, 54] are valid.                        |
| in | <i>in</i>        | Plaintext for encryption (or ciphertext for decryption).      |
|    | <i>[in/out]</i>  | out: Ciphertext for encryption (or plaintext for decryption). |
| in | <i>direction</i> | Encryption or decryption.                                     |

##### Returns

0 for success, a negative value for error.

Definition at line 170 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.6 she\_crypto\_ecb\_extend()

```
ehsm_uint32_t she_crypto_ecb_extend (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * in,
 ehsm_uint8_t * out,
 ehsm_uint32_t size,
 ehsm_uint32_t direction)
```

SHE encryption or decryption in ECB mode. No padding is supported.

##### Parameters

|    |                  |                                                               |
|----|------------------|---------------------------------------------------------------|
| in | <i>key_id</i>    | SHE key index, only [4, 54] are valid.                        |
| in | <i>in</i>        | Plaintext for encryption (or ciphertext for decryption).      |
|    | <i>[in/out]</i>  | out: Ciphertext for encryption (or plaintext for decryption). |
| in | <i>size</i>      | Size of input data in bytes, should be a multiple of 16.      |
| in | <i>direction</i> | Encryption or decryption.                                     |

##### Returns

0 for success, a negative value for error.

Definition at line 128 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.7 she\_debug()

```
ehsm_uint32_t she_debug (
 ehsm_uint8_t * challenge,
 const ehsm_uint8_t * auth)
```

Debug authentication.

##### Parameters

|    |                  |                              |
|----|------------------|------------------------------|
| in | <i>challenge</i> | The 128 bits challenge.      |
| in | <i>auth</i>      | The 128 bits authentication. |

##### Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for successful, other values for failure.

##### Return values

|                                |                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>            | No error has occurred and the command will be executed.                                                                                                                                  |
| <i>ERC_KEY_WRITE_PROTECTED</i> | This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected. |
| <i>ERC_NO_DEBUGGING</i>        | The error code is returned if internal debugging is not possible because the authentication with the challenge response protocol did not succeed.                                        |
| <i>ERC_MEMORY_FAILURE</i>      | This error code can be returned if the underlying memory technology is able to detect physical errors.                                                                                   |
| <i>ERC_GENERAL_ERROR</i>       | This error code is returned if an error not covered by the error codes above is detected inside SHE.                                                                                     |

##### Note

Definition at line 552 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.8 she\_export\_ram\_key()

```
ehsm_uint32_t she_export_ram_key (
 ehsm_uint8_t * m1,
 ehsm_uint8_t * m2,
 ehsm_uint8_t * m3,
 ehsm_uint8_t * m4,
 ehsm_uint8_t * m5)
```

she export ram key operation

## Parameters

|     |           |                                     |
|-----|-----------|-------------------------------------|
| out | <i>m1</i> | m1 in she spec. should be 16 bytes. |
| out | <i>m2</i> | m2 in she spec. should be 32 bytes. |
| out | <i>m3</i> | m3 in she spec. should be 16 bytes. |
| out | <i>m4</i> | m4 in she spec. should be 32 bytes. |
| out | <i>m5</i> | m5 in she spec. should be 16 bytes. |

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                           |                                                                                                                                                  |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | No error has occurred and the command will be executed.                                                                                          |
| <i>ERC_KEY_INVALID</i>    | This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation. |
| <i>ERC_MEMORY_FAILURE</i> | This error code can be returned if the underlying memory technology is able to detect physical errors.                                           |
| <i>ERC_KEY_EMPTY</i>      | This error code is returned by SHE if the application attempts to use a key that has not been initialized yet.                                   |
| <i>ERC_GENERAL_ERROR</i>  | This error code is returned if an error not covered by the error codes above is detected inside SHE.                                             |

## Note

Definition at line 372 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.9 she\_extend\_seed()

```
ehsm_uint32_t she_extend_seed (
 void)
```

Reseed the random number generator.

## Returns

ehsm\_uint32\_t

## Return values

|                     |                      |
|---------------------|----------------------|
| <i>ERC_NO_ERROR</i> | Successfully execute |
|---------------------|----------------------|

## Note

Since the rng will be reseeded by hardware, this function will do nothing

Definition at line 412 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.10 she\_generate\_mac()

```
ehsm_uint32_t she_generate_mac (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * msg,
 ehsm_uint32_t size,
 ehsm_uint8_t * mac)
```

SHE mac generation. No padding is supported.

##### Parameters

|    |                 |                                                             |
|----|-----------------|-------------------------------------------------------------|
| in | <i>key_id</i>   | SHE key index, only [4, 54] are valid.                      |
| in | <i>msg</i>      | Message.                                                    |
| in | <i>size</i>     | Size of message in bytes, should be a multiple of 16.       |
|    | <i>[in/out]</i> | mac: The generated mac, size should not be smaller than 16. |

##### Returns

0 for success, a negative value for error.

Definition at line 225 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.11 she\_get\_id()

```
ehsm_uint32_t she_get_id (
 const ehsm_uint8_t * challenge,
 ehsm_uint8_t * id,
 ehsm_uint8_t * sreg,
 ehsm_uint8_t * mac)
```

The function returns the identity (UID) and the value of the status register protected by a MAC over a challenge and the data.

##### Parameters

|    |                  |                                                           |
|----|------------------|-----------------------------------------------------------|
| in | <i>challenge</i> | The 128 bits challenge.                                   |
| in | <i>id</i>        | The output buffer to store UID, should be 120 bits.       |
| in | <i>sreg</i>      | The output buffer to store status info, should be 8 bits. |
| in | <i>mac</i>       | The output buffer to store MAC, should be 128 bits.       |

##### Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                              |                                                                                                        |
|------------------------------|--------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>          | No error has occurred and the command will be executed.                                                |
| <i>ERC_KEY_NOT_AVAILABLE</i> | This error code is returned if a key is locked due to failed boot measurement or an active debugger.   |
| <i>ERC_MEMORY_FAILURE</i>    | This error code can be returned if the underlying memory technology is able to detect physical errors. |
| <i>ERC_GENERAL_ERROR</i>     | This error code is returned if an error not covered by the error codes above is detected inside SHE.   |

## Note

Definition at line 505 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.12 she\_get\_status()

```
ehsm_uint32_t she_get_status (
 ehsm_uint8_t * sreg)
```

This function returns the eHSM status.

## Parameters

|    |      |                                                           |
|----|------|-----------------------------------------------------------|
| in | sreg | Buffer to store the status information, should be 1 byte. |
|----|------|-----------------------------------------------------------|

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>      | No error has occurred and the command will be executed.                                              |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE. |

## Note

Definition at line 488 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.13 she\_init\_rng()

```
ehsm_uint32_t she_init_rng (
 void)
```

Initializes the random number generator.

## Returns

ehsm\_uint32\_t

## Return values

|                     |                      |
|---------------------|----------------------|
| <i>ERC_NO_ERROR</i> | Successfully execute |
|---------------------|----------------------|

## Note

Since the rng will be initialized by hardware, this function will do nothing

Definition at line 407 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.14 she\_load\_key()

```
ehsm_uint32_t she_load_key (
 const ehsm_uint8_t * m1,
 const ehsm_uint8_t * m2,
 const ehsm_uint8_t * m3,
 ehsm_uint8_t * m4,
 ehsm_uint8_t * m5)
```

she load key operation

## Parameters

|     |           |                                     |
|-----|-----------|-------------------------------------|
| in  | <i>m1</i> | m1 in she spec. should be 16 bytes. |
| in  | <i>m2</i> | m2 in she spec. should be 32 bytes. |
| in  | <i>m3</i> | m3 in she spec. should be 16 bytes. |
| out | <i>m4</i> | m4 in she spec. should be 32 bytes. |
| out | <i>m5</i> | m5 in she spec. should be 16 bytes. |

## Returns

ehsm\_int32\_t *ERC\_NO\_ERROR* for success, other values for error.

## Return values

|                                |                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>            | No error has occurred and the command will be executed.                                                                                                                                  |
| <i>ERC_KEY_NOT_AVAILABLE</i>   | This error code is returned if a key is locked due to failed boot measurement or an active debugger.                                                                                     |
| <i>ERC_KEY_INVALID</i>         | This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation.                                         |
| <i>ERC_KEY_WRITE_PROTECTED</i> | This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected. |
| <i>ERC_KEY_UPDATE_ERROR</i>    | This error is returned when a key update did not succeed due to errors in verification of the messages.                                                                                  |
| <i>ERC_MEMORY_FAILURE</i>      | This error code can be returned if the underlying memory technology is able to detect physical errors.                                                                                   |

## Return values

|                          |                                                                                                                |
|--------------------------|----------------------------------------------------------------------------------------------------------------|
| <i>ERC_KEY_EMPTY</i>     | This error code is returned by SHE if the application attempts to use a key that has not been initialized yet. |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE.           |

## Note

Definition at line 313 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.15 she\_load\_key\_extend()

```
ehsm_uint32_t she_load_key_extend (
 const ehsm_uint8_t * m1,
 const ehsm_uint8_t * m2,
 const ehsm_uint8_t * m3,
 ehsm_uint8_t * m4,
 ehsm_uint8_t * m5)
```

she load key operation

## Parameters

|     |           |                                     |
|-----|-----------|-------------------------------------|
| in  | <i>m1</i> | m1 in she spec. should be 16 bytes. |
| in  | <i>m2</i> | m2 in she spec. should be 32 bytes. |
| in  | <i>m3</i> | m3 in she spec. should be 16 bytes. |
| out | <i>m4</i> | m4 in she spec. should be 32 bytes. |
| out | <i>m5</i> | m5 in she spec. should be 16 bytes. |

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                                |                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>            | No error has occurred and the command will be executed.                                                                                                                                  |
| <i>ERC_KEY_NOT_AVAILABLE</i>   | This error code is returned if a key is locked due to failed boot measurement or an active debugger.                                                                                     |
| <i>ERC_KEY_INVALID</i>         | This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation.                                         |
| <i>ERC_KEY_WRITE_PROTECTED</i> | This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected. |
| <i>ERC_KEY_UPDATE_ERROR</i>    | This error is returned when a key update did not succeed due to errors in verification of the messages.                                                                                  |
| <i>ERC_MEMORY_FAILURE</i>      | This error code can be returned if the underlying memory technology is able to detect physical errors.                                                                                   |



## Return values

|                          |                                                                                                                |
|--------------------------|----------------------------------------------------------------------------------------------------------------|
| <i>ERC_KEY_EMPTY</i>     | This error code is returned by SHE if the application attempts to use a key that has not been initialized yet. |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE.           |

## Note

Definition at line 330 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.16 she\_load\_plain\_key()

```
ehsm_uint32_t she_load_plain_key (
 const ehsm_uint8_t * key)
```

SHE load plain key operation.

## Parameters

|    |     |                           |
|----|-----|---------------------------|
| in | key | 16 bytes key in raw data. |
|----|-----|---------------------------|

## Returns

ehsm\_uint32\_t ERC\_NO\_ERROR for success, other values for error.

## Return values

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>      | No error has occurred and the command will be executed.                                              |
| <i>ERC_GENERAL_ERROR</i> | This error code is returned if an error not covered by the error codes above is detected inside SHE. |

## Note

Definition at line 347 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.17 she\_rnd()

```
ehsm_uint32_t she_rnd (
 ehsm_uint8_t * random_data_addr)
```

Generate 128 bit random number.

## Parameters

|    |                         |  |
|----|-------------------------|--|
| in | <i>random_data_addr</i> |  |
|----|-------------------------|--|

## Returns

ehsm\_uint32\_t

## Return values

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | No error has occurred and the command will be executed     |
| <i>ERC_SEQUENCE_ERROR</i> | The sequence of commands or subcommands is out of sequence |
| <i>ERC_RNG_SEED</i>       | The seed has not been initialized before                   |
| <i>ERC_GENERAL_ERROR</i>  | Other error code that is not included in SHE spec          |

## Note

Definition at line 417 of file eHSM\_If\_She\_Ip.c.

## 4.46.2.18 she\_secure\_boot()

```
ehsm_uint32_t she_secure_boot (
 ehsm_uint32_t size,
 const ehsm_uint8_t * data)
```

This function is called for the SOC secure boot in parallel.

## Parameters

|    |             |                                                                         |
|----|-------------|-------------------------------------------------------------------------|
| in | <i>size</i> | The SOC image size for secure boot. Should be (1024, 255 * 1024] bytes. |
| in | <i>data</i> | The SOC image data.                                                     |

## Returns

ehsm\_uint32\_t Secure boot result.

## Return values

|                           |                                                               |
|---------------------------|---------------------------------------------------------------|
| <i>ERC_NO_ERROR</i>       | Secure boot is successful.                                    |
| <i>ERC_GENERAL_ERROR</i>  | Secure boot failed since input is illegal.                    |
| <i>ERC_NO_SECURE_BOOT</i> | Secure boot verification failed or secure boot has been done. |
| <i>ERC_BUSY</i>           | The request can not be handled.                               |

## Note

This function can be called only once every power on. And can be called only parallel boot is enabled.

Definition at line 447 of file eHSM\_If\_She\_Ip.c.

#### 4.46.2.19 she\_verify\_mac()

```
ehsm_uint32_t she_verify_mac (
 ehsm_uint32_t key_id,
 const ehsm_uint8_t * msg,
 ehsm_uint32_t size,
 const ehsm_uint8_t * mac,
 ehsm_uint32_t mac_size,
 ehsm_uint32_t * vrf_status)
```

SHE mac verification. No padding is supported.

##### Parameters

|    |                 |                                                          |
|----|-----------------|----------------------------------------------------------|
| in | <i>key_id</i>   | SHE key index, only [4, 54] are valid.                   |
| in | <i>msg</i>      | Message.                                                 |
| in | <i>size</i>     | Size of message in bytes, should be a multiple of 16.    |
| in | <i>mac</i>      | The mac to be verified.                                  |
| in | <i>mac_size</i> | The mac size in byte to be verified. Belongs to [1, 16]. |
|    | <i>[in/out]</i> | vrf_status: 1 for mac matching, 0 for not matching.      |

##### Returns

0 for success, the result is written to vrf\_status. A negative value for error.

Definition at line 268 of file eHSM\_If\_She\_Ip.c.

## 4.47 eHSM\_If\_She\_Types\_Ip.h File Reference

```
#include "eHSM_Types_Ip.h"
```

### Classes

- struct [ehsm\\_she\\_key\\_st](#)

### Macros

- #define [EHSM\\_SHE\\_KEY\\_PROP\\_WR\\_PRT](#) (0x1 << 7)
- #define [EHSM\\_SHE\\_KEY\\_PROP\\_BOOT\\_PRT](#) (0x1 << 6)
- #define [EHSM\\_SHE\\_KEY\\_PROP\\_DEBUG\\_PRT](#) (0x1 << 5)
- #define [EHSM\\_SHE\\_KEY\\_PROP\\_KEY\\_USAGE](#) (0x1 << 4)
- #define [EHSM\\_SHE\\_KEY\\_PROP\\_WILDCARD](#) (0x1 << 3)
- #define [EHSM\\_SHE\\_KEY\\_MAC\\_VERIFY\\_ONLY](#) (0x1 << 2)
- #define [EHSM\\_SHE\\_M1\\_STD\\_SIZE](#) (16)
- #define [EHSM\\_SHE\\_M4\\_STD\\_SIZE](#) (32)

- #define `EHSM_SHE_KEY_SIZE` (sizeof(ehsm\_she\_key\_st))
- #define `EHSM_SHE_OTP_KEY_ATTR_SIZE` (8)
- #define `EHSM_SHE_KEY_MAX_SIZE` (16)
- #define `EHSM_SHE_UID_MAX_SIZE` (15)
- #define `EHSM_SHE_NVM_KEY_NUM` 50
- #define `ERC_NO_ERROR` 0x0U  
*No error has occurred and the command will be executed.*
- #define `ERC_GENERAL_ERROR` 0x1U  
*This error code is returned if an error not covered by the error codes above is detected inside SHE.*
- #define `ERC_SEQUENCE_ERROR` 0x2U  
*This error code is returned by SHE whenever the sequence of commands or subcommands is out of sequence, e.g. when a function is called while another function is still running.*
- #define `ERC_BUSY` 0x3U  
*This error code is returned whenever a function of SHE is called while another function is still processing, i.e., when  $SRE \leftrightarrow GBUSY = 1$ .*
- #define `ERC_KEY_NOT_AVAILABLE` 0x4U  
*This error code is returned if a key is locked due to failed boot measurement or an active debugger.*
- #define `ERC_KEY_INVALID` 0x5U  
*This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation.*
- #define `ERC_KEY_EMPTY` 0x6U  
*This error code is returned by SHE if the application attempts to use a key that has not been initialized yet.*
- #define `ERC_NO_SECURE_BOOT` 0x7U
- #define `ERC_KEY_WRITE_PROTECTED` 0x8U  
*This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to activate the debugger is started when a key is write-protected.*
- #define `ERC_KEY_UPDATE_ERROR` 0x9U  
*This error is returned when a key update did not succeed due to errors in verification of the messages.*
- #define `ERC_NO_DEBUGGING` 0xAU  
*The error code is returned if internal debugging is not possible because the authentication with the challenge response protocol did not succeed.*
- #define `ERC_RNG_SEED` 0xBU  
*The error code is returned by `CMD_RND` and `CMD_DEBUG` if the seed has not been initialized before.*
- #define `ERC_MEMORY_FAILURE` 0xCU  
*This error code can be returned if the underlying memory technology is able to detect physical errors, e.g. flipped bits etc., during memory read or write operations to notify the application.*

## Typedefs

- typedef enum `ehsm_she_status_type_e` `ehsm_she_status_type_e`

## Enumerations

- enum `ehsm_she_status_type_e` {  
`EHSM_SHE_STATUS_TYPE_BUSY` = (1 << 0), `EHSM_SHE_STATUS_TYPE_SECURE_BOOT` = (1 << 1), `EHSM_SHE_STATUS_TYPE_BOOT_INIT` = (1 << 2), `EHSM_SHE_STATUS_TYPE_BOOT_FINISHED` = (1 << 3),  
`EHSM_SHE_STATUS_TYPE_BOOT_OK` = (1 << 4), `EHSM_SHE_STATUS_TYPE_RND_INT` = (1 << 5), `EHSM_SHE_STATUS_TYPE_EXT_DEBUGGER` = (1 << 6), `EHSM_SHE_STATUS_TYPE_INT_DEBUGGER` = (1 << 7) }
- enum `ehsm_she_key_handle_e` { `EHSM_KEY_SHE_RAM_KEY` = 0, `EHSM_KEY_SHE_MASTER_ECU_KEY`, `EHSM_KEY_SHE_END` = `EHSM_SHE_NVM_KEY_NUM` + 1 }

## 4.47.1 Macro Definition Documentation

### 4.47.1.1 EHSM\_SHE\_KEY\_MAC\_VERIFY\_ONLY

```
#define EHSM_SHE_KEY_MAC_VERIFY_ONLY (0x1 << 2)
```

Definition at line 19 of file eHSM\_If\_She\_Types\_Ip.h.

### 4.47.1.2 EHSM\_SHE\_KEY\_MAX\_SIZE

```
#define EHSM_SHE_KEY_MAX_SIZE (16)
```

Definition at line 27 of file eHSM\_If\_She\_Types\_Ip.h.

### 4.47.1.3 EHSM\_SHE\_KEY\_PROP\_BOOT\_PRT

```
#define EHSM_SHE_KEY_PROP_BOOT_PRT (0x1 << 6)
```

Definition at line 15 of file eHSM\_If\_She\_Types\_Ip.h.

### 4.47.1.4 EHSM\_SHE\_KEY\_PROP\_DEBUG\_PRT

```
#define EHSM_SHE_KEY_PROP_DEBUG_PRT (0x1 << 5)
```

Definition at line 16 of file eHSM\_If\_She\_Types\_Ip.h.

### 4.47.1.5 EHSM\_SHE\_KEY\_PROP\_KEY\_USAGE

```
#define EHSM_SHE_KEY_PROP_KEY_USAGE (0x1 << 4)
```

Definition at line 17 of file eHSM\_If\_She\_Types\_Ip.h.

### 4.47.1.6 EHSM\_SHE\_KEY\_PROP\_WILDCARD

```
#define EHSM_SHE_KEY_PROP_WILDCARD (0x1 << 3)
```

Definition at line 18 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.7 EHSM\_SHE\_KEY\_PROP\_WR\_PRT

```
#define EHSM_SHE_KEY_PROP_WR_PRT (0x1 << 7)
```

Definition at line 14 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.8 EHSM\_SHE\_KEY\_SIZE

```
#define EHSM_SHE_KEY_SIZE (sizeof(ehsm_she_key_st))
```

Definition at line 24 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.9 EHSM\_SHE\_M1\_STD\_SIZE

```
#define EHSM_SHE_M1_STD_SIZE (16)
```

Definition at line 21 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.10 EHSM\_SHE\_M4\_STD\_SIZE

```
#define EHSM_SHE_M4_STD_SIZE (32)
```

Definition at line 22 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.11 EHSM\_SHE\_NVM\_KEY\_NUM

```
#define EHSM_SHE_NVM_KEY_NUM 50
```

Definition at line 30 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.12 EHSM\_SHE\_OTP\_KEY\_ATTR\_SIZE

```
#define EHSM_SHE_OTP_KEY_ATTR_SIZE (8)
```

Definition at line 25 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.13 EHSM\_SHE\_UID\_MAX\_SIZE

```
#define EHSM_SHE_UID_MAX_SIZE (15)
```

Definition at line 28 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.14 ERC\_BUSY

```
#define ERC_BUSY 0x3U
```

This error code is returned whenever a function of SHE is called while another function is still processing, i.e., when SREGBUSY = 1.

Definition at line 54 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.15 ERC\_GENERAL\_ERROR

```
#define ERC_GENERAL_ERROR 0x1U
```

This error code is returned if an error not covered by the error codes above is detected inside SHE.

Definition at line 40 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.16 ERC\_KEY\_EMPTY

```
#define ERC_KEY_EMPTY 0x6U
```

This error code is returned by SHE if the application attempts to use a key that has not been initialized yet.

Definition at line 70 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.17 ERC\_KEY\_INVALID

```
#define ERC_KEY_INVALID 0x5U
```

This error code is returned by SHE whenever a function is called to perform an operation with a key that is not allowed for the given operation.

Definition at line 65 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.18 ERC\_KEY\_NOT\_AVAILABLE

```
#define ERC_KEY_NOT_AVAILABLE 0x4U
```

This error code is returned if a key is locked due to failed boot measurement or an active debugger.

Definition at line 59 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.19 ERC\_KEY\_UPDATE\_ERROR

```
#define ERC_KEY_UPDATE_ERROR 0x9U
```

This error is returned when a key update did not succeed due to errors in verification of the messages.

Definition at line 83 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.20 ERC\_KEY\_WRITE\_PROTECTED

```
#define ERC_KEY_WRITE_PROTECTED 0x8U
```

This error is returned when a key update is attempted on a memory slot that has been write protected or when an attempt to active the debugger is started when a key is write-protected.

Definition at line 78 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.21 ERC\_MEMORY\_FAILURE

```
#define ERC_MEMORY_FAILURE 0xCU
```

This error code can be returned if the underlying memory technology is able to detect physical errors, e.g. flipped bits etc., during memory read or write operations to notify the application.

##### Note

This is not supported

Definition at line 104 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.22 ERC\_NO\_DEBUGGING

```
#define ERC_NO_DEBUGGING 0xAU
```

The error code is returned if internal debugging is not possible because the authentication with the challenge response protocol did not succeed.

Definition at line 89 of file eHSM\_If\_She\_Types\_Ip.h.



#### 4.47.1.23 ERC\_NO\_ERROR

```
#define ERC_NO_ERROR 0x0U
```

No error has occurred and the command will be executed.

Definition at line 35 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.24 ERC\_NO\_SECURE\_BOOT

```
#define ERC_NO_SECURE_BOOT 0x7U
```

Definition at line 72 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.25 ERC\_RNG\_SEED

```
#define ERC_RNG_SEED 0xBU
```

The error code is returned by CMD\_RND and CMD\_DEBUG if the seed has not been initialized before.

##### Note

This is not supported.

Definition at line 96 of file eHSM\_If\_She\_Types\_Ip.h.

#### 4.47.1.26 ERC\_SEQUENCE\_ERROR

```
#define ERC_SEQUENCE_ERROR 0x2U
```

This error code is returned by SHE whenever the sequence of commands or subcommands is out of sequence, e.g. when a function is called while another function is still running.

##### Note

This is not supported.

Definition at line 48 of file eHSM\_If\_She\_Types\_Ip.h.

### 4.47.2 Typedef Documentation

#### 4.47.2.1 ehsm\_she\_status\_type\_e

```
typedef enum ehsm_she_status_type_ ehsm_she_status_type_e
```

### 4.47.3 Enumeration Type Documentation

#### 4.47.3.1 ehsm\_she\_key\_handle\_e

```
enum ehsm_she_key_handle_e
```

## Enumerator

|                             |  |
|-----------------------------|--|
| EHSM_KEY_SHE_RAM_KEY        |  |
| EHSM_KEY_SHE_MASTER_ECU_KEY |  |
| EHSM_KEY_SHE_END            |  |

Definition at line 147 of file eHSM\_If\_She\_Types\_Ip.h.

## 4.47.3.2 ehsm\_she\_status\_type\_

```
enum ehsm_she_status_type_
```

## Enumerator

|                                    |  |
|------------------------------------|--|
| EHSM_SHE_STATUS_TYPE_BUSY          |  |
| EHSM_SHE_STATUS_TYPE_SECURE_BOOT   |  |
| EHSM_SHE_STATUS_TYPE_BOOT_INIT     |  |
| EHSM_SHE_STATUS_TYPE_BOOT_FINISHED |  |
| EHSM_SHE_STATUS_TYPE_BOOT_OK       |  |
| EHSM_SHE_STATUS_TYPE_RND_INT       |  |
| EHSM_SHE_STATUS_TYPE_EXT_DEBUGGER  |  |
| EHSM_SHE_STATUS_TYPE_INT_DEBUGGER  |  |

Definition at line 118 of file eHSM\_If\_She\_Types\_Ip.h.

## 4.48 eHSM\_IntCfg\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
```

## Macros

- #define [CONFIG\\_EHSM\\_ARCH\\_MULTI\\_CHANNEL](#)  
*configuration for multiple channels support*
- #define [CONFIG\\_EHSM\\_ARCH\\_OS\\_NONE](#)  
*configuration for operation system support*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_REQ\\_SKE\\_MAX\\_SIZE](#) 16U  
*configuration for RT-Thread support*
- #define [CONFIG\\_EHSM\\_ARCH\\_V\\_REQ\\_HASH\\_MAX\\_SIZE](#) 16U  
*The size of hash requirements in cache.*
- #define [CONFIG\\_EHSM\\_COUNTER\\_AUTO\\_INCREASE](#)  
*configuration for secure counter increased automatically support*
- #define [CONFIG\\_EHSM\\_CRYPT0\\_AEAD](#)  
*configuration for cryptographic primitives support*
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_AES](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_CTRDRBG](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_DH](#)

- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_ECC](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SECP256R1](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SECP384R1](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_ED25519](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_MD5](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_PBKDF2](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_RSA](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_RSA\\_1024](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_RSA\\_1024\\_CRT](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_RSA\\_2048](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_RSA\\_2048\\_CRT](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_RSASSA\\_PSS](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA1](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA2](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA224](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA256](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA384](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA512](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA512\\_224](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SHA512\\_256](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SM2](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SM3](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_SM4](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOFAM\\_X963](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_CBC](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_CBC\\_MAC](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_CFB](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_CMAC](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_CTR](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_ECB](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_GMAC](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_ALGOMODE\\_OFB](#)
- #define [CONFIG\\_EHSM\\_CRYPT0\\_RSA\\_CRT\\_MODE](#)
- configuration for certificate support*
  - #define [CONFIG\\_EHSM\\_CRYPT0\\_V\\_CERT\\_MAX\\_PUB\\_K\\_SIZE](#) 256U
  - The max public key size of x509 certificate.*
  - #define [CONFIG\\_EHSM\\_CRYPT0\\_V\\_CERT\\_MAX\\_SIZE](#) 1024U
  - The max size of x509 certificate data.*
  - #define [CONFIG\\_EHSM\\_CRYPT0\\_V\\_GCM\\_MAX\\_AAD\\_SIZE](#) 128U
  - The max size of associated data in bytes.*
  - #define [CONFIG\\_EHSM\\_CRYPT0\\_V\\_HMAC\\_MAX\\_KSIZE](#) 256U
  - The max key size in bytes of HMAC.*
- #define [CONFIG\\_EHSM\\_V\\_LOG\\_ERR](#) 0
- configuration for EVB board support*
  - #define [CONFIG\\_EHSM\\_V\\_LOG\\_WARN](#) 1
  - #define [CONFIG\\_EHSM\\_V\\_LOG\\_DEBUG](#) 2
  - #define [CONFIG\\_EHSM\\_V\\_LOG\\_INFO](#) 3
  - #define [CONFIG\\_EHSM\\_V\\_LOG\\_LEVEL](#) [CONFIG\\_EHSM\\_V\\_LOG\\_DEBUG](#)
  - #define [CONFIG\\_EHSM\\_FIRMWARE\\_UPGRADE](#)
  - configuration of debug load key for test*
  - #define [CONFIG\\_EHSM\\_HW\\_AHB\\_BYTE](#)
  - configuration for AHB byte operation support*
  - #define [CONFIG\\_EHSM\\_HW\\_BRANCH\\_2\\_0\\_0](#)
  - configuration for version 1*
  - #define [CONFIG\\_EHSM\\_HW\\_COUNTER](#)

*configuration for version 3*

- #define `CONFIG_EHSM_HW_FLASH`

*configuration for flash type*

- #define `CONFIG_EHSM_HW_V_FLASH_NONE` 0
- #define `CONFIG_EHSM_HW_V_FLASH_CUSTOMER` 1
- #define `CONFIG_EHSM_HW_V_FLASH_SIMULATE` 2
- #define `CONFIG_EHSM_HW_V_FLASH_TYPE` `CONFIG_EHSM_HW_V_FLASH_CUSTOMER`
- #define `CONFIG_EHSM_HW_V_FLASH_BASE_ADDR` (0x01120000)
- #define `CONFIG_EHSM_HW_V_FLASH_SIZE` (0x20000)
- #define `CONFIG_EHSM_HW_V_CODE_MAX_SIZE` (0x1E000)
- #define `CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE` (0x800)
- #define `CONFIG_EHSM_HW_V_FLASH_DATA_ADDR` (`CONFIG_EHSM_HW_V_FLASH_BASE_ADDR` + `CONFIG_EHSM_HW_V_CODE_MAX_SIZE`)
- #define `CONFIG_EHSM_HW_V_FLASH_DATA_SIZE` (`CONFIG_EHSM_HW_V_FLASH_SIZE` - `CONFIG_EHSM_HW_V_CODE_MAX_SIZE`)
- #define `CONFIG_EHSM_HW_V_FLASH_KEY_ADDR` (`CONFIG_EHSM_HW_V_FLASH_DATA_ADDR`)
- #define `CONFIG_EHSM_HW_V_FLASH_KEY_SIZE` (0x1000)
- #define `CONFIG_EHSM_HW_V_FLASH_LOG_ADDR` (`CONFIG_EHSM_HW_V_FLASH_KEY_ADDR` + `CONFIG_EHSM_HW_V_FLASH_KEY_SIZE`)
- #define `CONFIG_EHSM_HW_V_FLASH_LOG_SIZE` (`CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE`)
- #define `CONFIG_EHSM_HW_V_FLASH_SYS_ADDR` (`CONFIG_EHSM_HW_V_FLASH_LOG_ADDR` + `CONFIG_EHSM_HW_V_FLASH_LOG_SIZE`)
- #define `CONFIG_EHSM_HW_V_FLASH_SYS_SIZE` (`CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE`)
- #define `CONFIG_EHSM_HW_V_FLASH_FREE_ECC_IDX1` (0xFBFFBFFFBFFBFFF)
- #define `CONFIG_EHSM_HW_V_FLASH_FREE_ECC_IDX2` (0xF3BFBFEFF3BFBFEF)
- #define `CONFIG_EHSM_HW_V_FLASH_WRITE_MIN_BYTES` (8)
- #define `CONFIG_EHSM_HW_V_FLASH_ERASE_CELL_VALUE` (0xFF)
- #define `CONFIG_EHSM_HW_FLASH_ECC`

*configuration for flash ECC support*

- #define `CONFIG_EHSM_HW_GUOMI_LEVEL1`

*configuration for GuoMi level1*

- #define `CONFIG_EHSM_HW_HASH`

*configuration for GuoMi level2*

- #define `CONFIG_EHSM_HW_HASH_DMA`

*configuration for HASH DMA mode support*

- #define `CONFIG_EHSM_HW_HASH_LP`

*configuration for HASH HP support*

- #define `CONFIG_EHSM_HW_INSTALL_K_KEY`

*configuration for HMAC secure port support*

- #define `CONFIG_EHSM_HW_LOW_POWER`

*configuration for hardware KMU support*

- #define `CONFIG_EHSM_HW_LIFE_CYCLE_TEST_MODE` 0xFFFFFFFF

*configuration life cycle exist valid field*

- #define `CONFIG_EHSM_HW_LIFE_CYCLE_DEVELOP_MODE` 0xBD7E7BEB
- #define `CONFIG_EHSM_HW_LIFE_CYCLE_MANUFACTURE_MODE` 0xB93E5BE9
- #define `CONFIG_EHSM_HW_LIFE_CYCLE_USER_MODE` 0xA83E1369
- #define `CONFIG_EHSM_HW_LIFE_CYCLE_DEBUG_MODE` 0x283A0321
- #define `CONFIG_EHSM_HW_LIFE_CYCLE_DESTROY_MODE` 0x00000000
- #define `CONFIG_EHSM_HW_OTP`

*configuration for REG read write support*

- #define `CONFIG_EHSM_HW_OTP_MAP`

*configuration for OTP ECC support*

- #define `CONFIG_EHSM_HW_V_NONE_OTP` 0
- #define `CONFIG_EHSM_HW_V_CUSTOMER_OTP` 1
- #define `CONFIG_EHSM_HW_V_SIMULATE_OTP` 2

- #define `CONFIG_EHSM_HW_V_OTP_TYPE` `CONFIG_EHSM_HW_V_CUSTOMER_OTP`
- #define `CONFIG_EHSM_HW_V_OTP_BASE_ADDR` (0x01540000UL)
- #define `CONFIG_EHSM_HW_V_OTP_SIZE` (0x800)
- #define `CONFIG_EHSM_HW_V_OTP_PAGE_SIZE` (0x800)
- #define `CONFIG_EHSM_HW_V_OTP_KEY_NUM` (32)
- #define `CONFIG_EHSM_HW_V_OTP_WRITE_MIN_BYTES` (4)
- #define `CONFIG_EHSM_HW_V_OTP_K_ATTR_LENGTH` (64U)
- #define `CONFIG_EHSM_HW_V_OTP_VERSION_LENGTH` (0U)
- #define `CONFIG_EHSM_HW_PKE`  
*configuration for PKE support*
- #define `CONFIG_EHSM_HW_PKE_LP`  
*configuration for PKE HP support*
- #define `CONFIG_EHSM_HW_SKE_DMA`  
*configuration for PKE LP secure support*
- #define `CONFIG_EHSM_HW_SKE_LP`  
*configuration for SKE HP support*
- #define `CONFIG_EHSM_HW_SKE_SECURE_PORT`  
*configuration for SKE LP secure support*
- #define `CONFIG_EHSM_HW_TRNG`  
*configuration for TRNG support*
- #define `CONFIG_EHSM_HW_UTC_TIME`  
*configuration for UTC TIME support*
- #define `CONFIG_EHSM_HW_V_WORK_FREQ` 10000000U  
*configuration for HOST address size 64 bits*
- #define `CONFIG_EHSM_JTAG_DEBUG_AUTH`  
*configuration for jtag debug authentication support*
- #define `CONFIG_EHSM_KMGR_CHECK_OTP_K_ATTR`  
*configuration for flash key backup support*
- #define `CONFIG_EHSM_KMGR_PLAIN_K_IMPORT`  
*configuration for key signature support*
- #define `CONFIG_EHSM_KMGR_V_MAX_AUTH_CODE_SIZE` (32U)  
*Max authentication code size of EVITA key.*
- #define `CONFIG_EHSM_KMGR_V_OTP_EXT_K_OFF` (0x400)
- #define `CONFIG_EHSM_KMGR_V_OTP_EXT_K_SIZE` (0x400)
- #define `CONFIG_EHSM_KMGR_V_FLASH_K_START_OFFSET` (0x00)  
*configuration for flash key storage address*
- #define `CONFIG_EHSM_KMGR_V_FLASH_OTP_K_START_OFFSET` (`CONFIG_EHSM_KMGR_V_OTP_EXT_K_OFF`)
- #define `CONFIG_EHSM_KMGR_V_SHE_K_NUM` (40)  
*configuration for flash key number and size*
- #define `CONFIG_EHSM_KMGR_V_SHE_K_SLOT_SIZE` (48)
- #define `CONFIG_EHSM_KMGR_V_SHE_K_AREA_SIZE` (0x800)
- #define `CONFIG_EHSM_KMGR_V_SHE_K_START_ADDR` (`CONFIG_EHSM_KMGR_V_FLASH_K_START_OFFSET`)
- #define `CONFIG_EHSM_KMGR_V_SHE_K_END_ADDR`
- #define `CONFIG_EHSM_KMGR_V_SYM_K_NUM` (10)  
*configuration for EVITA key number and size*
- #define `CONFIG_EHSM_KMGR_V_SYM_K_SLOT_SIZE` (200)
- #define `CONFIG_EHSM_KMGR_V_SYM_K_AREA_SIZE` (0x800)
- #define `CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR` (`CONFIG_EHSM_KMGR_V_SHE_K_END_ADDR`)
- #define `CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR`
- #define `CONFIG_EHSM_KMGR_V_ECC_K_NUM` (7)
- #define `CONFIG_EHSM_KMGR_V_ECC_K_SLOT_SIZE` (256)
- #define `CONFIG_EHSM_KMGR_V_ECC_K_AREA_SIZE` (0x800)
- #define `CONFIG_EHSM_KMGR_V_ECC_K_START_ADDR` (`CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR`)

- #define `CONFIG_EHSM_KMGR_V_ECC_K_END_ADDR` (`CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR`)
- #define `CONFIG_EHSM_KMGR_V_RSA_K_NUM` (3)
- #define `CONFIG_EHSM_KMGR_V_RSA_K_SLOT_SIZE` (600)
- #define `CONFIG_EHSM_KMGR_V_RSA_K_AREA_SIZE` (0x800)
- #define `CONFIG_EHSM_KMGR_V_RSA_K_START_ADDR` (`CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR`)
- #define `CONFIG_EHSM_KMGR_V_RSA_K_END_ADDR` (`CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR`)
- #define `CONFIG_EHSM_KMGR_V_SM9_K_NUM` (0)
- #define `CONFIG_EHSM_KMGR_V_SM9_K_SLOT_SIZE` (200)
- #define `CONFIG_EHSM_KMGR_V_SM9_K_AREA_SIZE` (0)
- #define `CONFIG_EHSM_KMGR_V_SM9_K_START_ADDR` (`CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR`)
- #define `CONFIG_EHSM_KMGR_V_SM9_K_END_ADDR` (`CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR`)
- #define `CONFIG_EHSM_KMGR_V_RAM_K_MEM_SIZE` (2560)
  - configuration for EVITA ram key number and size*
- #define `CONFIG_EHSM_KMGR_V_SYM_RAM_K_NUM` (10)
- #define `CONFIG_EHSM_KMGR_V_ECC_RAM_K_NUM` (5)
- #define `CONFIG_EHSM_KMGR_V_RSA_RAM_K_NUM` (5)
- #define `CONFIG_EHSM_LOG`
  - configuration for eHSM log support*
- #define `CONFIG_EHSM_SOC_UPGRADE_AND_VERIFY`
  - configuration for OTP key crc byte reverse support*
- #define `CONFIG_EHSM_USER_AUTH_KYE_IN_KMU`
  - configuration for system data backup support*
- #define `CONFIG_EHSM_SHE_SOC_BOOT`
  - configuration for reverse otp data, the otp default bit is 0, it can be change to bit 1*
- #define `CONFIG_EHSM_SKE_CORE_NUM` 1
  - configuration for multicore and core number.*
- #define `CONFIG_EHSM_PKE_CORE_NUM` 1
- #define `CONFIG_EHSM_HASH_CORE_NUM` 1
- #define `CONFIG_EHSM_TRNG_CORE_NUM` 1
- #define `CONFIG_EHSM_ARCH_MAIN_HOOK`
  - configuration for key install used new mailbox/API format*
- #define `CONFIG_EHSM_ARCH_SHARE_MEM`
  - configuration for share memory which used for flash page buffer*
- #define `CONFIG_EHSM_UNIT_TEST`
  - config for unittest*

## 4.48.1 Macro Definition Documentation

### 4.48.1.1 CONFIG\_EHSM\_ARCH\_MAIN\_HOOK

```
#define CONFIG_EHSM_ARCH_MAIN_HOOK
```

configuration for key install used new mailbox/API format

configuration for TCM. configuration ehsm soft reset enable configuration for feature of main hook

Definition at line 641 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.2 CONFIG\_EHSM\_ARCH\_MULTI\_CHANNEL

```
#define CONFIG_EHSM_ARCH_MULTI_CHANNEL
```

configuration for multiple channels support

Definition at line 23 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.3 CONFIG\_EHSM\_ARCH\_OS\_NONE

```
#define CONFIG_EHSM_ARCH_OS_NONE
```

configuration for operation system support

Definition at line 28 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.4 CONFIG\_EHSM\_ARCH\_SHARE\_MEM

```
#define CONFIG_EHSM_ARCH_SHARE_MEM
```

configuration for share memory whitch used for flash page buffer

Definition at line 646 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.5 CONFIG\_EHSM\_ARCH\_V\_REQ\_HASH\_MAX\_SIZE

```
#define CONFIG_EHSM_ARCH_V_REQ_HASH_MAX_SIZE 16U
```

The size of hash requirements in cache.

Definition at line 43 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.6 CONFIG\_EHSM\_ARCH\_V\_REQ\_SKE\_MAX\_SIZE

```
#define CONFIG_EHSM_ARCH_V_REQ_SKE_MAX_SIZE 16U
```

configuration for RT-Thread support

The size of ske requirements in cache.

Definition at line 38 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.7 CONFIG\_EHSM\_COUNTER\_AUTO\_INCREASE

```
#define CONFIG_EHSM_COUNTER_AUTO_INCREASE
```

configuration for secure counter increased automatically support

Definition at line 48 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.8 CONFIG\_EHSM\_CRYPT0\_AEAD

```
#define CONFIG_EHSM_CRYPT0_AEAD
```

configuration for cryptographic primitives support

Definition at line 53 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.9 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_AES

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_AES
```

Definition at line 55 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.10 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_CTRDRBG

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_CTRDRBG
```

Definition at line 56 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.11 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_DH

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_DH
```

Definition at line 58 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.12 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_ECC

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_ECC
```

Definition at line 59 of file eHSM\_IntCfg\_Ip.h.



**4.48.1.13 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_ED25519**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_ED25519
```

Definition at line 74 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.14 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_MD5**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_MD5
```

Definition at line 75 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.15 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_PBKDF2**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_PBKDF2
```

Definition at line 76 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.16 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_RSA
```

Definition at line 77 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.17 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_1024**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_RSA_1024
```

Definition at line 79 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.18 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_1024\_CRT**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_RSA_1024_CRT
```

Definition at line 80 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.19 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_2048**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_RSA_2048
```

Definition at line 81 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.20 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_2048\_CRT**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_RSA_2048_CRT
```

Definition at line 82 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.21 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SECP256R1**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SECP256R1
```

Definition at line 70 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.22 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SECP384R1**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SECP384R1
```

Definition at line 71 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.23 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA1**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA1
```

Definition at line 89 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.24 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA2**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA2
```

Definition at line 90 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.25 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA224**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA224
```

Definition at line 91 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.26 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA256**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA256
```

Definition at line 92 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.27 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA384**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA384
```

Definition at line 93 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.28 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA512**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA512
```

Definition at line 94 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.29 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA512\_224**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA512_224
```

Definition at line 95 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.30 CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA512\_256**

```
#define CONFIG_EHSM_CRYPT0_ALGOFAM_SHA512_256
```

Definition at line 96 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.31 CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SM2**

```
#define CONFIG_EHSM_CRYPTO_ALGOFAM_SM2
```

Definition at line 98 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.32 CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SM3**

```
#define CONFIG_EHSM_CRYPTO_ALGOFAM_SM3
```

Definition at line 99 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.33 CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SM4**

```
#define CONFIG_EHSM_CRYPTO_ALGOFAM_SM4
```

Definition at line 100 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.34 CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_X963**

```
#define CONFIG_EHSM_CRYPTO_ALGOFAM_X963
```

Definition at line 103 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.35 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CBC**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_CBC
```

Definition at line 104 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.36 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CBC\_MAC**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_CBC_MAC
```

Definition at line 105 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.37 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CFB**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_CFB
```

Definition at line 107 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.38 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CMAC**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_CMAC
```

Definition at line 108 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.39 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CTR**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_CTR
```

Definition at line 109 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.40 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_ECB**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_ECB
```

Definition at line 110 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.41 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_GMAC**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_GMAC
```

Definition at line 112 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.42 CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_OFB**

```
#define CONFIG_EHSM_CRYPTO_ALGOMODE_OFB
```

Definition at line 113 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.43 CONFIG\_EHSM\_CRYPT0\_ALGOMODE\_RSASSA\_PSS**

```
#define CONFIG_EHSM_CRYPT0_ALGOMODE_RSASSA_PSS
```

Definition at line 87 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.44 CONFIG\_EHSM\_CRYPT0\_RSA\_CERT\_MODE**

```
#define CONFIG_EHSM_CRYPT0_RSA_CERT_MODE
```

configuration for certificate support

configuration for fast cmac support configuration for RSA CRT mode support

Definition at line 129 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.45 CONFIG\_EHSM\_CRYPT0\_V\_CERT\_MAX\_PUB\_K\_SIZE**

```
#define CONFIG_EHSM_CRYPT0_V_CERT_MAX_PUB_K_SIZE 256U
```

The max public key size of x509 certificate.

Definition at line 134 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.46 CONFIG\_EHSM\_CRYPT0\_V\_CERT\_MAX\_SIZE**

```
#define CONFIG_EHSM_CRYPT0_V_CERT_MAX_SIZE 1024U
```

The max size of x509 certificate data.

Definition at line 139 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.47 CONFIG\_EHSM\_CRYPT0\_V\_GCM\_MAX\_AAD\_SIZE**

```
#define CONFIG_EHSM_CRYPT0_V_GCM_MAX_AAD_SIZE 128U
```

The max size of associated data in bytes.

Definition at line 144 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.48 CONFIG\_EHSM\_CRYPT0\_V\_HMAC\_MAX\_KSIZE**

```
#define CONFIG_EHSM_CRYPT0_V_HMAC_MAX_KSIZE 256U
```

The max key size in bytes of HMAC.

Definition at line 149 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.49 CONFIG\_EHSM\_FIRMWARE\_UPGRADE**

```
#define CONFIG_EHSM_FIRMWARE_UPGRADE
```

configuration of debug load key for test

configuration for firmware upgrade support

Definition at line 174 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.50 CONFIG\_EHSM\_HASH\_CORE\_NUM**

```
#define CONFIG_EHSM_HASH_CORE_NUM 1
```

Definition at line 619 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.51 CONFIG\_EHSM\_HW\_AHB\_BYTE**

```
#define CONFIG_EHSM_HW_AHB_BYTE
```

configuration for AHB byte operation support

Definition at line 179 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.52 CONFIG\_EHSM\_HW\_BRANCH\_2\_0\_0**

```
#define CONFIG_EHSM_HW_BRANCH_2_0_0
```

configuration for version 1

configuration for version 2

Definition at line 189 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.53 CONFIG\_EHSM\_HW\_COUNTER**

```
#define CONFIG_EHSM_HW_COUNTER
```

configuration for version 3

configuration custom HW configuration for hardware counter support

Definition at line 204 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.54 CONFIG\_EHSM\_HW\_FLASH**

```
#define CONFIG_EHSM_HW_FLASH
```

configuration for flash type

Definition at line 209 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.55 CONFIG\_EHSM\_HW\_FLASH\_ECC**

```
#define CONFIG_EHSM_HW_FLASH_ECC
```

configuration for flash ECC support

Definition at line 243 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.56 CONFIG\_EHSM\_HW\_GUOMI\_LEVEL1**

```
#define CONFIG_EHSM_HW_GUOMI_LEVEL1
```

configuration for GuoMi level1

Definition at line 248 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.57 CONFIG\_EHSM\_HW\_HASH**

```
#define CONFIG_EHSM_HW_HASH
```

configuration for GuoMi level2

configuration for HASH support

Definition at line 258 of file eHSM\_IntCfg\_Ip.h.



**4.48.1.58 CONFIG\_EHSM\_HW\_HASH\_DMA**

```
#define CONFIG_EHSM_HW_HASH_DMA
```

configuration for HASH DMA mode support

Definition at line 263 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.59 CONFIG\_EHSM\_HW\_HASH\_LP**

```
#define CONFIG_EHSM_HW_HASH_LP
```

configuration for HASH HP support

configuration for HASH LP support

Definition at line 273 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.60 CONFIG\_EHSM\_HW\_INSTALL\_K\_KEK**

```
#define CONFIG_EHSM_HW_INSTALL_K_KEK
```

configuration for HMAC secure port support

configuration for protection in encrypt key install

Definition at line 283 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.61 CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DEBUG\_MODE**

```
#define CONFIG_EHSM_HW_LIFE_CYCLE_DEBUG_MODE 0x283A0321
```

Definition at line 307 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.62 CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DESTROY\_MODE**

```
#define CONFIG_EHSM_HW_LIFE_CYCLE_DESTROY_MODE 0x00000000
```

Definition at line 308 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.63 CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DEVELOP\_MODE

```
#define CONFIG_EHSM_HW_LIFE_CYCLE_DEVELOP_MODE 0xBD7E7BEB
```

Definition at line 304 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.64 CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_MANUFACTURE\_MODE

```
#define CONFIG_EHSM_HW_LIFE_CYCLE_MANUFACTURE_MODE 0xB93E5BE9
```

Definition at line 305 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.65 CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_TEST\_MODE

```
#define CONFIG_EHSM_HW_LIFE_CYCLE_TEST_MODE 0xFFFFFFFF
```

configuration life cycle exist valid field

configuration life cycle mode value

Definition at line 303 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.66 CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_USER\_MODE

```
#define CONFIG_EHSM_HW_LIFE_CYCLE_USER_MODE 0xA83E1369
```

Definition at line 306 of file eHSM\_IntCfg\_Ip.h.

#### 4.48.1.67 CONFIG\_EHSM\_HW\_LOW\_POWER

```
#define CONFIG_EHSM_HW_LOW_POWER
```

configuration for hardware KMU support

configuration for low power support

Definition at line 293 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.68 CONFIG\_EHSM\_HW\_OTP**

```
#define CONFIG_EHSM_HW_OTP
```

configuration for REG read write support

configuration for OTP support

Definition at line 318 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.69 CONFIG\_EHSM\_HW\_OTP\_MAP**

```
#define CONFIG_EHSM_HW_OTP_MAP
```

configuration for OTP ECC support

configuration for OTP version2 support

Definition at line 328 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.70 CONFIG\_EHSM\_HW\_PKE**

```
#define CONFIG_EHSM_HW_PKE
```

configuration for PKE support

Definition at line 376 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.71 CONFIG\_EHSM\_HW\_PKE\_LP**

```
#define CONFIG_EHSM_HW_PKE_LP
```

configuration for PKE HP support

configuration for PKE LP support

Definition at line 386 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.72 CONFIG\_EHSM\_HW\_SKE\_DMA**

```
#define CONFIG_EHSM_HW_SKE_DMA
```

configuration for PKE LP secure support

configuration for PKE secure support configuration for PKE UHP support configuration for PKE UHP ECC support configuration for hardware initialize ram support configuration for SKE DMA mode support

Definition at line 416 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.73 CONFIG\_EHSM\_HW\_SKE\_LP**

```
#define CONFIG_EHSM_HW_SKE_LP
```

configuration for SKE HP support

configuration for SKE LP support

Definition at line 426 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.74 CONFIG\_EHSM\_HW\_SKE\_SECURE\_PORT**

```
#define CONFIG_EHSM_HW_SKE_SECURE_PORT
```

configuration for SKE LP secure support

configuration for SKE secure port support

Definition at line 436 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.75 CONFIG\_EHSM\_HW\_TRNG**

```
#define CONFIG_EHSM_HW_TRNG
```

configuration for TRNG support

Definition at line 441 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.76 CONFIG\_EHSM\_HW\_UTC\_TIME**

```
#define CONFIG_EHSM_HW_UTC_TIME
```

configuration for UTC TIME support

Definition at line 446 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.77 CONFIG\_EHSM\_HW\_V\_CODE\_MAX\_SIZE**

```
#define CONFIG_EHSM_HW_V_CODE_MAX_SIZE (0x1E000)
```

Definition at line 222 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.78 CONFIG\_EHSM\_HW\_V\_CUSTOMER\_OTP**

```
#define CONFIG_EHSM_HW_V_CUSTOMER_OTP 1
```

Definition at line 331 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.79 CONFIG\_EHSM\_HW\_V\_FLASH\_BASE\_ADDR**

```
#define CONFIG_EHSM_HW_V_FLASH_BASE_ADDR (0x01120000)
```

Definition at line 216 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.80 CONFIG\_EHSM\_HW\_V\_FLASH\_CUSTOMER**

```
#define CONFIG_EHSM_HW_V_FLASH_CUSTOMER 1
```

Definition at line 211 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.81 CONFIG\_EHSM\_HW\_V\_FLASH\_DATA\_ADDR**

```
#define CONFIG_EHSM_HW_V_FLASH_DATA_ADDR (CONFIG_EHSM_HW_V_FLASH_BASE_ADDR + CONFIG_EHSM_HW_V_CODE↵
_MAX_SIZE)
```

Definition at line 225 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.82 CONFIG\_EHSM\_HW\_V\_FLASH\_DATA\_SIZE**

```
#define CONFIG_EHSM_HW_V_FLASH_DATA_SIZE (CONFIG_EHSM_HW_V_FLASH_SIZE - CONFIG_EHSM_HW_V_CODE_MAX_↵
SIZE)
```

Definition at line 226 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.83 CONFIG\_EHSM\_HW\_V\_FLASH\_ERASE\_CELL\_VALUE**

```
#define CONFIG_EHSM_HW_V_FLASH_ERASE_CELL_VALUE (0xFF)
```

Definition at line 238 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.84 CONFIG\_EHSM\_HW\_V\_FLASH\_FREE\_ECC\_IDX1**

```
#define CONFIG_EHSM_HW_V_FLASH_FREE_ECC_IDX1 (0xFBFFBFFFBFFBFF)
```

Definition at line 235 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.85 CONFIG\_EHSM\_HW\_V\_FLASH\_FREE\_ECC\_IDX2**

```
#define CONFIG_EHSM_HW_V_FLASH_FREE_ECC_IDX2 (0xF3BFBBEFF3BFBBEF)
```

Definition at line 236 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.86 CONFIG\_EHSM\_HW\_V\_FLASH\_KEY\_ADDR**

```
#define CONFIG_EHSM_HW_V_FLASH_KEY_ADDR (CONFIG_EHSM_HW_V_FLASH_DATA_ADDR)
```

Definition at line 228 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.87 CONFIG\_EHSM\_HW\_V\_FLASH\_KEY\_SIZE**

```
#define CONFIG_EHSM_HW_V_FLASH_KEY_SIZE (0x1000)
```

Definition at line 229 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.88 CONFIG\_EHSM\_HW\_V\_FLASH\_LOG\_ADDR**

```
#define CONFIG_EHSM_HW_V_FLASH_LOG_ADDR (CONFIG_EHSM_HW_V_FLASH_KEY_ADDR + CONFIG_EHSM_HW_V_FLASH_↔
KEY_SIZE)
```

Definition at line 230 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.89 CONFIG\_EHSM\_HW\_V\_FLASH\_LOG\_SIZE**

```
#define CONFIG_EHSM_HW_V_FLASH_LOG_SIZE (CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE)
```

Definition at line 231 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.90 CONFIG\_EHSM\_HW\_V\_FLASH\_NONE**

```
#define CONFIG_EHSM_HW_V_FLASH_NONE 0
```

Definition at line 210 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.91 CONFIG\_EHSM\_HW\_V\_FLASH\_PAGE\_SIZE**

```
#define CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE (0x800)
```

Definition at line 223 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.92 CONFIG\_EHSM\_HW\_V\_FLASH\_SIMULATE**

```
#define CONFIG_EHSM_HW_V_FLASH_SIMULATE 2
```

Definition at line 212 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.93 CONFIG\_EHSM\_HW\_V\_FLASH\_SIZE**

```
#define CONFIG_EHSM_HW_V_FLASH_SIZE (0x20000)
```

Definition at line 221 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.94 CONFIG\_EHSM\_HW\_V\_FLASH\_SYS\_ADDR**

```
#define CONFIG_EHSM_HW_V_FLASH_SYS_ADDR (CONFIG_EHSM_HW_V_FLASH_LOG_ADDR + CONFIG_EHSM_HW_V_FLASH_↔
LOG_SIZE)
```

Definition at line 232 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.95 CONFIG\_EHSM\_HW\_V\_FLASH\_SYS\_SIZE**

```
#define CONFIG_EHSM_HW_V_FLASH_SYS_SIZE (CONFIG_EHSM_HW_V_FLASH_PAGE_SIZE)
```

Definition at line 233 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.96 CONFIG\_EHSM\_HW\_V\_FLASH\_TYPE**

```
#define CONFIG_EHSM_HW_V_FLASH_TYPE CONFIG_EHSM_HW_V_FLASH_CUSTOMER
```

Definition at line 213 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.97 CONFIG\_EHSM\_HW\_V\_FLASH\_WRITE\_MIN\_BYTES**

```
#define CONFIG_EHSM_HW_V_FLASH_WRITE_MIN_BYTES (8)
```

Definition at line 237 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.98 CONFIG\_EHSM\_HW\_V\_NONE\_OTP**

```
#define CONFIG_EHSM_HW_V_NONE_OTP 0
```

Definition at line 330 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.99 CONFIG\_EHSM\_HW\_V\_OTP\_BASE\_ADDR**

```
#define CONFIG_EHSM_HW_V_OTP_BASE_ADDR (0x01540000UL)
```

Definition at line 336 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.100 CONFIG\_EHSM\_HW\_V\_OTP\_K\_ATTR\_LENGTH**

```
#define CONFIG_EHSM_HW_V_OTP_K_ATTR_LENGTH (64U)
```

Definition at line 370 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.101 CONFIG\_EHSM\_HW\_V\_OTP\_KEY\_NUM**

```
#define CONFIG_EHSM_HW_V_OTP_KEY_NUM (32)
```

Definition at line 342 of file eHSM\_IntCfg\_Ip.h.



**4.48.1.102 CONFIG\_EHSM\_HW\_V\_OTP\_PAGE\_SIZE**

```
#define CONFIG_EHSM_HW_V_OTP_PAGE_SIZE (0x800)
```

Definition at line 341 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.103 CONFIG\_EHSM\_HW\_V\_OTP\_SIZE**

```
#define CONFIG_EHSM_HW_V_OTP_SIZE (0x800)
```

Definition at line 340 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.104 CONFIG\_EHSM\_HW\_V\_OTP\_TYPE**

```
#define CONFIG_EHSM_HW_V_OTP_TYPE CONFIG_EHSM_HW_V_CUSTOMER_OTP
```

Definition at line 333 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.105 CONFIG\_EHSM\_HW\_V\_OTP\_VERSION\_LENGTH**

```
#define CONFIG_EHSM_HW_V_OTP_VERSION_LENGTH (0U)
```

Definition at line 371 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.106 CONFIG\_EHSM\_HW\_V\_OTP\_WRITE\_MIN\_BYTES**

```
#define CONFIG_EHSM_HW_V_OTP_WRITE_MIN_BYTES (4)
```

Definition at line 343 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.107 CONFIG\_EHSM\_HW\_V\_SIMULATE\_OTP**

```
#define CONFIG_EHSM_HW_V_SIMULATE_OTP 2
```

Definition at line 332 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.108 CONFIG\_EHSM\_HW\_V\_WORK\_FREQ**

```
#define CONFIG_EHSM_HW_V_WORK_FREQ 10000000U
```

configuration for HOST address size 64 bits

configuration work frequency of eHSM

Definition at line 456 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.109 CONFIG\_EHSM\_JTAG\_DEBUG\_AUTH**

```
#define CONFIG_EHSM_JTAG_DEBUG_AUTH
```

configuration for jtag debug authentication support

Definition at line 461 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.110 CONFIG\_EHSM\_KMGR\_CHECK\_OTP\_K\_ATTR**

```
#define CONFIG_EHSM_KMGR_CHECK_OTP_K_ATTR
```

configuration for flash key backup support

configuration for batch\_write\_otp\_key\_attr support configuration for checking OTP key attributes

Definition at line 476 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.111 CONFIG\_EHSM\_KMGR\_PLAIN\_K\_IMPORT**

```
#define CONFIG_EHSM_KMGR_PLAIN_K_IMPORT
```

configuration for key signature support

configuration for import/export plain key support

Definition at line 486 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.112 CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_AREA\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_ECC_K_AREA_SIZE (0x800)
```

Definition at line 524 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.113 CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_END\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_ECC_K_END_ADDR (CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR)
```

Definition at line 526 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.114 CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_ECC_K_NUM (7)
```

Definition at line 522 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.115 CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_SLOT\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_ECC_K_SLOT_SIZE (256)
```

Definition at line 523 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.116 CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_START\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_ECC_K_START_ADDR (CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR)
```

Definition at line 525 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.117 CONFIG\_EHSM\_KMGR\_V\_ECC\_RAM\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_ECC_RAM_K_NUM (5)
```

Definition at line 545 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.118 CONFIG\_EHSM\_KMGR\_V\_FLASH\_K\_START\_OFFSET**

```
#define CONFIG_EHSM_KMGR_V_FLASH_K_START_OFFSET (0x00)
```

configuration for flash key storage address

Definition at line 499 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.119 CONFIG\_EHSM\_KMGR\_V\_FLASH\_OTP\_K\_START\_OFFSET**

```
#define CONFIG_EHSM_KMGR_V_FLASH_OTP_K_START_OFFSET (CONFIG_EHSM_KMGR_V_OTP_EXT_K_OFF)
```

Definition at line 500 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.120 CONFIG\_EHSM\_KMGR\_V\_MAX\_AUTH\_CODE\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_MAX_AUTH_CODE_SIZE (32U)
```

Max authentication code size of EVITA key.

Definition at line 491 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.121 CONFIG\_EHSM\_KMGR\_V\_OTP\_EXT\_K\_OFF**

```
#define CONFIG_EHSM_KMGR_V_OTP_EXT_K_OFF (0x400)
```

Definition at line 493 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.122 CONFIG\_EHSM\_KMGR\_V\_OTP\_EXT\_K\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_OTP_EXT_K_SIZE (0x400)
```

Definition at line 494 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.123 CONFIG\_EHSM\_KMGR\_V\_RAM\_K\_MEM\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_RAM_K_MEM_SIZE (2560)
```

configuration for EVITA ram key number and size

Definition at line 543 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.124 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_AREA\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_RSA_K_AREA_SIZE (0x800)
```

Definition at line 530 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.125 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_END\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_RSA_K_END_ADDR (CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR)
```

Definition at line 532 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.126 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_RSA_K_NUM (3)
```

Definition at line 528 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.127 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_SLOT\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_RSA_K_SLOT_SIZE (600)
```

Definition at line 529 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.128 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_START\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_RSA_K_START_ADDR (CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR)
```

Definition at line 531 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.129 CONFIG\_EHSM\_KMGR\_V\_RSA\_RAM\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_RSA_RAM_K_NUM (5)
```

Definition at line 546 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.130 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_AREA\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_SHE_K_AREA_SIZE (0x800)
```

Definition at line 507 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.131 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_END\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_SHE_K_END_ADDR
```

**Value:**

```
(CONFIG_EHSM_KMGR_V_SHE_K_START_ADDR + \
 CONFIG_EHSM_KMGR_V_SHE_K_AREA_SIZE)
```

Definition at line 509 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.132 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_SHE_K_NUM (40)
```

configuration for flash key number and size

Definition at line 505 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.133 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_SLOT\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_SHE_K_SLOT_SIZE (48)
```

Definition at line 506 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.134 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_START\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_SHE_K_START_ADDR (CONFIG_EHSM_KMGR_V_FLASH_K_START_OFFSET)
```

Definition at line 508 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.135 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_AREA\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_SM9_K_AREA_SIZE (0)
```

Definition at line 536 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.136 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_END\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_SM9_K_END_ADDR (CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR)
```

Definition at line 538 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.137 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_SM9_K_NUM (0)
```

Definition at line 534 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.138 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_SLOT\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_SM9_K_SLOT_SIZE (200)
```

Definition at line 535 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.139 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_START\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_SM9_K_START_ADDR (CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR)
```

Definition at line 537 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.140 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_AREA\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_SYM_K_AREA_SIZE (0x800)
```

Definition at line 517 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.141 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_END\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_SYM_K_END_ADDR
```

**Value:**

```
(CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR + \
 CONFIG_EHSM_KMGR_V_SYM_K_AREA_SIZE)
```

Definition at line 519 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.142 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_SYM_K_NUM (10)
```

configuration for EVITA key number and size

Definition at line 515 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.143 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_SLOT\_SIZE**

```
#define CONFIG_EHSM_KMGR_V_SYM_K_SLOT_SIZE (200)
```

Definition at line 516 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.144 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_START\_ADDR**

```
#define CONFIG_EHSM_KMGR_V_SYM_K_START_ADDR (CONFIG_EHSM_KMGR_V_SHE_K_END_ADDR)
```

Definition at line 518 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.145 CONFIG\_EHSM\_KMGR\_V\_SYM\_RAM\_K\_NUM**

```
#define CONFIG_EHSM_KMGR_V_SYM_RAM_K_NUM (10)
```

Definition at line 544 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.146 CONFIG\_EHSM\_LOG**

```
#define CONFIG_EHSM_LOG
```

configuration for eHSM log support

Definition at line 551 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.147 CONFIG\_EHSM\_PKE\_CORE\_NUM**

```
#define CONFIG_EHSM_PKE_CORE_NUM 1
```

Definition at line 618 of file eHSM\_IntCfg\_Ip.h.



**4.48.1.148 CONFIG\_EHSM\_SHE\_SOC\_BOOT**

```
#define CONFIG_EHSM_SHE_SOC_BOOT
```

configuration for reverse otp data, the otp default bit is 0, it can be change to bit 1

configuration for checking write protection of SHE keys when doing SHE debug authentication configuration for she soc boot

Definition at line 611 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.149 CONFIG\_EHSM\_SKE\_CORE\_NUM**

```
#define CONFIG_EHSM_SKE_CORE_NUM 1
```

configuration for multicore and core number.

Definition at line 617 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.150 CONFIG\_EHSM\_SOC\_UPGRADE\_AND\_VERIFY**

```
#define CONFIG_EHSM_SOC_UPGRADE_AND_VERIFY
```

configuration for OTP key crc byte reverse support

configuration for soc upgrade and verify support

Definition at line 561 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.151 CONFIG\_EHSM\_TRNG\_CORE\_NUM**

```
#define CONFIG_EHSM_TRNG_CORE_NUM 1
```

Definition at line 620 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.152 CONFIG\_EHSM\_UNIT\_TEST**

```
#define CONFIG_EHSM_UNIT_TEST
```

config for unittest

Definition at line 655 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.153 CONFIG\_EHSM\_USER\_AUTH\_KYE\_IN\_KMU**

```
#define CONFIG_EHSM_USER_AUTH_KYE_IN_KMU
```

configuration for system data backup support

configuration for debug with symmetric algorithms support PATCH for EHSM, iram space equal to rom space configuration for getting EMU status support configuration for handling EMU error configuration for self test support configuration for user auth key exist on KMU

Definition at line 596 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.154 CONFIG\_EHSM\_V\_LOG\_DEBUG**

```
#define CONFIG_EHSM_V_LOG_DEBUG 2
```

Definition at line 162 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.155 CONFIG\_EHSM\_V\_LOG\_ERR**

```
#define CONFIG_EHSM_V_LOG_ERR 0
```

configuration for EVB board support

configuration for log support

Definition at line 160 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.156 CONFIG\_EHSM\_V\_LOG\_INFO**

```
#define CONFIG_EHSM_V_LOG_INFO 3
```

Definition at line 163 of file eHSM\_IntCfg\_Ip.h.

**4.48.1.157 CONFIG\_EHSM\_V\_LOG\_LEVEL**

```
#define CONFIG_EHSM_V_LOG_LEVEL CONFIG_EHSM_V_LOG_DEBUG
```

Definition at line 164 of file eHSM\_IntCfg\_Ip.h.

## 4.48.1.158 CONFIG\_EHSM\_V\_LOG\_WARN

```
#define CONFIG_EHSM_V_LOG_WARN 1
```

Definition at line 161 of file eHSM\_IntCfg\_Ip.h.

## 4.49 eHSM\_Mailbox\_CmdId\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
```

### Macros

- #define [EHSM\\_CMD\\_WRITE\\_OTP\\_DATA](#) 0x00FEFF01U
- #define [EHSM\\_CMD\\_READ\\_OTP\\_DATA](#) 0x00FDFF02U
- #define [EHSM\\_CMD\\_GET\\_CHALLENGE](#) 0x00FCFF03U
- #define [EHSM\\_CMD\\_DEBUG\\_AUTHENCATION](#) 0x00FBFF04U
- #define [EHSM\\_CMD\\_IMAGE\\_UPGRADE](#) 0x00FAFF05U
- #define [EHSM\\_CMD\\_IMAGE\\_VERIFY](#) 0x00F9FF06U
- #define [EHSM\\_CMD\\_SOC\\_BOOT\\_STATUS](#) 0x00F8FF07U
- #define [EHSM\\_CMD\\_FW\\_GET\\_RANDOM\\_KEY](#) 0x00F7FF08U
- #define [EHSM\\_CMD\\_FW\\_ENCRYPT\\_KEY](#) 0x00F6FF09U
- #define [EHSM\\_CMD\\_CLOSE\\_DEBUG](#) 0x00F1FF0EU
- #define [EHSM\\_CMD\\_CREATE\\_COUNTER](#) 0xFFEF0010U
- #define [EHSM\\_CMD\\_READ\\_COUNTER](#) 0xFFEE0011U
- #define [EHSM\\_CMD\\_INCREASE\\_COUNTER](#) 0xFFED0012U
- #define [EHSM\\_CMD\\_DELETE\\_COUNTER](#) 0xFFEC0013U
- #define [EHSM\\_CMD\\_SELF\\_TEST](#) 0xFFDF0020U
- #define [EHSM\\_CMD\\_GET\\_SHE\\_STATUS](#) 0xFFBF0040U
- #define [EHSM\\_CMD\\_MODULE\\_STATUS](#) 0xFFAF0050U
- #define [EHSM\\_CMD\\_GET\\_SHE\\_ID](#) 0xFF9F0060U
- #define [EHSM\\_CMD\\_RESET\\_FIRMWARE](#) 0xFFEF00EEU
- #define [EHSM\\_CMD\\_SYM\\_CIPHER](#) 0xFEFE0101U
- #define [EHSM\\_CMD\\_AEAD\\_GCM](#) 0xFEFD0102U
- #define [EHSM\\_CMD\\_AEAD\\_CCM](#) 0xFEFC0103U
- #define [EHSM\\_CMD\\_SYM\\_GEN\\_KEY](#) 0xFEFB0104U
- #define [EHSM\\_CMD\\_MAC](#) 0xFDFE0201U
- #define [EHSM\\_CMD\\_HASH](#) 0xFCFE0301U
- #define [EHSM\\_CMD\\_SM2\\_CIPHER](#) 0xFBFE0401U
- #define [EHSM\\_CMD\\_SM2\\_SIGN](#) 0xFBFD0402U
- #define [EHSM\\_CMD\\_SM2\\_GEN\\_KEY](#) 0xFBFC0403U
- #define [EHSM\\_CMD\\_RSA\\_CIPHER](#) 0FAFE0501U
- #define [EHSM\\_CMD\\_RSA\\_SIGN](#) 0FAFD0502U
- #define [EHSM\\_CMD\\_RSA\\_GEN\\_KEY](#) 0FAFC0503U
- #define [EHSM\\_CMD\\_ECDSA](#) 0xF9FE0601U
- #define [EHSM\\_CMD\\_ECIES](#) 0xF9FD0602U
- #define [EHSM\\_CMD\\_ECCP\\_GEN\\_KEY](#) 0xF9FC0603U
- #define [EHSM\\_CMD\\_IMPORT\\_KEY](#) 0xF7FE0801U
- #define [EHSM\\_CMD\\_EXPORT\\_KEY](#) 0xF7FD0802U
- #define [EHSM\\_CMD\\_DERIVE\\_KEY](#) 0xF7FC0803U
- #define [EHSM\\_CMD\\_CREATE\\_DH\\_KEY](#) 0xF7FB0804U
- #define [EHSM\\_CMD\\_GET\\_PUB\\_FROM\\_PRIV](#) 0xF7FA0805U

- #define [EHSM\\_CMD\\_KEY\\_REMOVE](#) 0xF7F90806U
- #define [EHSM\\_CMD\\_KEY\\_STATUS](#) 0xF7F80807U
- #define [EHSM\\_CMD\\_SHE\\_LOAD\\_KEY](#) 0xF7F70808U
- #define [EHSM\\_CMD\\_SHE\\_LOAD\\_PLAIN\\_KEY](#) 0xF7F60809U
- #define [EHSM\\_CMD\\_SHE\\_RAM\\_KEY\\_EXPORT](#) 0xF7F5080AU
- #define [EHSM\\_CMD\\_COPY\\_EVITA\\_KEY](#) 0xF7F4080BU
- #define [EHSM\\_CMD\\_GEN\\_DH\\_KEY\\_PAIR](#) 0xF7F3080CU
- #define [EHSM\\_CMD\\_RNG\\_GENERATE](#) 0xF5FE0A01U
- #define [EHSM\\_CMD\\_CHANGE\\_LIFECYCLE](#) 0xF0FE0F01U
- #define [EHSM\\_CMD\\_CHANGE\\_CONTROL\\_FIELD](#) 0xF0FD0F02U
- #define [EHSM\\_CMD\\_LOW\\_POWER](#) 0xF0F90F03U
- #define [EHSM\\_CMD\\_SET\\_BAUDRATE](#) 0xF0FB0F04U
- #define [EHSM\\_CMD\\_SENSOR\\_RESP\\_INIT](#) 0xF0FA0F05U
- #define [EHSM\\_CMD\\_UART\\_COMMAND](#) 0xFF000001U
- #define [EHSM\\_CMD\\_UART\\_COMMAND](#) 0xFF000001U
- #define [EHSM\\_CMD\\_SOC\\_IMAGE\\_VERIFY](#) 0x00DDFF22U
- #define [EHSM\\_START](#) (1U)
- #define [EHSM\\_UPDATE](#) (2U)
- #define [EHSM\\_STREAMSTART](#) (3U)
- #define [EHSM\\_FINISH](#) (4U)
- #define [EHSM\\_ONEPASS](#) (7U)
- #define [EHSM\\_ENCRYPTION](#) (0U)
- #define [EHSM\\_DECRYPTION](#) (1U)
- #define [EHSM\\_MAC\\_GENERATION](#) (0U)
- #define [EHSM\\_MAC\\_VERIFICATION](#) (1U)
- #define [EHSM\\_SIGN\\_GENERATION](#) (1U)
- #define [EHSM\\_SIGN\\_VERIFICATION](#) (0U)
- #define [EHSM\\_INVALID\\_DIR](#) (0xFFU)
- #define [EHSM\\_NOPADDING](#) (0U)
- #define [EHSM\\_RSASSA\\_PPS](#) (1U)
- #define [EHSM\\_PKCS7](#) (2U)
- #define [EHSM\\_ONEWITHZEROS](#) (3U)
- #define [EHSM\\_SM4\\_CTRDRBG](#) (0U)
- #define [EHSM\\_AES\\_CTRDRBG](#) (1U)
- #define [EHSM\\_DES](#) (0U)
- #define [EHSM\\_TDES\\_128](#) (1U)
- #define [EHSM\\_TDES\\_192](#) (2U)
- #define [EHSM\\_AES\\_128](#) (5U)
- #define [EHSM\\_AES\\_192](#) (6U)
- #define [EHSM\\_AES\\_256](#) (7U)
- #define [EHSM\\_SM4](#) (8U)
- #define [EHSM\\_SM3](#) (0U)
- #define [EHSM\\_MD5](#) (1U)
- #define [EHSM\\_SHA256](#) (2U)
- #define [EHSM\\_SHA384](#) (3U)
- #define [EHSM\\_SHA512](#) (4U)
- #define [EHSM\\_SHA1](#) (5U)
- #define [EHSM\\_SHA224](#) (6U)
- #define [EHSM\\_SHA512\\_224](#) (7U)
- #define [EHSM\\_SHA512\\_256](#) (8U)
- #define [EHSM\\_SHA3\\_224](#) (9U)
- #define [EHSM\\_SHA3\\_256](#) (10U)
- #define [EHSM\\_SHA3\\_384](#) (11U)
- #define [EHSM\\_SHA3\\_512](#) (12U)
- #define [EHSM\\_INVALID\\_ALG](#) (0xFFU)
- #define [EHSM\\_ECB\\_MODE](#) (1U)
- #define [EHSM\\_XTS\\_MODE](#) (2U)

- #define [EHSM\\_CBC\\_MODE](#) (3U)
- #define [EHSM\\_CFB\\_MODE](#) (4U)
- #define [EHSM\\_OFB\\_MODE](#) (5U)
- #define [EHSM\\_CTR\\_MODE](#) (6U)
- #define [EHSM\\_NONE\\_CRT](#) (0U)
- #define [EHSM\\_CRT\\_MODE](#) (1U)
- #define [EHSM\\_CMAC\\_MODE](#) (7U)
- #define [EHSM\\_CBC\\_MAC\\_MODE](#) (8U)
- #define [EHSM\\_GMAC\\_MODE](#) (9U)
- #define [EHSM\\_NO\\_TIME\\_STAMP](#) (0U)
- #define [EHSM\\_USE\\_TIME\\_STAMP](#) (1U)
- #define [EHSM\\_XOR](#) (0U)

## Enumerations

- enum [cmd\\_type\\_e](#) {  
    [CMD\\_TYPE\\_SKE](#) = 1U, [CMD\\_TYPE\\_MAC](#) = 2U, [CMD\\_TYPE\\_HASH](#) = 3U, [CMD\\_TYPE\\_SM2](#) = 4U,  
    [CMD\\_TYPE\\_RSA](#) = 5U, [CMD\\_TYPE\\_ECC](#) = 6U, [CMD\\_TYPE\\_SM9](#) = 7U, [CMD\\_TYPE\\_RNG](#) = 10U }

### 4.49.1 Macro Definition Documentation

#### 4.49.1.1 EHSM\_AES\_128

```
#define EHSM_AES_128 (5U)
```

Definition at line 165 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.2 EHSM\_AES\_192

```
#define EHSM_AES_192 (6U)
```

Definition at line 166 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.3 EHSM\_AES\_256

```
#define EHSM_AES_256 (7U)
```

Definition at line 167 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.4 EHSM\_AES\_CTRDRBG

```
#define EHSM_AES_CTRDRBG (1U)
```

Definition at line 160 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.5 EHSM\_CBC\_MAC\_MODE

```
#define EHSM_CBC_MAC_MODE (8U)
```

Definition at line 197 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.6 EHSM\_CBC\_MODE

```
#define EHSM_CBC_MODE (3U)
```

Definition at line 188 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.7 EHSM\_CFB\_MODE

```
#define EHSM_CFB_MODE (4U)
```

Definition at line 189 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.8 EHSM\_CMAC\_MODE

```
#define EHSM_CMAC_MODE (7U)
```

Definition at line 196 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.9 EHSM\_CMD\_AEAD\_CCM

```
#define EHSM_CMD_AEAD_CCM 0xFEFC0103U
```

Definition at line 66 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.10 EHSM\_CMD\_AEAD\_GCM

```
#define EHSM_CMD_AEAD_GCM 0xFEFD0102U
```

Definition at line 65 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.11 EHSM\_CMD\_CHANGE\_CONTROL\_FIELD

```
#define EHSM_CMD_CHANGE_CONTROL_FIELD 0xF0FD0F02U
```

Definition at line 106 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.12 EHSM\_CMD\_CHANGE\_LIFECYCLE

```
#define EHSM_CMD_CHANGE_LIFECYCLE 0xF0FE0F01U
```

Definition at line 105 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.13 EHSM\_CMD\_CLOSE\_DEBUG

```
#define EHSM_CMD_CLOSE_DEBUG 0x00F1FF0EU
```

Definition at line 37 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.14 EHSM\_CMD\_COPY\_EVITA\_KEY

```
#define EHSM_CMD_COPY_EVITA_KEY 0xF7F4080BU
```

Definition at line 98 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.15 EHSM\_CMD\_CREATE\_COUNTER

```
#define EHSM_CMD_CREATE_COUNTER 0xFFEF0010U
```

Definition at line 49 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.16 EHSM\_CMD\_CREATE\_DH\_KEY**

```
#define EHSM_CMD_CREATE_DH_KEY 0xF7FB0804U
```

Definition at line 91 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.17 EHSM\_CMD\_DEBUG\_AUTHENCATION**

```
#define EHSM_CMD_DEBUG_AUTHENCATION 0x00FBFF04U
```

Definition at line 24 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.18 EHSM\_CMD\_DELETE\_COUNTER**

```
#define EHSM_CMD_DELETE_COUNTER 0xFFEC0013U
```

Definition at line 52 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.19 EHSM\_CMD\_DERIVE\_KEY**

```
#define EHSM_CMD_DERIVE_KEY 0xF7FC0803U
```

Definition at line 90 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.20 EHSM\_CMD\_ECCP\_GEN\_KEY**

```
#define EHSM_CMD_ECCP_GEN_KEY 0xF9FC0603U
```

Definition at line 84 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.21 EHSM\_CMD\_ECDSA**

```
#define EHSM_CMD_ECDSA 0xF9FE0601U
```

Definition at line 82 of file eHSM\_Mailbox\_CmdId\_Ip.h.



#### 4.49.1.22 EHSM\_CMD\_ECIES

```
#define EHSM_CMD_ECIES 0xF9FD0602U
```

Definition at line 83 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.23 EHSM\_CMD\_EXPORT\_KEY

```
#define EHSM_CMD_EXPORT_KEY 0xF7FD0802U
```

Definition at line 89 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.24 EHSM\_CMD\_FW\_ENCRYPT\_KEY

```
#define EHSM_CMD_FW_ENCRYPT_KEY 0x00F6FF09U
```

Definition at line 34 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.25 EHSM\_CMD\_FW\_GET\_RANDOM\_KEY

```
#define EHSM_CMD_FW_GET_RANDOM_KEY 0x00F7FF08U
```

Definition at line 33 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.26 EHSM\_CMD\_GEN\_DH\_KEY\_PAIR

```
#define EHSM_CMD_GEN_DH_KEY_PAIR 0xF7F3080CU
```

Definition at line 99 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.27 EHSM\_CMD\_GET\_CHALLENGE

```
#define EHSM_CMD_GET_CHALLENGE 0x00FCFF03U
```

Definition at line 23 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.28 EHSM\_CMD\_GET\_PUB\_FROM\_PRIV**

```
#define EHSM_CMD_GET_PUB_FROM_PRIV 0xF7FA0805U
```

Definition at line 92 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.29 EHSM\_CMD\_GET\_SHE\_ID**

```
#define EHSM_CMD_GET_SHE_ID 0xFF9F0060U
```

Definition at line 58 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.30 EHSM\_CMD\_GET\_SHE\_STATUS**

```
#define EHSM_CMD_GET_SHE_STATUS 0xFFBF0040U
```

Definition at line 56 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.31 EHSM\_CMD\_HASH**

```
#define EHSM_CMD_HASH 0xFCFE0301U
```

Definition at line 73 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.32 EHSM\_CMD\_IMAGE\_UPGRADE**

```
#define EHSM_CMD_IMAGE_UPGRADE 0x00FAFF05U
```

Definition at line 26 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.33 EHSM\_CMD\_IMAGE\_VERIFY**

```
#define EHSM_CMD_IMAGE_VERIFY 0x00F9FF06U
```

Definition at line 27 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.34 EHSM\_CMD\_IMPORT\_KEY

```
#define EHSM_CMD_IMPORT_KEY 0xF7FE0801U
```

Definition at line 88 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.35 EHSM\_CMD\_INCREASE\_COUNTER

```
#define EHSM_CMD_INCREASE_COUNTER 0xFFED0012U
```

Definition at line 51 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.36 EHSM\_CMD\_KEY\_REMOVE

```
#define EHSM_CMD_KEY_REMOVE 0xF7F90806U
```

Definition at line 93 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.37 EHSM\_CMD\_KEY\_STATUS

```
#define EHSM_CMD_KEY_STATUS 0xF7F80807U
```

Definition at line 94 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.38 EHSM\_CMD\_LOW\_POWER

```
#define EHSM_CMD_LOW_POWER 0xF0F90F03U
```

Definition at line 107 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.39 EHSM\_CMD\_MAC

```
#define EHSM_CMD_MAC 0xFDFE0201U
```

Definition at line 70 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.40 EHSM\_CMD\_MODULE\_STATUS

```
#define EHSM_CMD_MODULE_STATUS 0xFFAF0050U
```

Definition at line 57 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.41 EHSM\_CMD\_READ\_COUNTER

```
#define EHSM_CMD_READ_COUNTER 0xFFEE0011U
```

Definition at line 50 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.42 EHSM\_CMD\_READ\_OTP\_DATA

```
#define EHSM_CMD_READ_OTP_DATA 0x00FDFF02U
```

Definition at line 21 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.43 EHSM\_CMD\_RESET\_FIRMWARE

```
#define EHSM_CMD_RESET_FIRMWARE 0xFFEF00EEU
```

Definition at line 61 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.44 EHSM\_CMD\_RNG\_GENERATE

```
#define EHSM_CMD_RNG_GENERATE 0xF5FE0A01U
```

Definition at line 103 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.45 EHSM\_CMD\_RSA\_CIPHER

```
#define EHSM_CMD_RSA_CIPHER 0xFAFE0501U
```

Definition at line 79 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.46 EHSM\_CMD\_RSA\_GEN\_KEY**

```
#define EHSM_CMD_RSA_GEN_KEY 0xFAFC0503U
```

Definition at line 81 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.47 EHSM\_CMD\_RSA\_SIGN**

```
#define EHSM_CMD_RSA_SIGN 0xFAFD0502U
```

Definition at line 80 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.48 EHSM\_CMD\_SELF\_TEST**

```
#define EHSM_CMD_SELF_TEST 0xFFDF0020U
```

Definition at line 55 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.49 EHSM\_CMD\_SENSOR\_RESP\_INIT**

```
#define EHSM_CMD_SENSOR_RESP_INIT 0xF0FA0F05U
```

Definition at line 109 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.50 EHSM\_CMD\_SET\_BAUDRATE**

```
#define EHSM_CMD_SET_BAUDRATE 0xF0FB0F04U
```

Definition at line 108 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.51 EHSM\_CMD\_SHE\_LOAD\_KEY**

```
#define EHSM_CMD_SHE_LOAD_KEY 0xF7F70808U
```

Definition at line 95 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.52 EHSM\_CMD\_SHE\_LOAD\_PLAIN\_KEY**

```
#define EHSM_CMD_SHE_LOAD_PLAIN_KEY 0xF7F60809U
```

Definition at line 96 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.53 EHSM\_CMD\_SHE\_RAM\_KEY\_EXPORT**

```
#define EHSM_CMD_SHE_RAM_KEY_EXPORT 0xF7F5080AU
```

Definition at line 97 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.54 EHSM\_CMD\_SM2\_CIPHER**

```
#define EHSM_CMD_SM2_CIPHER 0xFBFE0401U
```

Definition at line 76 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.55 EHSM\_CMD\_SM2\_GEN\_KEY**

```
#define EHSM_CMD_SM2_GEN_KEY 0xFBFC0403U
```

Definition at line 78 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.56 EHSM\_CMD\_SM2\_SIGN**

```
#define EHSM_CMD_SM2_SIGN 0xFBFD0402U
```

Definition at line 77 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.57 EHSM\_CMD\_SOC\_BOOT\_STATUS**

```
#define EHSM_CMD_SOC_BOOT_STATUS 0x00F8FF07U
```

Definition at line 30 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.58 EHSM\_CMD\_SOC\_IMAGE\_VERIFY**

```
#define EHSM_CMD_SOC_IMAGE_VERIFY 0x00DDFF22U
```

Definition at line 129 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.59 EHSM\_CMD\_SYM\_CIPHER**

```
#define EHSM_CMD_SYM_CIPHER 0xFEFE0101U
```

Definition at line 64 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.60 EHSM\_CMD\_SYM\_GEN\_KEY**

```
#define EHSM_CMD_SYM_GEN_KEY 0xFEFB0104U
```

Definition at line 67 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.61 EHSM\_CMD\_UART\_COMMAND** [1/2]

```
#define EHSM_CMD_UART_COMMAND 0xFF000001U
```

Definition at line 122 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.62 EHSM\_CMD\_UART\_COMMAND** [2/2]

```
#define EHSM_CMD_UART_COMMAND 0xFF000001U
```

Definition at line 122 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.63 EHSM\_CMD\_WRITE\_OTP\_DATA**

```
#define EHSM_CMD_WRITE_OTP_DATA 0x00FEFF01U
```

Definition at line 20 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.64 EHSM\_CRT\_MODE

```
#define EHSM_CRT_MODE (1U)
```

Definition at line 194 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.65 EHSM\_CTR\_MODE

```
#define EHSM_CTR_MODE (6U)
```

Definition at line 191 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.66 EHSM\_DECRYPTION

```
#define EHSM_DECRYPTION (1U)
```

Definition at line 145 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.67 EHSM\_DES

```
#define EHSM_DES (0U)
```

Definition at line 162 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.68 EHSM\_ECB\_MODE

```
#define EHSM_ECB_MODE (1U)
```

Definition at line 186 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.69 EHSM\_ENCRYPTION

```
#define EHSM_ENCRYPTION (0U)
```

Definition at line 144 of file eHSM\_Mailbox\_CmdId\_Ip.h.



**4.49.1.70 EHSM\_FINISH**

```
#define EHSM_FINISH (4U)
```

Definition at line 140 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.71 EHSM\_GMAC\_MODE**

```
#define EHSM_GMAC_MODE (9U)
```

Definition at line 198 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.72 EHSM\_INVALID\_ALG**

```
#define EHSM_INVALID_ALG (0xFFU)
```

Definition at line 183 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.73 EHSM\_INVALID\_DIR**

```
#define EHSM_INVALID_DIR (0xFFU)
```

Definition at line 150 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.74 EHSM\_MAC\_GENERATION**

```
#define EHSM_MAC_GENERATION (0U)
```

Definition at line 146 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.75 EHSM\_MAC\_VERIFICATION**

```
#define EHSM_MAC_VERIFICATION (1U)
```

Definition at line 147 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.76 EHSM\_MD5**

```
#define EHSM_MD5 (1U)
```

Definition at line 171 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.77 EHSM\_NO\_TIME\_STAMP**

```
#define EHSM_NO_TIME_STAMP (0U)
```

Definition at line 201 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.78 EHSM\_NONE\_CRT**

```
#define EHSM_NONE_CRT (0U)
```

Definition at line 193 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.79 EHSM\_NOPADDING**

```
#define EHSM_NOPADDING (0U)
```

Definition at line 153 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.80 EHSM\_OFB\_MODE**

```
#define EHSM_OFB_MODE (5U)
```

Definition at line 190 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.81 EHSM\_ONEPASS**

```
#define EHSM_ONEPASS (7U)
```

Definition at line 141 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.82 EHSM\_ONEWITHZEROS

```
#define EHSM_ONEWITHZEROS (3U)
```

Definition at line 156 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.83 EHSM\_PKCS7

```
#define EHSM_PKCS7 (2U)
```

Definition at line 155 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.84 EHSM\_RSASSA\_PPS

```
#define EHSM_RSASSA_PPS (1U)
```

Definition at line 154 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.85 EHSM\_SHA1

```
#define EHSM_SHA1 (5U)
```

Definition at line 175 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.86 EHSM\_SHA224

```
#define EHSM_SHA224 (6U)
```

Definition at line 176 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.87 EHSM\_SHA256

```
#define EHSM_SHA256 (2U)
```

Definition at line 172 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.88 EHSM\_SHA384**

```
#define EHSM_SHA384 (3U)
```

Definition at line 173 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.89 EHSM\_SHA3\_224**

```
#define EHSM_SHA3_224 (9U)
```

Definition at line 179 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.90 EHSM\_SHA3\_256**

```
#define EHSM_SHA3_256 (10U)
```

Definition at line 180 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.91 EHSM\_SHA3\_384**

```
#define EHSM_SHA3_384 (11U)
```

Definition at line 181 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.92 EHSM\_SHA3\_512**

```
#define EHSM_SHA3_512 (12U)
```

Definition at line 182 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.93 EHSM\_SHA512**

```
#define EHSM_SHA512 (4U)
```

Definition at line 174 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.94 EHSM\_SHA512\_224**

```
#define EHSM_SHA512_224 (7U)
```

Definition at line 177 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.95 EHSM\_SHA512\_256**

```
#define EHSM_SHA512_256 (8U)
```

Definition at line 178 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.96 EHSM\_SIGN\_GENERATION**

```
#define EHSM_SIGN_GENERATION (1U)
```

Definition at line 148 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.97 EHSM\_SIGN\_VERIFICATION**

```
#define EHSM_SIGN_VERIFICATION (0U)
```

Definition at line 149 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.98 EHSM\_SM3**

```
#define EHSM_SM3 (0U)
```

Definition at line 170 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.99 EHSM\_SM4**

```
#define EHSM_SM4 (8U)
```

Definition at line 168 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.100 EHSM\_SM4\_CTRDRBG**

```
#define EHSM_SM4_CTRDRBG (0U)
```

Definition at line 159 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.101 EHSM\_START**

```
#define EHSM_START (1U)
```

Definition at line 137 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.102 EHSM\_STREAMSTART**

```
#define EHSM_STREAMSTART (3U)
```

Definition at line 139 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.103 EHSM\_TDES\_128**

```
#define EHSM_TDES_128 (1U)
```

Definition at line 163 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.104 EHSM\_TDES\_192**

```
#define EHSM_TDES_192 (2U)
```

Definition at line 164 of file eHSM\_Mailbox\_CmdId\_Ip.h.

**4.49.1.105 EHSM\_UPDATE**

```
#define EHSM_UPDATE (2U)
```

Definition at line 138 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.106 EHSM\_USE\_TIME\_STAMP

```
#define EHSM_USE_TIME_STAMP (1U)
```

Definition at line 202 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.107 EHSM\_XOR

```
#define EHSM_XOR (0U)
```

Definition at line 208 of file eHSM\_Mailbox\_CmdId\_Ip.h.

#### 4.49.1.108 EHSM\_XTS\_MODE

```
#define EHSM_XTS_MODE (2U)
```

Definition at line 187 of file eHSM\_Mailbox\_CmdId\_Ip.h.

### 4.49.2 Enumeration Type Documentation

#### 4.49.2.1 cmd\_type\_e

```
enum cmd_type_e
```

##### Enumerator

|               |  |
|---------------|--|
| CMD_TYPE_SKE  |  |
| CMD_TYPE_MAC  |  |
| CMD_TYPE_HASH |  |
| CMD_TYPE_SM2  |  |
| CMD_TYPE_RSA  |  |
| CMD_TYPE_ECC  |  |
| CMD_TYPE_SM9  |  |
| CMD_TYPE_RNG  |  |

Definition at line 216 of file eHSM\_Mailbox\_CmdId\_Ip.h.

## 4.50 eHSM\_Mailbox\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
```

```
#include <string.h>
#include "eHSM_Mailbox_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Mailbox_Reg_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Dspt_lp.h"
#include "eHSM_Exclusive_Area.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "test_int_config.h"
#include "AC784xx_Hsm_Reg.h"
#include "Device_Register.h"
#include "Core_Hal.h"
```

## Macros

- `#define HSMMBX_IRQ_PRIO` `PRIORITY_4`

## Functions

- `ehsm_uint32_t __attribute__((weak))`
- `void MAILBOX_Handler (void)`
- `ehsm_int32_t ehsm_mbox_init (void)`
- `void ehsm_mbox_polling ()`
- `ehsm_int32_t ehsm_mbox_send_cmd (ehsm_cmd_req_st *cmd)`
- `ehsm_uint32_t ehsm_is_cmd_addr_null (const ehsm_uint8_t cmd_addr[HOST_ADDRESS_SIZE])`

## Variables

- `mailbox_channel_st mbox_channel []`

### 4.50.1 Macro Definition Documentation

#### 4.50.1.1 HSMMBX\_IRQ\_PRIO

```
#define HSMMBX_IRQ_PRIO PRIORITY_4
```

Definition at line 30 of file eHSM\_Mailbox\_Ip.c.

### 4.50.2 Function Documentation



#### 4.50.2.1 \_\_attribute\_\_()

```
ehsm_uint32_t __attribute__ (
 (weak))
```

Definition at line 109 of file eHSM\_Mailbox\_Ip.c.

#### 4.50.2.2 ehsm\_is\_cmd\_addr\_null()

```
ehsm_uint32_t ehsm_is_cmd_addr_null (
 const ehsm_uint8_t cmd_addr[HOST_ADDRESS_SIZE])
```

Definition at line 404 of file eHSM\_Mailbox\_Ip.c.

#### 4.50.2.3 ehsm\_mbox\_init()

```
ehsm_int32_t ehsm_mbox_init (
 void)
```

Definition at line 294 of file eHSM\_Mailbox\_Ip.c.

#### 4.50.2.4 ehsm\_mbox\_polling()

```
void ehsm_mbox_polling ()
```

Definition at line 309 of file eHSM\_Mailbox\_Ip.c.

#### 4.50.2.5 ehsm\_mbox\_send\_cmd()

```
ehsm_int32_t ehsm_mbox_send_cmd (
 ehsm_cmd_req_st * cmd)
```

Definition at line 314 of file eHSM\_Mailbox\_Ip.c.

#### 4.50.2.6 MAILBOX\_Handler()

```
void MAILBOX_Handler (
 void)
```

Definition at line 279 of file eHSM\_Mailbox\_Ip.c.

### 4.50.3 Variable Documentation

#### 4.50.3.1 mbox\_channel

```
mailbox_channel_st mbox_channel[]
```

Definition at line 50 of file eHSM\_Mailbox\_Ip.c.

## 4.51 eHSM\_Mailbox\_Ip.h File Reference

```
#include "eHSM_Types_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
```

### Classes

- struct [mailbox\\_channel](#)

### Typedefs

- typedef void(\* [mailbox\\_callback](#)) ([ehsm\\_uint8\\_t](#) \*data\_ptr, [ehsm\\_uint32\\_t](#) size, [mailbox\\_channel\\_e](#) channel)
- typedef struct [mailbox\\_channel](#) [mailbox\\_channel\\_st](#)

### Functions

- [ehsm\\_int32\\_t ehsm\\_mbox\\_init](#) (void)
- [ehsm\\_int32\\_t ehsm\\_mbox\\_send\\_cmd](#) ([ehsm\\_cmd\\_req\\_st](#) \*cmd)
- [ehsm\\_uint32\\_t ehsm\\_is\\_cmd\\_addr\\_null](#) (const [ehsm\\_uint8\\_t](#) cmd\_addr[HOST\_ADDRESS\_SIZE])
- [ehsm\\_uint32\\_t ehsm\\_tick\\_form\\_ms](#) ([ehsm\\_uint32\\_t](#) ms)
- [ehsm\\_uint32\\_t ehsm\\_get\\_tick](#) (void)

### 4.51.1 Typedef Documentation

#### 4.51.1.1 mailbox\_callback

```
typedef void(* mailbox_callback) (ehsm_uint8_t *data_ptr, ehsm_uint32_t size, mailbox_channel_e channel)
```

Definition at line 21 of file eHSM\_Mailbox\_Ip.h.

#### 4.51.1.2 mailbox\_channel\_st

```
typedef struct mailbox_channel mailbox_channel_st
```

### 4.51.2 Function Documentation

#### 4.51.2.1 ehsm\_get\_tick()

```
ehsm_uint32_t ehsm_get_tick (
 void)
```

#### 4.51.2.2 ehsm\_is\_cmd\_addr\_null()

```
ehsm_uint32_t ehsm_is_cmd_addr_null (
 const ehsm_uint8_t cmd_addr[HOST_ADDRESS_SIZE])
```

Definition at line 404 of file eHSM\_Mailbox\_Ip.c.

#### 4.51.2.3 ehsm\_mbox\_init()

```
ehsm_int32_t ehsm_mbox_init (
 void)
```

Definition at line 294 of file eHSM\_Mailbox\_Ip.c.

#### 4.51.2.4 ehsm\_mbox\_send\_cmd()

```
ehsm_int32_t ehsm_mbox_send_cmd (
 ehsm_cmd_req_st * cmd)
```

Definition at line 314 of file eHSM\_Mailbox\_Ip.c.

#### 4.51.2.5 ehsm\_tick\_form\_ms()

```
ehsm_uint32_t ehsm_tick_form_ms (
 ehsm_uint32_t ms)
```

## 4.52 eHSM\_Mailbox\_Prtcl\_Ip.h File Reference

```
#include <eHSM_Types_Ip.h>
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Mailbox_CmdId_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
```

### Classes

- struct [ehsm\\_otp\\_read\\_cmd](#)
- struct [ehsm\\_otp\\_write\\_cmd](#)
- struct [ehsm\\_get\\_challenge\\_cmd](#)
- struct [ehsm\\_debug\\_authentication\\_cmd](#)
- struct [ehsm\\_close\\_debug\\_cmd](#)
- struct [ehsm\\_fw\\_get\\_random\\_key\\_cmd](#)
- struct [ehsm\\_fw\\_encrypt\\_key\\_cmd](#)
- struct [ehsm\\_image\\_upgrade\\_cmd](#)
- struct [ehsm\\_image\\_verify\\_cmd](#)
- struct [ehsm\\_soc\\_image\\_verify\\_cmd](#)
- struct [ehsm\\_derive\\_key\\_cmd](#)
- struct [ehsm\\_rng\\_generate\\_cmd](#)
- struct [ehsm\\_import\\_key\\_cmd](#)
- struct [ehsm\\_export\\_key\\_cmd](#)
- struct [ehsm\\_create\\_dh\\_sm2\\_ext\\_param](#)
- struct [ehsm\\_create\\_dh\\_key\\_cmd](#)
- struct [ehsm\\_get\\_pub\\_from\\_priv\\_cmd](#)
- struct [ehsm\\_key\\_remove\\_cmd](#)
- struct [ehsm\\_key\\_status\\_cmd](#)
- struct [ehsm\\_copy\\_key\\_cmd](#)
- struct [ehsm\\_she\\_load\\_export\\_key\\_cmd](#)
- struct [ehsm\\_she\\_load\\_plain\\_key\\_cmd](#)
- struct [ehsm\\_gen\\_key\\_cmd](#)
- struct [ehsm\\_sm9\\_exchg\\_key\\_cmd\\_child](#)
- struct [ehsm\\_sm9\\_exchg\\_key\\_cmd](#)
- struct [ehsm\\_uart\\_cmd](#)

*Struct for command of getting challenge, the command is from uart or JTAG channel of mailbox.*

- struct [ehsm\\_gen\\_sm9\\_userpriv\\_key\\_cmd](#)
- struct [ehsm\\_sm9\\_wrap\\_key\\_cmd](#)
- struct [ehsm\\_sm9\\_unwrap\\_key\\_cmd](#)
- struct [ehsm\\_sm9\\_export\\_key\\_cmd](#)
- struct [ehsm\\_sm9\\_import\\_key\\_cmd](#)
- struct [ehsm\\_sm9\\_get\\_mast\\_pubkey\\_cmd](#)
- struct [ehsm\\_sm9\\_get\\_tmp\\_pubkey\\_cmd](#)
- struct [ehsm\\_sm9\\_exchg\\_gen\\_usertmp\\_cmd](#)
- struct [ehsm\\_sm9\\_remove\\_key\\_cmd](#)
- struct [ehsm\\_low\\_power\\_cmd](#)
- struct [ehsm\\_change\\_lifecycle\\_cmd](#)
- struct [ehsm\\_change\\_control\\_field\\_cmd](#)
- struct [ehsm\\_sensor\\_resp\\_init\\_cmd](#)
- struct [ehsm\\_set\\_baudrate\\_cmd](#)
- struct [ehsm\\_get\\_she\\_id\\_cmd](#)
- struct [ehsm\\_get\\_emu\\_cmd](#)
- struct [ehsm\\_module\\_status\\_cmd](#)
- struct [ehsm\\_soc\\_secure\\_boot\\_status\\_st](#)

- struct [ehsm\\_mailbox\\_req](#)
- struct [ehsm\\_self\\_test\\_cmd](#)
- struct [ehsm\\_mbox\\_mgr\\_channel\\_req](#)
- struct [ehsm\\_mbox\\_cancel\\_channel\\_req](#)
- struct [ehsm\\_mbox\\_cancel\\_channel\\_rps](#)
- struct [ehsm\\_cmd\\_hdr\\_ske\\_st](#)  
*Header for Ske/Aead/Mac/Hash.*
- struct [ehsm\\_cmd\\_hdr\\_pke\\_st](#)  
*Header for sm2/rsa/ecdsa.*
- struct [ehsm\\_cmd\\_hdr\\_ecise\\_st](#)  
*Header for ecies.*
- struct [ehsm\\_cmd\\_hdr\\_eccp\\_keygen\\_st](#)  
*Header for sm2 and eccp key generation.*
- struct [ehsm\\_cmd\\_hdr\\_rsa\\_keygen\\_st](#)  
*Header for rsa key generation.*
- struct [ehsm\\_cmd\\_hdr\\_sm9\\_st](#)  
*Header for SM9 ciphert and signature.*
- struct [ehsm\\_cmd\\_sm9\\_sig\\_vry\\_output\\_ptr\\_st](#)  
*SM9 signature and verification structure. It's the content of [ehsm\\_cmd\\_cipher\\_st.output\\_addr](#).*
- struct [ehsm\\_cmd\\_aead\\_ptr\\_st](#)  
*GCM/CCM data structure. It's the content of [ehsm\\_cmd\\_cipher\\_st.input\\_addr/output\\_addr](#).*
- struct [ehsm\\_cmd\\_hdr\\_rng\\_st](#)  
*Header for RNG.*
- struct [ehsm\\_cmd\\_cipher\\_st](#)
- union [ehsm\\_cmd\\_cipher\\_st::osr\\_cmd\\_hdr\\_u](#)

## Macros

- #define [HOST\\_ADDRESS\\_SIZE](#) 4
- #define [CMD\\_TAG\\_WORD\\_SIZE](#) HOST\_ADDRESS\_SIZE/4
- #define [CMD\\_TAG\\_BYTE\\_SIZE](#) HOST\_ADDRESS\_SIZE
- #define [RESPONSE\\_TAG\\_INDEX](#) 0x10
- #define [MAX\\_KEY\\_DERIVED\\_PWD\\_SIZE](#) (256)
- #define [MAX\\_KEY\\_AUTH\\_VALUE\\_SIZE](#) EHSM\_EVITA\_AUTH\_VALUE\_MAX\_SIZE
- #define [MAX\\_IMPORT\\_KEY\\_SIZE](#) 256
- #define [MAX\\_SM9\\_USER\\_ID\\_SIZE](#) 1024
- #define [MAX\\_SM9\\_WARP\\_KEY\\_SIZE](#) 512
- #define [MAX\\_SM9\\_CIPHER\\_KEY\\_SIZE](#) 64
- #define [S2H\\_SRV\\_GENERAL\\_WORD\\_SIZE](#) 18U
- #define [S2H\\_SRV\\_MGR\\_WORD\\_SIZE](#) 3U
- #define [S2H\\_SRV\\_CMD\\_CANCEL\\_WORD\\_SIZE](#) 2U
- #define [S2H\\_SRV\\_JTAG\\_WORD\\_SIZE](#) 1U
- #define [S2H\\_SRV\\_GENERAL\\_WORD\\_INDEX](#) 0U
- #define [S2H\\_SRV\\_MGR\\_WORD\\_INDEX](#) 25U
- #define [S2H\\_SRV\\_CMD\\_CANCEL\\_WORD\\_INDEX](#) 28U
- #define [S2H\\_SRV\\_CMD\\_JTAG\\_WORD\\_INDEX](#) 31U
- #define [S2H\\_SRV\\_GENERAL\\_NOTE\\_BIT](#) ((ehsm\_uint32\_t)0x01<<0)
- #define [S2H\\_SRV\\_MGR\\_NOTE\\_BIT](#) ((ehsm\_uint32\_t)0x01<<6)
- #define [S2H\\_SRV\\_CMD\\_CANCEL\\_NOTE\\_BIT](#) ((ehsm\_uint32\_t)0x01<<9)
- #define [S2H\\_SRV\\_CMD\\_JTAG\\_NOTE\\_BIT](#) ((ehsm\_uint32\_t)0x01<<21)
- #define [S2H\\_SRV\\_CMD\\_JTAG\\_END\\_BIT](#) ((ehsm\_uint32\_t)0x01<<22)
- #define [H2S\\_SRV\\_GENERAL\\_WORD\\_SIZE](#) 5U
- #define [H2S\\_SRV\\_MGR\\_WORD\\_SIZE](#) 2U
- #define [H2S\\_SRV\\_CMD\\_CANCEL\\_WORD\\_SIZE](#) 3U

- #define H2S\_SRV\_JTAG\_WORD\_SIZE 1U
- #define H2S\_SRV\_GENERAL\_WORD\_INDEX 0U
- #define H2S\_SRV\_MGR\_WORD\_INDEX 7U
- #define H2S\_SRV\_CMD\_CANCEL\_WORD\_INDEX 9U
- #define H2S\_SRV\_JTAG\_WORD\_INDEX 14U
- #define H2S\_SRV\_GENERAL\_NOTE\_BIT ((ehsm\_uint32\_t)0x01<<0)
- #define H2S\_SRV\_MGR\_NOTE\_BIT ((ehsm\_uint32\_t)0x01<<6)
- #define H2S\_SRV\_CMD\_CANCEL\_NOTE\_BIT ((ehsm\_uint32\_t)0x01<<9)
- #define H2S\_SRV\_CMD\_JTAG\_NOTE\_BIT ((ehsm\_uint32\_t)0x01<<21)
- #define H2S\_SRV\_CMD\_JTAG\_END\_BIT ((ehsm\_uint32\_t)0x01<<22)
- #define EHSM\_LOW\_POWER\_MODE 0x1U
- #define EHSM\_NORMAL\_MODE 0x2U
- #define EHSM\_CANCEL\_SINGLE\_CMD 0x1
- #define EHSM\_CANCEL\_CERT\_TYPE\_CMD 0x2
- #define EHSM\_CMD\_CIPHER\_KEY\_TYPE\_SHE 0x01
- #define EHSM\_CMD\_CIPHER\_KEY\_TYPE\_EVITA 0x02
- #define CMD\_TAG\_WORD\_INDEX (S2H\_SRV\_GENERAL\_WORD\_SIZE - 1)
- #define MAILBOX\_CMD\_MAX\_SIZE (S2H\_SRV\_GENERAL\_WORD\_SIZE \* 4 + 4)

## Typedefs

- typedef struct ehsm\_otp\_read\_cmd ehsm\_otp\_read\_cmd\_st
  - typedef struct ehsm\_otp\_write\_cmd ehsm\_otp\_write\_cmd\_st
  - typedef struct ehsm\_get\_challenge\_cmd ehsm\_get\_challenge\_cmd\_st
  - typedef struct ehsm\_debug\_authentication\_cmd ehsm\_debug\_authentication\_cmd\_st
  - typedef struct ehsm\_close\_debug\_cmd ehsm\_close\_debug\_cmd\_st
  - typedef struct ehsm\_fw\_get\_random\_key\_cmd ehsm\_fw\_get\_random\_key\_cmd\_st
  - typedef struct ehsm\_fw\_encrypt\_key\_cmd ehsm\_fw\_encrypt\_key\_cmd\_st
  - typedef struct ehsm\_image\_upgrade\_cmd ehsm\_image\_upgrade\_cmd\_st
  - typedef struct ehsm\_image\_verfiy\_cmd ehsm\_image\_verify\_cmd\_st
  - typedef struct ehsm\_soc\_image\_verify\_cmd ehsm\_soc\_image\_verify\_cmd\_st
  - typedef struct ehsm\_derive\_key\_cmd ehsm\_derive\_key\_cmd\_st
  - typedef struct ehsm\_rng\_generate\_cmd ehsm\_rng\_generate\_cmd\_st
  - typedef struct ehsm\_import\_key\_cmd ehsm\_import\_key\_cmd\_st
  - typedef struct ehsm\_export\_key\_cmd ehsm\_export\_key\_cmd\_st
  - typedef struct ehsm\_create\_dh\_sm2\_ext\_param ehsm\_create\_dh\_sm2\_ext\_param\_st
  - typedef struct ehsm\_create\_dh\_key\_cmd ehsm\_create\_dh\_key\_cmd\_st
  - typedef struct ehsm\_get\_pub\_from\_priv\_cmd ehsm\_get\_pub\_from\_priv\_cmd\_st
  - typedef struct ehsm\_key\_remove\_cmd ehsm\_key\_remove\_cmd\_st
  - typedef struct ehsm\_key\_status\_cmd ehsm\_key\_status\_cmd\_st
  - typedef struct ehsm\_copy\_key\_cmd ehsm\_copy\_key\_cmd\_st
  - typedef struct ehsm\_she\_load\_export\_key\_cmd ehsm\_she\_load\_export\_key\_cmd\_st
  - typedef struct ehsm\_she\_load\_plain\_key\_cmd ehsm\_she\_load\_plain\_key\_cmd\_st
  - typedef struct ehsm\_gen\_key\_cmd ehsm\_gen\_key\_cmd\_st
  - typedef struct ehsm\_sm9\_exchg\_key\_cmd\_child ehsm\_sm9\_exchg\_key\_cmd\_child\_st
  - typedef struct ehsm\_sm9\_exchg\_key\_cmd ehsm\_sm9\_exchg\_key\_cmd\_st
  - typedef struct ehsm\_uart\_cmd ehsm\_uart\_cmd\_st
- Struct for command of getting challenge, the command is from uart or JTAG channel of mailbox.*
- typedef struct ehsm\_gen\_sm9\_userpriv\_key\_cmd ehsm\_gen\_sm9\_userpriv\_key\_cmd\_st
  - typedef struct ehsm\_sm9\_wrap\_key\_cmd ehsm\_sm9\_wrap\_key\_cmd\_st
  - typedef struct ehsm\_sm9\_unwrap\_key\_cmd ehsm\_sm9\_unwrap\_key\_cmd\_st
  - typedef struct ehsm\_sm9\_export\_key\_cmd ehsm\_sm9\_export\_key\_cmd\_st
  - typedef struct ehsm\_sm9\_import\_key\_cmd ehsm\_sm9\_import\_key\_cmd\_st
  - typedef struct ehsm\_sm9\_get\_mast\_pubkey\_cmd ehsm\_sm9\_get\_mast\_pubkey\_cmd\_st
  - typedef struct ehsm\_sm9\_get\_tmp\_pubkey\_cmd ehsm\_sm9\_get\_tmp\_pubkey\_cmd\_st
  - typedef struct ehsm\_sm9\_exchg\_gen\_usertmp\_cmd ehsm\_sm9\_exchg\_gen\_usertmp\_cmd\_st

- typedef struct ehsm\_sm9\_remove\_key\_cmd ehsm\_sm9\_remove\_key\_cmd\_st
- typedef struct ehsm\_low\_power\_cmd ehsm\_low\_power\_cmd\_st
- typedef struct ehsm\_change\_lifecycle\_cmd ehsm\_change\_lifecycle\_cmd\_st
- typedef struct ehsm\_change\_control\_field\_cmd ehsm\_change\_control\_field\_cmd\_st
- typedef struct ehsm\_sensor\_resp\_init\_cmd ehsm\_sensor\_resp\_init\_cmd\_st
- typedef struct ehsm\_set\_baudrate\_cmd ehsm\_set\_baudrate\_cmd\_st
- typedef struct ehsm\_get\_she\_id\_cmd ehsm\_get\_she\_id\_cmd\_st
- typedef struct ehsm\_get\_emu\_cmd ehsm\_get\_emu\_cmd\_st
- typedef struct ehsm\_module\_status\_cmd ehsm\_module\_status\_cmd\_st
- typedef struct ehsm\_mailbox\_req ehsm\_mailbox\_req\_st
- typedef struct ehsm\_self\_test\_cmd ehsm\_self\_test\_cmd\_st
- typedef struct ehsm\_mbox\_mgr\_channel\_req ehsm\_mbox\_mgr\_channel\_req\_st
- typedef struct ehsm\_mbox\_cancel\_channel\_req ehsm\_mbox\_cancel\_channel\_req\_st
- typedef struct ehsm\_mbox\_cancel\_channel\_rps ehsm\_mbox\_cancel\_channel\_rps\_st

## Enumerations

- enum mailbox\_channel\_e {  
MAILBOX\_CHANNE\_GENERAL\_SERVICE = 0U, MAILBOX\_CHANNE\_CMD\_CANCLE, MAILBOX\_CHANNE\_J↔  
TAG, MAILBOX\_CHANNE\_MGR\_SERVICE,  
MAILBOX\_CHANNE\_MAX }  
*mailbox channel*

## 4.52.1 Macro Definition Documentation

### 4.52.1.1 CMD\_TAG\_BYTE\_SIZE

```
#define CMD_TAG_BYTE_SIZE HOST_ADDRESS_SIZE
```

Definition at line 21 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

### 4.52.1.2 CMD\_TAG\_WORD\_INDEX

```
#define CMD_TAG_WORD_INDEX (S2H_SRV_GENERAL_WORD_SIZE - 1)
```

Definition at line 81 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

### 4.52.1.3 CMD\_TAG\_WORD\_SIZE

```
#define CMD_TAG_WORD_SIZE HOST_ADDRESS_SIZE/4
```

Definition at line 20 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.4 EHSM\_CANCEL\_CERT\_TYPE\_CMD

```
#define EHSM_CANCEL_CERT_TYPE_CMD 0x2
```

Definition at line 75 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.5 EHSM\_CANCEL\_SINGLE\_CMD

```
#define EHSM_CANCEL_SINGLE_CMD 0x1
```

Definition at line 74 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.6 EHSM\_CMD\_CIPHER\_KEY\_TYPE\_EVITA

```
#define EHSM_CMD_CIPHER_KEY_TYPE_EVITA 0x02
```

Definition at line 79 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.7 EHSM\_CMD\_CIPHER\_KEY\_TYPE\_SHE

```
#define EHSM_CMD_CIPHER_KEY_TYPE_SHE 0x01
```

Definition at line 78 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.8 EHSM\_LOW\_POWER\_MODE

```
#define EHSM_LOW_POWER_MODE 0x1U
```

Definition at line 70 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.9 EHSM\_NORMAL\_MODE

```
#define EHSM_NORMAL_MODE 0x2U
```

Definition at line 71 of file eHSM\_Mailbox\_Prtcl\_Ip.h.



#### 4.52.1.10 H2S\_SRV\_CMD\_CANCEL\_NOTE\_BIT

```
#define H2S_SRV_CMD_CANCEL_NOTE_BIT ((ehsm_uint32_t)0x01<<9)
```

Definition at line 65 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.11 H2S\_SRV\_CMD\_CANCEL\_WORD\_INDEX

```
#define H2S_SRV_CMD_CANCEL_WORD_INDEX 9U
```

Definition at line 59 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.12 H2S\_SRV\_CMD\_CANCEL\_WORD\_SIZE

```
#define H2S_SRV_CMD_CANCEL_WORD_SIZE 3U
```

Definition at line 53 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.13 H2S\_SRV\_CMD\_JTAG\_END\_BIT

```
#define H2S_SRV_CMD_JTAG_END_BIT ((ehsm_uint32_t)0x01<<22)
```

Definition at line 67 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.14 H2S\_SRV\_CMD\_JTAG\_NOTE\_BIT

```
#define H2S_SRV_CMD_JTAG_NOTE_BIT ((ehsm_uint32_t)0x01<<21)
```

Definition at line 66 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.15 H2S\_SRV\_GENERAL\_NOTE\_BIT

```
#define H2S_SRV_GENERAL_NOTE_BIT ((ehsm_uint32_t)0x01<<0)
```

Definition at line 63 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.16 H2S\_SRV\_GENERAL\_WORD\_INDEX

```
#define H2S_SRV_GENERAL_WORD_INDEX 0U
```

Definition at line 57 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.17 H2S\_SRV\_GENERAL\_WORD\_SIZE

```
#define H2S_SRV_GENERAL_WORD_SIZE 5U
```

Definition at line 51 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.18 H2S\_SRV\_JTAG\_WORD\_INDEX

```
#define H2S_SRV_JTAG_WORD_INDEX 14U
```

Definition at line 60 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.19 H2S\_SRV\_JTAG\_WORD\_SIZE

```
#define H2S_SRV_JTAG_WORD_SIZE 1U
```

Definition at line 54 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.20 H2S\_SRV\_MGR\_NOTE\_BIT

```
#define H2S_SRV_MGR_NOTE_BIT ((ehsm_uint32_t)0x01<<6)
```

Definition at line 64 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.21 H2S\_SRV\_MGR\_WORD\_INDEX

```
#define H2S_SRV_MGR_WORD_INDEX 7U
```

Definition at line 58 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.22 H2S\_SRV\_MGR\_WORD\_SIZE

```
#define H2S_SRV_MGR_WORD_SIZE 2U
```

Definition at line 52 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.23 HOST\_ADDRESS\_SIZE

```
#define HOST_ADDRESS_SIZE 4
```

Definition at line 18 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.24 MAILBOX\_CMD\_MAX\_SIZE

```
#define MAILBOX_CMD_MAX_SIZE (S2H_SRV_GENERAL_WORD_SIZE * 4 + 4)
```

Definition at line 82 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.25 MAX\_IMPORT\_KEY\_SIZE

```
#define MAX_IMPORT_KEY_SIZE 256
```

Definition at line 25 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.26 MAX\_KEY\_AUTH\_VALUE\_SIZE

```
#define MAX_KEY_AUTH_VALUE_SIZE EHSM_EVITA_AUTH_VALUE_MAX_SIZE
```

Definition at line 24 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.27 MAX\_KEY\_DERIVED\_PWD\_SIZE

```
#define MAX_KEY_DERIVED_PWD_SIZE (256)
```

Definition at line 23 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.28 MAX\_SM9\_CIPHER\_KEY\_SIZE

```
#define MAX_SM9_CIPHER_KEY_SIZE 64
```

Definition at line 28 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.29 MAX\_SM9\_USER\_ID\_SIZE

```
#define MAX_SM9_USER_ID_SIZE 1024
```

Definition at line 26 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.30 MAX\_SM9\_WARP\_KEY\_SIZE

```
#define MAX_SM9_WARP_KEY_SIZE 512
```

Definition at line 27 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.31 RESPONSE\_TAG\_INDEX

```
#define RESPONSE_TAG_INDEX 0x10
```

Definition at line 22 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.32 S2H\_SRV\_CMD\_CANCEL\_NOTE\_BIT

```
#define S2H_SRV_CMD_CANCEL_NOTE_BIT ((ehsm_uint32_t)0x01<<9)
```

Definition at line 46 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.33 S2H\_SRV\_CMD\_CANCEL\_WORD\_INDEX

```
#define S2H_SRV_CMD_CANCEL_WORD_INDEX 28U
```

Definition at line 40 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.34 S2H\_SRV\_CMD\_CANCEL\_WORD\_SIZE

```
#define S2H_SRV_CMD_CANCEL_WORD_SIZE 2U
```

Definition at line 34 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.35 S2H\_SRV\_CMD\_JTAG\_END\_BIT

```
#define S2H_SRV_CMD_JTAG_END_BIT ((ehsm_uint32_t)0x01<<22)
```

Definition at line 48 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.36 S2H\_SRV\_CMD\_JTAG\_NOTE\_BIT

```
#define S2H_SRV_CMD_JTAG_NOTE_BIT ((ehsm_uint32_t)0x01<<21)
```

Definition at line 47 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.37 S2H\_SRV\_CMD\_JTAG\_WORD\_INDEX

```
#define S2H_SRV_CMD_JTAG_WORD_INDEX 31U
```

Definition at line 41 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.38 S2H\_SRV\_GENERAL\_NOTE\_BIT

```
#define S2H_SRV_GENERAL_NOTE_BIT ((ehsm_uint32_t)0x01<<0)
```

Definition at line 44 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.39 S2H\_SRV\_GENERAL\_WORD\_INDEX

```
#define S2H_SRV_GENERAL_WORD_INDEX 0U
```

Definition at line 38 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.40 S2H\_SRV\_GENERAL\_WORD\_SIZE

```
#define S2H_SRV_GENERAL_WORD_SIZE 18U
```

Definition at line 32 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.41 S2H\_SRV\_JTAG\_WORD\_SIZE

```
#define S2H_SRV_JTAG_WORD_SIZE 1U
```

Definition at line 35 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.42 S2H\_SRV\_MGR\_NOTE\_BIT

```
#define S2H_SRV_MGR_NOTE_BIT ((ehsm_uint32_t)0x01<<6)
```

Definition at line 45 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.43 S2H\_SRV\_MGR\_WORD\_INDEX

```
#define S2H_SRV_MGR_WORD_INDEX 25U
```

Definition at line 39 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

#### 4.52.1.44 S2H\_SRV\_MGR\_WORD\_SIZE

```
#define S2H_SRV_MGR_WORD_SIZE 3U
```

Definition at line 33 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

### 4.52.2 Typedef Documentation

#### 4.52.2.1 ehsm\_change\_control\_field\_cmd\_st

```
typedef struct ehsm_change_control_field_cmd ehsm_change_control_field_cmd_st
```

#### 4.52.2.2 ehsm\_change\_lifecycle\_cmd\_st

```
typedef struct ehsm_change_lifecycle_cmd ehsm_change_lifecycle_cmd_st
```

#### 4.52.2.3 ehsm\_close\_debug\_cmd\_st

```
typedef struct ehsm_close_debug_cmd ehsm_close_debug_cmd_st
```

#### 4.52.2.4 ehsm\_copy\_key\_cmd\_st

```
typedef struct ehsm_copy_key_cmd ehsm_copy_key_cmd_st
```

#### 4.52.2.5 ehsm\_create\_dh\_key\_cmd\_st

```
typedef struct ehsm_create_dh_key_cmd ehsm_create_dh_key_cmd_st
```

#### 4.52.2.6 ehsm\_create\_dh\_sm2\_ext\_param\_st

```
typedef struct ehsm_create_dh_sm2_ext_param ehsm_create_dh_sm2_ext_param_st
```

#### 4.52.2.7 ehsm\_debug\_authentication\_cmd\_st

```
typedef struct ehsm_debug_authentication_cmd ehsm_debug_authentication_cmd_st
```

#### 4.52.2.8 ehsm\_derive\_key\_cmd\_st

```
typedef struct ehsm_derive_key_cmd ehsm_derive_key_cmd_st
```

#### 4.52.2.9 ehsm\_export\_key\_cmd\_st

```
typedef struct ehsm_export_key_cmd ehsm_export_key_cmd_st
```

**4.52.2.10 ehsm\_fw\_encrypt\_key\_cmd\_st**

```
typedef struct ehsm_fw_encrypt_key_cmd ehsm_fw_encrypt_key_cmd_st
```

**4.52.2.11 ehsm\_fw\_get\_random\_key\_cmd\_st**

```
typedef struct ehsm_fw_get_random_key_cmd ehsm_fw_get_random_key_cmd_st
```

**4.52.2.12 ehsm\_gen\_key\_cmd\_st**

```
typedef struct ehsm_gen_key_cmd ehsm_gen_key_cmd_st
```

**4.52.2.13 ehsm\_gen\_sm9\_userpriv\_key\_cmd\_st**

```
typedef struct ehsm_gen_sm9_userpriv_key_cmd ehsm_gen_sm9_userpriv_key_cmd_st
```

**4.52.2.14 ehsm\_get\_challenge\_cmd\_st**

```
typedef struct ehsm_get_challenge_cmd ehsm_get_challenge_cmd_st
```

**4.52.2.15 ehsm\_get\_emu\_cmd\_st**

```
typedef struct ehsm_get_emu_cmd ehsm_get_emu_cmd_st
```

**4.52.2.16 ehsm\_get\_pub\_from\_priv\_cmd\_st**

```
typedef struct ehsm_get_pub_from_priv_cmd ehsm_get_pub_from_priv_cmd_st
```

**4.52.2.17 ehsm\_get\_she\_id\_cmd\_st**

```
typedef struct ehsm_get_she_id_cmd ehsm_get_she_id_cmd_st
```



**4.52.2.18 ehsm\_image\_upgrade\_cmd\_st**

```
typedef struct ehsm_image_upgrade_cmd ehsm_image_upgrade_cmd_st
```

**4.52.2.19 ehsm\_image\_verify\_cmd\_st**

```
typedef struct ehsm_image_verfiy_cmd ehsm_image_verify_cmd_st
```

**4.52.2.20 ehsm\_import\_key\_cmd\_st**

```
typedef struct ehsm_import_key_cmd ehsm_import_key_cmd_st
```

**4.52.2.21 ehsm\_key\_remove\_cmd\_st**

```
typedef struct ehsm_key_remove_cmd ehsm_key_remove_cmd_st
```

**4.52.2.22 ehsm\_key\_status\_cmd\_st**

```
typedef struct ehsm_key_status_cmd ehsm_key_status_cmd_st
```

**4.52.2.23 ehsm\_low\_power\_cmd\_st**

```
typedef struct ehsm_low_power_cmd ehsm_low_power_cmd_st
```

**4.52.2.24 ehsm\_mailbox\_req\_st**

```
typedef struct ehsm_mailbox_req ehsm_mailbox_req_st
```

**4.52.2.25 ehsm\_mbox\_cancel\_channel\_req\_st**

```
typedef struct ehsm_mbox_cancel_channel_req ehsm_mbox_cancel_channel_req_st
```

**4.52.2.26 ehsm\_mbox\_cancel\_channel\_rps\_st**

```
typedef struct ehsm_mbox_cancel_channel_rps ehsm_mbox_cancel_channel_rps_st
```

**4.52.2.27 ehsm\_mbox\_mgr\_channel\_req\_st**

```
typedef struct ehsm_mbox_mgr_channel_req ehsm_mbox_mgr_channel_req_st
```

**4.52.2.28 ehsm\_module\_status\_cmd\_st**

```
typedef struct ehsm_module_status_cmd ehsm_module_status_cmd_st
```

**4.52.2.29 ehsm\_otp\_read\_cmd\_st**

```
typedef struct ehsm_otp_read_cmd ehsm_otp_read_cmd_st
```

**4.52.2.30 ehsm\_otp\_write\_cmd\_st**

```
typedef struct ehsm_otp_write_cmd ehsm_otp_write_cmd_st
```

**4.52.2.31 ehsm\_rng\_generate\_cmd\_st**

```
typedef struct ehsm_rng_generate_cmd ehsm_rng_generate_cmd_st
```

**4.52.2.32 ehsm\_self\_test\_cmd\_st**

```
typedef struct ehsm_self_test_cmd ehsm_self_test_cmd_st
```

**4.52.2.33 ehsm\_sensor\_resp\_init\_cmd\_st**

```
typedef struct ehsm_sensor_resp_init_cmd ehsm_sensor_resp_init_cmd_st
```

**4.52.2.34 ehsm\_set\_baudrate\_cmd\_st**

```
typedef struct ehsm_set_baudrate_cmd ehsm_set_baudrate_cmd_st
```

**4.52.2.35 ehsm\_she\_load\_export\_key\_cmd\_st**

```
typedef struct ehsm_she_load_export_key_cmd ehsm_she_load_export_key_cmd_st
```

**4.52.2.36 ehsm\_she\_load\_plain\_key\_cmd\_st**

```
typedef struct ehsm_she_load_plain_key_cmd ehsm_she_load_plain_key_cmd_st
```

**4.52.2.37 ehsm\_sm9\_exchg\_gen\_usertmp\_cmd\_st**

```
typedef struct ehsm_sm9_exchg_gen_usertmp_cmd ehsm_sm9_exchg_gen_usertmp_cmd_st
```

**4.52.2.38 ehsm\_sm9\_exchg\_key\_cmd\_child\_st**

```
typedef struct ehsm_sm9_exchg_key_cmd_child ehsm_sm9_exchg_key_cmd_child_st
```

**4.52.2.39 ehsm\_sm9\_exchg\_key\_cmd\_st**

```
typedef struct ehsm_sm9_exchg_key_cmd ehsm_sm9_exchg_key_cmd_st
```

**4.52.2.40 ehsm\_sm9\_export\_key\_cmd\_st**

```
typedef struct ehsm_sm9_export_key_cmd ehsm_sm9_export_key_cmd_st
```

**4.52.2.41 ehsm\_sm9\_get\_mast\_pubkey\_cmd\_st**

```
typedef struct ehsm_sm9_get_mast_pubkey_cmd ehsm_sm9_get_mast_pubkey_cmd_st
```

#### 4.52.2.42 ehsm\_sm9\_get\_tmp\_pubkey\_cmd\_st

```
typedef struct ehsm_sm9_get_tmp_pubkey_cmd ehsm_sm9_get_tmp_pubkey_cmd_st
```

#### 4.52.2.43 ehsm\_sm9\_import\_key\_cmd\_st

```
typedef struct ehsm_sm9_import_key_cmd ehsm_sm9_import_key_cmd_st
```

#### 4.52.2.44 ehsm\_sm9\_remove\_key\_cmd\_st

```
typedef struct ehsm_sm9_remove_key_cmd ehsm_sm9_remove_key_cmd_st
```

#### 4.52.2.45 ehsm\_sm9\_unwrap\_key\_cmd\_st

```
typedef struct ehsm_sm9_unwrap_key_cmd ehsm_sm9_unwrap_key_cmd_st
```

#### 4.52.2.46 ehsm\_sm9\_wrap\_key\_cmd\_st

```
typedef struct ehsm_sm9_wrap_key_cmd ehsm_sm9_wrap_key_cmd_st
```

#### 4.52.2.47 ehsm\_soc\_image\_verify\_cmd\_st

```
typedef struct ehsm_soc_image_verify_cmd ehsm_soc_image_verify_cmd_st
```

#### 4.52.2.48 ehsm\_uart\_cmd\_st

```
typedef struct ehsm_uart_cmd ehsm_uart_cmd_st
```

Struct for command of getting challenge, the command is from uart or JTAG channel of mailbox.

### 4.52.3 Enumeration Type Documentation

#### 4.52.3.1 mailbox\_channel\_e

```
enum mailbox_channel_e
```

mailbox channel

## Enumerator

|                                |  |
|--------------------------------|--|
| MAILBOX_CHANNE_GENERAL_SERVICE |  |
| MAILBOX_CHANNE_CMD_CANCEL      |  |
| MAILBOX_CHANNE_JTAG            |  |
| MAILBOX_CHANNE_MGR_SERVICE     |  |
| MAILBOX_CHANNE_MAX             |  |

Definition at line 90 of file eHSM\_Mailbox\_Prtcl\_Ip.h.

## 4.53 eHSM\_Mailbox\_Reg\_Ip.h File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
```

## Macros

- #define [HOST2HSM\\_ACCESS\\_PATH](#) \*((volatile uint32 \*)0xE0084018UL)
- #define [STATUS\\_BASE](#) (0x40010000)
- #define [HSM\\_STATUS\\_IN](#) \*((volatile unsigned int \*)(STATUS\_BASE + 0x60))
- #define [HSM\\_STATUS\\_IN1](#) \*((volatile unsigned int \*)(STATUS\_BASE + 0x64))
- #define [MBOX\\_SOCBASE](#) (0x400F0000)
- #define [rSOCMBOX\\_CMD\\_D0](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x00\*4))
- #define [rSOCMBOX\\_CMD\\_D1](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x01\*4))
- #define [rSOCMBOX\\_CMD\\_D2](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x02\*4))
- #define [rSOCMBOX\\_CMD\\_D3](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x03\*4))
- #define [rSOCMBOX\\_CMD\\_D4](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x04\*4))
- #define [rSOCMBOX\\_CMD\\_D5](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x05\*4))
- #define [rSOCMBOX\\_CMD\\_D6](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x06\*4))
- #define [rSOCMBOX\\_CMD\\_D7](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x07\*4))
- #define [rSOCMBOX\\_CMD\\_D8](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x08\*4))
- #define [rSOCMBOX\\_CMD\\_D9](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x09\*4))
- #define [rSOCMBOX\\_CMD\\_D10](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x0A\*4))
- #define [rSOCMBOX\\_CMD\\_D11](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x0B\*4))
- #define [rSOCMBOX\\_CMD\\_D12](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x0C\*4))
- #define [rSOCMBOX\\_CMD\\_D13](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x0D\*4))
- #define [rSOCMBOX\\_CMD\\_D14](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x0E\*4))
- #define [rSOCMBOX\\_CMD\\_D15](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x0F\*4))
- #define [rSOCMBOX\\_RSP\\_D0](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x80))
- #define [rSOCMBOX\\_RSP\\_D01](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x84))
- #define [MB\\_S2H\\_NOTE](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x100))
- #define [MB\\_H2S\\_NOTE](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x104))
- #define [MB\\_H2S\\_SOC\\_INT](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x120))
- #define [MB\\_H2S\\_SOC\\_INT\\_EN](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x124))
- #define [MB\\_HSM\\_STATUS0](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x200))
- #define [MB\\_HSM\\_STATUS1](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x210))
- #define [MB\\_S2H\\_SOC\\_INT](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x110))
- #define [MB\\_S2H\\_SOC\\_INT\\_EN](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x114))
- #define [MBOX\\_HOST2HSM\\_HOST\\_INT](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x110UL))
- #define [MBOX\\_HOST2HSM\\_HOST\\_INT\\_EN](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x114UL))
- #define [MBOX\\_HOST2HSM\\_HSM\\_INT](#) \*((volatile unsigned int \*)(MBOX\_SOCBASE + 0x118UL))

- #define `MBOX_HOST2HSM_HSM_INT_EN` \*((volatile unsigned int \*)(`MBOX_SOCBASE` + 0x11CUL))
- #define `MBOX_HSM2HOST_HOST_INT` \*((volatile unsigned int \*)(`MBOX_SOCBASE` + 0x120UL))
- #define `MBOX_HSM2HOST_HOST_INT_EN` \*((volatile unsigned int \*)(`MBOX_SOCBASE` + 0x124UL))
- #define `MBOX_HSMHOST_HSM_INT` \*((volatile unsigned int \*)(`MBOX_SOCBASE` + 0x128UL))
- #define `MBOX_HSMHOST_HSM_INT_EN` \*((volatile unsigned int \*)(`MBOX_SOCBASE` + 0x12CUL))
- #define `KMU_BASE` (0xC21D4000)
- #define `rKMU_CTRL` \*((volatile unsigned int \*)(`KMU_BASE` + 0x00))
- #define `rKMU_STA` \*((volatile unsigned int \*)(`KMU_BASE` + 0x01))
- #define `rKMU_INT_EN` \*((volatile unsigned int \*)(`KMU_BASE` + 0x02))
- #define `rVER_D0` \*((volatile unsigned int \*)(`KMU_BASE` + 0x04))
- #define `rVER_D1` \*((volatile unsigned int \*)(`KMU_BASE` + 0x05))
- #define `rVER_D2` \*((volatile unsigned int \*)(`KMU_BASE` + 0x06))
- #define `rVER_D3` \*((volatile unsigned int \*)(`KMU_BASE` + 0x07))
- #define `rSN_D0` \*((volatile unsigned int \*)(`KMU_BASE` + 0x08))
- #define `rSN_D1` \*((volatile unsigned int \*)(`KMU_BASE` + 0x09))
- #define `rATTR_D0` \*((volatile unsigned int \*)(`KMU_BASE` + 0x0A))
- #define `rATTR_D1` \*((volatile unsigned int \*)(`KMU_BASE` + 0x0B))
- #define `rATTR_D2` \*((volatile unsigned int \*)(`KMU_BASE` + 0x0C))
- #define `KBUF_BASE` (`rAHB_BASE` + 0x14200 )
- #define `rKBUF` \*((volatile unsigned int \*)(`KBUF_BASE` + 0x00))
- #define `rERR_ST` \*((volatile unsigned int \*) (0x021D0000 + (0x0A<<2)))

### 4.53.1 Macro Definition Documentation

#### 4.53.1.1 HOST2HSM\_ACCESS\_PATH

```
#define HOST2HSM_ACCESS_PATH (*(volatile uint32 *)0xE0084018UL)
```

Definition at line 18 of file `eHSM_Mailbox_Reg_Ip.h`.

#### 4.53.1.2 HSM\_STATUS\_IN

```
#define HSM_STATUS_IN *((volatile unsigned int *) (STATUS_BASE + 0x60))
```

Definition at line 22 of file `eHSM_Mailbox_Reg_Ip.h`.

#### 4.53.1.3 HSM\_STATUS\_IN1

```
#define HSM_STATUS_IN1 *((volatile unsigned int *) (STATUS_BASE + 0x64))
```

Definition at line 23 of file `eHSM_Mailbox_Reg_Ip.h`.

#### 4.53.1.4 KBUF\_BASE

```
#define KBUF_BASE (rAHB_BASE + 0x14200)
```

Definition at line 88 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.5 KMU\_BASE

```
#define KMU_BASE (0xC21D4000)
```

Definition at line 75 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.6 MB\_H2S\_NOTE

```
#define MB_H2S_NOTE *((volatile unsigned int *) (MBOX_SOCBASE + 0x104))
```

Definition at line 47 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.7 MB\_H2S\_SOC\_INT

```
#define MB_H2S_SOC_INT *((volatile unsigned int *) (MBOX_SOCBASE + 0x120))
```

Definition at line 48 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.8 MB\_H2S\_SOC\_INT\_EN

```
#define MB_H2S_SOC_INT_EN *((volatile unsigned int *) (MBOX_SOCBASE + 0x124))
```

Definition at line 49 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.9 MB\_HSM\_STATUS0

```
#define MB_HSM_STATUS0 *((volatile unsigned int *) (MBOX_SOCBASE + 0x200))
```

Definition at line 50 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.10 MB\_HSM\_STATUS1

```
#define MB_HSM_STATUS1 *((volatile unsigned int *) (MBOX_SOCBASE + 0x210))
```

Definition at line 51 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.11 MB\_S2H\_NOTE

```
#define MB_S2H_NOTE *((volatile unsigned int *) (MBOX_SOCBASE + 0x100))
```

Definition at line 46 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.12 MB\_S2H\_SOC\_INT

```
#define MB_S2H_SOC_INT *((volatile unsigned int *) (MBOX_SOCBASE + 0x110))
```

Definition at line 52 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.13 MB\_S2H\_SOC\_INT\_EN

```
#define MB_S2H_SOC_INT_EN *((volatile unsigned int *) (MBOX_SOCBASE + 0x114))
```

Definition at line 53 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.14 MBOX\_HOST2HSM\_HOST\_INT

```
#define MBOX_HOST2HSM_HOST_INT *((volatile unsigned int *) (MBOX_SOCBASE + 0x110UL))
```

Definition at line 55 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.15 MBOX\_HOST2HSM\_HOST\_INT\_EN

```
#define MBOX_HOST2HSM_HOST_INT_EN *((volatile unsigned int *) (MBOX_SOCBASE + 0x114UL))
```

Definition at line 56 of file eHSM\_Mailbox\_Reg\_Ip.h.



**4.53.1.16 MBOX\_HOST2HSM\_HSM\_INT**

```
#define MBOX_HOST2HSM_HSM_INT *((volatile unsigned int *) (MBOX_SOCBASE + 0x118UL))
```

Definition at line 57 of file eHSM\_Mailbox\_Reg\_Ip.h.

**4.53.1.17 MBOX\_HOST2HSM\_HSM\_INT\_EN**

```
#define MBOX_HOST2HSM_HSM_INT_EN *((volatile unsigned int *) (MBOX_SOCBASE + 0x11CUL))
```

Definition at line 58 of file eHSM\_Mailbox\_Reg\_Ip.h.

**4.53.1.18 MBOX\_HSM2HOST\_HOST\_INT**

```
#define MBOX_HSM2HOST_HOST_INT *((volatile unsigned int *) (MBOX_SOCBASE + 0x120UL))
```

Definition at line 60 of file eHSM\_Mailbox\_Reg\_Ip.h.

**4.53.1.19 MBOX\_HSM2HOST\_HOST\_INT\_EN**

```
#define MBOX_HSM2HOST_HOST_INT_EN *((volatile unsigned int *) (MBOX_SOCBASE + 0x124UL))
```

Definition at line 61 of file eHSM\_Mailbox\_Reg\_Ip.h.

**4.53.1.20 MBOX\_HSMHOST\_HSM\_INT**

```
#define MBOX_HSMHOST_HSM_INT *((volatile unsigned int *) (MBOX_SOCBASE + 0x128UL))
```

Definition at line 62 of file eHSM\_Mailbox\_Reg\_Ip.h.

**4.53.1.21 MBOX\_HSMHOST\_HSM\_INT\_EN**

```
#define MBOX_HSMHOST_HSM_INT_EN *((volatile unsigned int *) (MBOX_SOCBASE + 0x12CUL))
```

Definition at line 63 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.22 MBOX\_SOCBASE

```
#define MBOX_SOCBASE (0x400F0000)
```

Definition at line 25 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.23 rATTR\_D0

```
#define rATTR_D0 *((volatile unsigned int *) (KMU_BASE +0x0A))
```

Definition at line 85 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.24 rATTR\_D1

```
#define rATTR_D1 *((volatile unsigned int *) (KMU_BASE +0x0B))
```

Definition at line 86 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.25 rATTR\_D2

```
#define rATTR_D2 *((volatile unsigned int *) (KMU_BASE +0x0C))
```

Definition at line 87 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.26 rERR\_ST

```
#define rERR_ST *((volatile unsigned int *) (0x021D0000 + (0x0A<<2)))
```

Definition at line 91 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.27 rKBUF

```
#define rKBUF *((volatile unsigned int *) (KBUF_BASE +0x00))
```

Definition at line 89 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.28 rKMU\_CTRL

```
#define rKMU_CTRL *((volatile unsigned int *) (KMU_BASE +0x00))
```

Definition at line 76 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.29 rKMU\_INT\_EN

```
#define rKMU_INT_EN *((volatile unsigned int *) (KMU_BASE +0x02))
```

Definition at line 78 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.30 rKMU\_STA

```
#define rKMU_STA *((volatile unsigned int *) (KMU_BASE +0x01))
```

Definition at line 77 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.31 rSN\_D0

```
#define rSN_D0 *((volatile unsigned int *) (KMU_BASE +0x08))
```

Definition at line 83 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.32 rSN\_D1

```
#define rSN_D1 *((volatile unsigned int *) (KMU_BASE +0x09))
```

Definition at line 84 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.33 rSOCMBOX\_CMD\_D0

```
#define rSOCMBOX_CMD_D0 *((volatile unsigned int *) (MBOX_SOCBASE + 0x00*4))
```

Definition at line 27 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.34 rSOCMBOX\_CMD\_D1

```
#define rSOCMBOX_CMD_D1 *((volatile unsigned int *) (MBOX_SOCBASE + 0x01*4))
```

Definition at line 28 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.35 rSOCMBOX\_CMD\_D10

```
#define rSOCMBOX_CMD_D10 *((volatile unsigned int *) (MBOX_SOCBASE + 0x0A*4))
```

Definition at line 37 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.36 rSOCMBOX\_CMD\_D11

```
#define rSOCMBOX_CMD_D11 *((volatile unsigned int *) (MBOX_SOCBASE + 0x0B*4))
```

Definition at line 38 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.37 rSOCMBOX\_CMD\_D12

```
#define rSOCMBOX_CMD_D12 *((volatile unsigned int *) (MBOX_SOCBASE + 0x0C*4))
```

Definition at line 39 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.38 rSOCMBOX\_CMD\_D13

```
#define rSOCMBOX_CMD_D13 *((volatile unsigned int *) (MBOX_SOCBASE + 0x0D*4))
```

Definition at line 40 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.39 rSOCMBOX\_CMD\_D14

```
#define rSOCMBOX_CMD_D14 *((volatile unsigned int *) (MBOX_SOCBASE + 0x0E*4))
```

Definition at line 41 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.40 rSOCMBOX\_CMD\_D15

```
#define rSOCMBOX_CMD_D15 *((volatile unsigned int *) (MBOX_SOCBASE + 0x0F*4))
```

Definition at line 42 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.41 rSOCMBOX\_CMD\_D2

```
#define rSOCMBOX_CMD_D2 *((volatile unsigned int *) (MBOX_SOCBASE + 0x02*4))
```

Definition at line 29 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.42 rSOCMBOX\_CMD\_D3

```
#define rSOCMBOX_CMD_D3 *((volatile unsigned int *) (MBOX_SOCBASE + 0x03*4))
```

Definition at line 30 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.43 rSOCMBOX\_CMD\_D4

```
#define rSOCMBOX_CMD_D4 *((volatile unsigned int *) (MBOX_SOCBASE + 0x04*4))
```

Definition at line 31 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.44 rSOCMBOX\_CMD\_D5

```
#define rSOCMBOX_CMD_D5 *((volatile unsigned int *) (MBOX_SOCBASE + 0x05*4))
```

Definition at line 32 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.45 rSOCMBOX\_CMD\_D6

```
#define rSOCMBOX_CMD_D6 *((volatile unsigned int *) (MBOX_SOCBASE + 0x06*4))
```

Definition at line 33 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.46 rSOCMBOX\_CMD\_D7

```
#define rSOCMBOX_CMD_D7 *((volatile unsigned int *) (MBOX_SOCBASE + 0x07*4))
```

Definition at line 34 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.47 rSOCMBOX\_CMD\_D8

```
#define rSOCMBOX_CMD_D8 *((volatile unsigned int *) (MBOX_SOCBASE + 0x08*4))
```

Definition at line 35 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.48 rSOCMBOX\_CMD\_D9

```
#define rSOCMBOX_CMD_D9 *((volatile unsigned int *) (MBOX_SOCBASE + 0x09*4))
```

Definition at line 36 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.49 rSOCMBOX\_RSP\_D0

```
#define rSOCMBOX_RSP_D0 *((volatile unsigned int *) (MBOX_SOCBASE + 0x80))
```

Definition at line 43 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.50 rSOCMBOX\_RSP\_D01

```
#define rSOCMBOX_RSP_D01 *((volatile unsigned int *) (MBOX_SOCBASE + 0x84))
```

Definition at line 44 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.51 rVER\_D0

```
#define rVER_D0 *((volatile unsigned int *) (KMU_BASE +0x04))
```

Definition at line 79 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.52 rVER\_D1

```
#define rVER_D1 *((volatile unsigned int *) (KMU_BASE +0x05))
```

Definition at line 80 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.53 rVER\_D2

```
#define rVER_D2 *((volatile unsigned int *) (KMU_BASE +0x06))
```

Definition at line 81 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.54 rVER\_D3

```
#define rVER_D3 *((volatile unsigned int *) (KMU_BASE +0x07))
```

Definition at line 82 of file eHSM\_Mailbox\_Reg\_Ip.h.

#### 4.53.1.55 STATUS\_BASE

```
#define STATUS_BASE (0x40010000)
```

Definition at line 20 of file eHSM\_Mailbox\_Reg\_Ip.h.

## 4.54 eHSM\_Mgr\_Ctx\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include <string.h>
#include "eHSM_Mgr_Ctx_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Types_Ip.h"
#include "Asr_Standard_Types.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_Exclusive_Area.h"
```

## Functions

- `ehsm_uint32_t eHSM_Mgr_Cipher_Ctx_Get_Free (ehsm_uint32_t *session_id, cipher_session_st **ctx)`
- `cipher_session_st * eHSM_Mgr_Cipher_Ctx_Find (ehsm_uint32_t index)`
- `void eHSM_Mgr_Cipher_Ctx_Free (ehsm_uint32_t index)`
- `Std_HsmReturnTypes ehsm_mgr_ctx_get_free (Crypto_JobType *job, cipher_session_st **ctx)`
- `Std_HsmReturnTypes ehsm_mgr_ctx_find (Crypto_JobType *job, cipher_session_st **ctx)`
- `void ehsm_mgr_ctx_free (Crypto_JobType *job)`
- `ehsm_uint32_t ehsm_cipher_ctx_data_check (ehsm_ctx_block_mgr_st *block_mgr, ehsm_uint8_t **input_data, ehsm_uint32_t *input_sz, ehsm_bool_t keep_last_block)`
- `ehsm_uint32_t ehsm_init_block_mgr (ehsm_ctx_block_mgr_st *block_mgr, ehsm_uint32_t block_sz)`
- `ehsm_uint32_t ehsm_disjunction_updata (ehsm_uint32_t service_id, void *param, ehsm_ctx_block_mgr_st *block_mgr)`
- `ehsm_uint32_t ehsm_disjunction_finish (ehsm_ctx_block_mgr_st *block_mgr, ehsm_addr_t *input_addr, ehsm_uint32_t *input_sz)`

### 4.54.1 Function Documentation

#### 4.54.1.1 ehsm\_cipher\_ctx\_data\_check()

```
ehsm_uint32_t ehsm_cipher_ctx_data_check (
 ehsm_ctx_block_mgr_st * block_mgr,
 ehsm_uint8_t ** input_data,
 ehsm_uint32_t * input_sz,
 ehsm_bool_t keep_last_block)
```

Definition at line 210 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.2 ehsm\_disjunction\_finish()

```
ehsm_uint32_t ehsm_disjunction_finish (
 ehsm_ctx_block_mgr_st * block_mgr,
 ehsm_addr_t * input_addr,
 ehsm_uint32_t * input_sz)
```

Definition at line 362 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.3 ehsm\_disjunction\_updata()

```
ehsm_uint32_t ehsm_disjunction_updata (
 ehsm_uint32_t service_id,
 void * param,
 ehsm_ctx_block_mgr_st * block_mgr)
```

Definition at line 305 of file eHSM\_Mgr\_Ctx\_Ip.c.



#### 4.54.1.4 ehsm\_init\_block\_mgr()

```
ehsm_uint32_t ehsm_init_block_mgr (
 ehsm_ctx_block_mgr_st * block_mgr,
 ehsm_uint32_t block_sz)
```

Definition at line 289 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.5 eHSM\_Mgr\_Cipher\_Ctx\_Find()

```
cipher_session_st* eHSM_Mgr_Cipher_Ctx_Find (
 ehsm_uint32_t index)
```

Definition at line 98 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.6 eHSM\_Mgr\_Cipher\_Ctx\_Free()

```
void eHSM_Mgr_Cipher_Ctx_Free (
 ehsm_uint32_t index)
```

Definition at line 114 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.7 eHSM\_Mgr\_Cipher\_Ctx\_Get\_Free()

```
ehsm_uint32_t eHSM_Mgr_Cipher_Ctx_Get_Free (
 ehsm_uint32_t * session_id,
 cipher_session_st ** ctx)
```

Definition at line 62 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.8 ehsm\_mgr\_ctx\_find()

```
Std_HsmReturnTypes ehsm_mgr_ctx_find (
 Crypto_JobType * job,
 cipher_session_st ** ctx)
```

Definition at line 163 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.9 ehsm\_mgr\_ctx\_free()

```
void ehsm_mgr_ctx_free (
 Crypto_JobType * job)
```

Definition at line 193 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.54.1.10 ehsm\_mgr\_ctx\_get\_free()

```
Std_HsmReturnType ehsm_mgr_ctx_get_free (
 Crypto_JobType * job,
 cipher_session_st ** ctx)
```

Definition at line 127 of file eHSM\_Mgr\_Ctx\_Ip.c.

## 4.55 eHSM\_Mgr\_Ctx\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "Asr_Standard_Types.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
```

### Classes

- struct [ehsm\\_ctx\\_block\\_mgr](#)
- struct [ehsm\\_aead\\_data\\_ptr](#)
- struct [ehsm\\_ctx\\_session\\_st](#)
- struct [cipher\\_session\\_st](#)

### Macros

- #define [EHSM\\_ERR\\_CTX\\_MGR\\_BUFFER\\_DATA\\_VALID](#) (0xdeadbee)
- #define [EHSM\\_ERR\\_CTX\\_MGR\\_DATA\\_NOT\\_READY](#) (0xbadcafe)

### Typedefs

- typedef struct [ehsm\\_ctx\\_block\\_mgr](#) [ehsm\\_ctx\\_block\\_mgr\\_st](#)
- typedef struct [ehsm\\_aead\\_data\\_ptr](#) [ehsm\\_aead\\_data\\_ptr\\_st](#)

### Enumerations

- enum [ehsm\\_asym\\_alg\\_e](#) { [EVITA\\_ASYM\\_SM2](#), [EVITA\\_ASYM\\_ECDSA](#), [EVITA\\_ASYM\\_RSA](#) }

## Functions

- `ehsm_uint32_t eHSM_Mgr_Cipher_Ctx_Get_Free (ehsm_uint32_t *session_id, cipher_session_st **ctx)`
- `cipher_session_st * eHSM_Mgr_Cipher_Ctx_Find (ehsm_uint32_t index)`
- `void eHSM_Mgr_Cipher_Ctx_Free (ehsm_uint32_t index)`
- `Std_HsmReturnType ehsm_mgr_ctx_get_free (Crypto_JobType *job, cipher_session_st **ctx)`
- `Std_HsmReturnType ehsm_mgr_ctx_find (Crypto_JobType *job, cipher_session_st **ctx)`
- `void ehsm_mgr_ctx_free (Crypto_JobType *job)`
- `ehsm_uint32_t ehsm_cipher_ctx_data_check (ehsm_ctx_block_mgr_st *block_mgr, ehsm_uint8_t **input_data, ehsm_uint32_t *input_sz, ehsm_bool_t keep_last_block)`
- `ehsm_uint32_t ehsm_disjunction_update (ehsm_uint32_t service_id, void *param, ehsm_ctx_block_mgr_st *block_mgr)`
- `ehsm_uint32_t ehsm_init_block_mgr (ehsm_ctx_block_mgr_st *block_mgr, ehsm_uint32_t block_sz)`
- `ehsm_uint32_t ehsm_disjunction_finish (ehsm_ctx_block_mgr_st *block_mgr, ehsm_addr_t *input_addr, ehsm_uint32_t *input_sz)`

## 4.55.1 Macro Definition Documentation

### 4.55.1.1 EHSM\_ERR\_CTX\_MGR\_BUFFER\_DATA\_VALID

```
#define EHSM_ERR_CTX_MGR_BUFFER_DATA_VALID (0xdeadbee)
```

Definition at line 25 of file eHSM\_Mgr\_Ctx\_Ip.h.

### 4.55.1.2 EHSM\_ERR\_CTX\_MGR\_DATA\_NOT\_READY

```
#define EHSM_ERR_CTX_MGR_DATA_NOT_READY (0xbadcafe)
```

Definition at line 26 of file eHSM\_Mgr\_Ctx\_Ip.h.

## 4.55.2 Typedef Documentation

### 4.55.2.1 ehsm\_aead\_data\_ptr\_st

```
typedef struct ehsm_aead_data_ptr ehsm_aead_data_ptr_st
```

### 4.55.2.2 ehsm\_ctx\_block\_mgr\_st

```
typedef struct ehsm_ctx_block_mgr ehsm_ctx_block_mgr_st
```

## 4.55.3 Enumeration Type Documentation

### 4.55.3.1 ehsm\_asym\_alg\_e

```
enum ehsm_asym_alg_e
```

## Enumerator

|                  |  |
|------------------|--|
| EVITA_ASYM_SM2   |  |
| EVITA_ASYM_ECDSA |  |
| EVITA_ASYM_RSA   |  |

Definition at line 77 of file eHSM\_Mgr\_Ctx\_Ip.h.

#### 4.55.4 Function Documentation

##### 4.55.4.1 ehsm\_cipher\_ctx\_data\_check()

```
ehsm_uint32_t ehsm_cipher_ctx_data_check (
 ehsm_ctx_block_mgr_st * block_mgr,
 ehsm_uint8_t ** input_data,
 ehsm_uint32_t * input_sz,
 ehsm_bool_t keep_last_block)
```

Definition at line 210 of file eHSM\_Mgr\_Ctx\_Ip.c.

##### 4.55.4.2 ehsm\_disjunction\_finish()

```
ehsm_uint32_t ehsm_disjunction_finish (
 ehsm_ctx_block_mgr_st * block_mgr,
 ehsm_addr_t * input_addr,
 ehsm_uint32_t * input_sz)
```

Definition at line 362 of file eHSM\_Mgr\_Ctx\_Ip.c.

##### 4.55.4.3 ehsm\_disjunction\_updata()

```
ehsm_uint32_t ehsm_disjunction_updata (
 ehsm_uint32_t service_id,
 void * param,
 ehsm_ctx_block_mgr_st * block_mgr)
```

Definition at line 305 of file eHSM\_Mgr\_Ctx\_Ip.c.

##### 4.55.4.4 ehsm\_init\_block\_mgr()

```
ehsm_uint32_t ehsm_init_block_mgr (
 ehsm_ctx_block_mgr_st * block_mgr,
 ehsm_uint32_t block_sz)
```

Definition at line 289 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.55.4.5 eHSM\_Mgr\_Cipher\_Ctx\_Find()

```
cipher_session_st* eHSM_Mgr_Cipher_Ctx_Find (
 ehsm_uint32_t index)
```

Definition at line 98 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.55.4.6 eHSM\_Mgr\_Cipher\_Ctx\_Free()

```
void eHSM_Mgr_Cipher_Ctx_Free (
 ehsm_uint32_t index)
```

Definition at line 114 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.55.4.7 eHSM\_Mgr\_Cipher\_Ctx\_Get\_Free()

```
ehsm_uint32_t eHSM_Mgr_Cipher_Ctx_Get_Free (
 ehsm_uint32_t * session_id,
 cipher_session_st ** ctx)
```

Definition at line 62 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.55.4.8 ehsm\_mgr\_ctx\_find()

```
Std_HsmReturnType ehsm_mgr_ctx_find (
 Crypto_JobType * job,
 cipher_session_st ** ctx)
```

Definition at line 163 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.55.4.9 ehsm\_mgr\_ctx\_free()

```
void ehsm_mgr_ctx_free (
 Crypto_JobType * job)
```

Definition at line 193 of file eHSM\_Mgr\_Ctx\_Ip.c.

#### 4.55.4.10 ehsm\_mgr\_ctx\_get\_free()

```
Std_HsmReturnType ehsm_mgr_ctx_get_free (
 Crypto_JobType * job,
 cipher_session_st ** ctx)
```

Definition at line 127 of file eHSM\_Mgr\_Ctx\_Ip.c.

## 4.56 ehsm\_register\_addr.h File Reference

## 4.57 eHSM\_Srv\_AsymCper\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
```

### Variables

- [ehsm\\_service\\_st srv\\_crypto\\_pke](#)

### 4.57.1 Variable Documentation

#### 4.57.1.1 srv\_crypto\_pke

[ehsm\\_service\\_st](#) `srv_crypto_pke`

#### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_CRYPTO_PKE,
 .reqhdl = _srv_crypto_cipher_reqhdl,
 .rsphdl = _srv_crypto_cipher_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 41 of file eHSM\_Srv\_AsymCper\_Ip.c.

## 4.58 eHSM\_Srv\_Ciper\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
```

## Functions

- [ehsm\\_int32\\_t \\_srv\\_crypto\\_cipher\\_reqhdl](#) (void \*para, ehsm\_cmd\_req\_st \*req)
- [ehsm\\_int32\\_t \\_srv\\_crypto\\_cipher\\_rsphdl](#) (void \*para, ehsm\_cmd\_req\_st \*req)

## Variables

- [ehsm\\_service\\_st srv\\_crypto\\_ske](#)

### 4.58.1 Function Documentation

#### 4.58.1.1 \_srv\_crypto\_cipher\_reqhdl()

```
ehsm_int32_t _srv_crypto_cipher_reqhdl (
 void * para,
 ehsm_cmd_req_st * req)
```

Definition at line 55 of file eHSM\_Srv\_Ciper\_Ip.c.

#### 4.58.1.2 \_srv\_crypto\_cipher\_rsphdl()

```
ehsm_int32_t _srv_crypto_cipher_rsphdl (
 void * para,
 ehsm_cmd_req_st * req)
```

Definition at line 187 of file eHSM\_Srv\_Ciper\_Ip.c.

### 4.58.2 Variable Documentation

#### 4.58.2.1 srv\_crypto\_ske

```
ehsm_service_st srv_crypto_ske
```

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_CRYPTOSKE,
 .reqhdl = _srv_crypto_cipher_reqhdl,
 .rsphdl = _srv_crypto_cipher_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 201 of file eHSM\_Srv\_Ciper\_Ip.c.

## 4.59 eHSM\_Srv\_Cipher\_Ip.h File Reference

```
#include "eHSM_Types_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
```

### Classes

- struct [ehsm\\_cmd](#)

### Typedefs

- typedef struct [ehsm\\_cmd](#) [ehsm\\_cmd\\_cipher\\_with\\_rps\\_st](#)

### Functions

- [ehsm\\_int32\\_t \\_srv\\_crypto\\_cipher\\_reqhdl](#) (void \*para, [ehsm\\_cmd\\_req\\_st](#) \*req)
- [ehsm\\_int32\\_t \\_srv\\_crypto\\_cipher\\_rsphdl](#) (void \*para, [ehsm\\_cmd\\_req\\_st](#) \*req)

#### 4.59.1 Typedef Documentation

##### 4.59.1.1 ehsm\_cmd\_cipher\_with\_rps\_st

```
typedef struct ehsm_cmd ehsm_cmd_cipher_with_rps_st
```

#### 4.59.2 Function Documentation

##### 4.59.2.1 \_srv\_crypto\_cipher\_reqhdl()

```
ehsm_int32_t _srv_crypto_cipher_reqhdl (
 void * para,
 ehsm_cmd_req_st * req)
```

Definition at line 55 of file eHSM\_Srv\_Ciper\_Ip.c.

##### 4.59.2.2 \_srv\_crypto\_cipher\_rsphdl()

```
ehsm_int32_t _srv_crypto_cipher_rsphdl (
 void * para,
 ehsm_cmd_req_st * req)
```

Definition at line 187 of file eHSM\_Srv\_Ciper\_Ip.c.



## 4.60 eHSM\_Srv\_CmdReq\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Compt_List.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
```

### Classes

- struct [ehsm\\_cmd\\_req](#)

### Macros

- #define [MAX\\_RESPONSE\\_DATA\\_SIZE](#) (sizeof([ehsm\\_sm9\\_exchg\\_key\\_cmd\\_child\\_st](#)) + 4U)
- #define [EHSM\\_CMD\\_PRIORITY\\_DEFAULT](#) 10

### Typedefs

- typedef struct [ehsm\\_cmd\\_req](#) [ehsm\\_cmd\\_req\\_st](#)
- typedef [ehsm\\_uint32\\_t](#)(\* [cmd\\_req\\_cb](#)) (void \*para, [ehsm\\_cmd\\_req\\_st](#) \*req)
- typedef void(\* [cmd\\_release\\_cb](#)) ([ehsm\\_cmd\\_req\\_st](#) \*req)

### Enumerations

- enum [ehsm\\_cmd\\_req\\_type\\_e](#) { [EHSM\\_CMD\\_REQ\\_TYPE\\_SYNC](#) = 0, [EHSM\\_CMD\\_REQ\\_TYPE\\_ASYNC](#), [EHSM\\_CMD\\_REQ\\_TYPE\\_NO\\_RSP](#) }
- enum [ehsm\\_cmd\\_req\\_state\\_e](#) { [EHSM\\_CMD\\_REQ\\_STATE\\_IDLE](#) = 0, [EHSM\\_CMD\\_REQ\\_STATE\\_INIT](#), [EHSM\\_CMD\\_REQ\\_STATE\\_PROCESSING](#), [EHSM\\_CMD\\_REQ\\_STATE\\_CANCELED](#), [EHSM\\_CMD\\_REQ\\_STATE\\_DONE](#) }

#### 4.60.1 Macro Definition Documentation

##### 4.60.1.1 EHSM\_CMD\_PRIORITY\_DEFAULT

```
#define EHSM_CMD_PRIORITY_DEFAULT 10
```

Definition at line 25 of file [eHSM\\_Srv\\_CmdReq\\_Ip.h](#).

##### 4.60.1.2 MAX\_RESPONSE\_DATA\_SIZE

```
#define MAX_RESPONSE_DATA_SIZE (sizeof(ehsm_sm9_exchg_key_cmd_child_st) + 4U)
```

Definition at line 22 of file [eHSM\\_Srv\\_CmdReq\\_Ip.h](#).

## 4.60.2 Typedef Documentation

### 4.60.2.1 cmd\_release\_cb

```
typedef void(* cmd_release_cb) (ehsm_cmd_req_st *req)
```

Definition at line 32 of file eHSM\_Srv\_CmdReq\_Ip.h.

### 4.60.2.2 cmd\_req\_cb

```
typedef ehsm_uint32_t(* cmd_req_cb) (void *para, ehsm_cmd_req_st *req)
```

Definition at line 31 of file eHSM\_Srv\_CmdReq\_Ip.h.

### 4.60.2.3 ehsm\_cmd\_req\_st

```
typedef struct ehsm_cmd_req ehsm_cmd_req_st
```

Definition at line 30 of file eHSM\_Srv\_CmdReq\_Ip.h.

## 4.60.3 Enumeration Type Documentation

### 4.60.3.1 ehsm\_cmd\_req\_state\_e

```
enum ehsm_cmd_req_state_e
```

#### Enumerator

|                               |  |
|-------------------------------|--|
| EHSM_CMD_REQ_STATE_IDLE       |  |
| EHSM_CMD_REQ_STATE_INIT       |  |
| EHSM_CMD_REQ_STATE_PROCESSING |  |
| EHSM_CMD_REQ_STATE_CANCELED   |  |
| EHSM_CMD_REQ_STATE_DONE       |  |

Definition at line 41 of file eHSM\_Srv\_CmdReq\_Ip.h.

#### 4.60.3.2 ehsm\_cmd\_req\_type\_e

enum [ehsm\\_cmd\\_req\\_type\\_e](#)

## Enumerator

|                          |  |
|--------------------------|--|
| EHSM_CMD_REQ_TYPE_SYNC   |  |
| EHSM_CMD_REQ_TYPE_ASYNC  |  |
| EHSM_CMD_REQ_TYPE_NO_RSP |  |

Definition at line 34 of file eHSM\_Srv\_CmdReq\_Ip.h.

## 4.61 eHSM\_Srv\_Counter\_Ip.c File Reference

```
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include <string.h>
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Mailbox_CmdId_Ip.h"
```

## 4.62 eHSM\_Srv\_Ext\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include <string.h>
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Mailbox_CmdId_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_If_Ext_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
```

## Variables

- [ehsm\\_service\\_st srv\\_fw\\_encrypt\\_key](#)
- [ehsm\\_service\\_st srv\\_fw\\_get\\_random\\_key](#)
- [ehsm\\_service\\_st srv\\_image\\_upgrade](#)
- [ehsm\\_service\\_st srv\\_image\\_verify](#)
- [ehsm\\_service\\_st g\\_srv\\_soc\\_image\\_verify](#)
- [ehsm\\_service\\_st srv\\_read\\_otp\\_data](#)
- [ehsm\\_service\\_st srv\\_write\\_otp\\_data](#)
- [ehsm\\_service\\_st srv\\_self\\_test](#)
- [ehsm\\_service\\_st srv\\_low\\_power](#)
- [ehsm\\_service\\_st srv\\_set\\_baudrate](#)
- [ehsm\\_service\\_st srv\\_change\\_lifecycle](#)
- [ehsm\\_service\\_st srv\\_change\\_control\\_field](#)

- [ehsm\\_service\\_st srv\\_get\\_she\\_status](#)
- [ehsm\\_service\\_st srv\\_get\\_she\\_id](#)
- [ehsm\\_service\\_st srv\\_get\\_module\\_status](#)
- [ehsm\\_service\\_st srv\\_bootloader\\_cmd](#)
- [ehsm\\_service\\_st srv\\_get\\_challenge](#)
- [ehsm\\_service\\_st srv\\_debug\\_auth](#)
- [ehsm\\_service\\_st srv\\_soc\\_boot\\_status](#)
- [ehsm\\_service\\_st srv\\_close\\_debug](#)
- [ehsm\\_service\\_st srv\\_asr\\_cancle\\_cmd](#)
- [ehsm\\_service\\_st srv\\_she\\_cancle\\_cmd](#)

## 4.62.1 Variable Documentation

### 4.62.1.1 g\_srv\_soc\_image\_verify

`ehsm_service_st g_srv_soc_image_verify`

#### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_SOC_IMAGE_VERIFY,
 .reqhdl = _soc_image_verify_reqhdl,
 .rsphdl = _image_verify_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 295 of file eHSM\_Srv\_Ext\_Ip.c.

### 4.62.1.2 srv\_asr\_cancle\_cmd

`ehsm_service_st srv_asr_cancle_cmd`

#### Initial value:

```
= {
 .service_id = EHSM_SRV_SYS_ASR_CANCEL,
 .reqhdl = _asr_cancle_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 1233 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.3 srv\_bootloader\_cmd

ehsm\_service\_st srv\_bootloader\_cmd

##### Initial value:

```
= {
 .service_id = EHSM_SRV_BOOTLOADER_CMD,
 .reqhdl = _bootloader_cmd_reqhdl,
 .rsphdl = _bootloader_cmd_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 875 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.4 srv\_change\_control\_field

ehsm\_service\_st srv\_change\_control\_field

##### Initial value:

```
= {
 .service_id = EHSM_SRV_CHANGE_CONTROLFIELD,
 .reqhdl = _change_control_field_reqhdl,
 .rsphdl = _change_control_field_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 670 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.5 srv\_change\_lifecycle

ehsm\_service\_st srv\_change\_lifecycle

##### Initial value:

```
= {
 .service_id = EHSM_SRV_CHANGE_LIFECYCLE,
 .reqhdl = _change_lifecycle_reqhdl,
 .rsphdl = _change_lifecycle_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 629 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.6 srv\_close\_debug

ehsm\_service\_st srv\_close\_debug

##### Initial value:

```
= {
 .service_id = EHSM_SRV_SYS_CLOSE_DEBUG,
 .reqhdl = _close_debug_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 1052 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.7 srv\_debug\_auth

ehsm\_service\_st srv\_debug\_auth

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_DEBUG_AUTHENTICATION,
 .reqhdl = _debug_auth_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 967 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.8 srv\_fw\_encrypt\_key

ehsm\_service\_st srv\_fw\_encrypt\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_FW_ENCRYPT_KEY,
 .reqhdl = _fw_encrypt_key_reqhdl,
 .rsphdl = _fw_encrypt_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 122 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.9 srv\_fw\_get\_random\_key

ehsm\_service\_st srv\_fw\_get\_random\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_FW_GET_RANDOM_KEY,
 .reqhdl = _fw_get_random_key_reqhdl,
 .rsphdl = _fw_get_random_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 168 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.10 srv\_get\_challenge

ehsm\_service\_st srv\_get\_challenge

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_GET_CHALLENGE,
 .reqhdl = _get_challenge_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 914 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.11 srv\_get\_module\_status

ehsm\_service\_st srv\_get\_module\_status

##### Initial value:

```
= {
 .service_id = EHSM_SRV_MODULE_STATUS,
 .reqhdl = _module_status_reqhdl,
 .rsphdl = _module_status_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 815 of file eHSM\_Srv\_Ext\_Ip.c.



#### 4.62.1.12 srv\_get\_she\_id

ehsm\_service\_st srv\_get\_she\_id

##### Initial value:

```
= {
 .service_id = EHSM_SRV_GET_SHE_ID,
 .reqhdl = _get_she_id_reqhdl,
 .rsphdl = _get_she_id_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 763 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.13 srv\_get\_she\_status

ehsm\_service\_st srv\_get\_she\_status

##### Initial value:

```
= {
 .service_id = EHSM_SRV_GET_SHE_STATUS,
 .reqhdl = _get_she_status_reqhdl,
 .rsphdl = _get_she_status_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 715 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.14 srv\_image\_upgrade

ehsm\_service\_st srv\_image\_upgrade

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_IMAGE_UPGRADE,
 .reqhdl = _image_upgrade_reqhdl,
 .rsphdl = _image_upgrade_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 212 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.15 srv\_image\_verify

ehsm\_service\_st srv\_image\_verify

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_IMAGE_VERIFY,
 .reqhdl = _image_verify_reqhdl,
 .rsphdl = _image_verify_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 261 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.16 srv\_low\_power

ehsm\_service\_st srv\_low\_power

##### Initial value:

```
= {
 .service_id = EHSM_SRV_LOW_POWER,
 .reqhdl = _low_power_reqhdl,
 .rsphdl = _low_power_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 534 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.17 srv\_read\_otp\_data

ehsm\_service\_st srv\_read\_otp\_data

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_OTP_READ,
 .reqhdl = _read_otp_data_reqhdl,
 .rsphdl = _read_otp_data_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 399 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.18 srv\_self\_test

ehsm\_service\_st srv\_self\_test

##### Initial value:

```
= {
 .service_id = EHSM_SRV_SYS_SELF_TEST,
 .reqhdl = _self_test_reqhdl,
 .rsphdl = _self_test_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 492 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.19 srv\_set\_baudrate

ehsm\_service\_st srv\_set\_baudrate

##### Initial value:

```
= {
 .service_id = EHSM_SRV_SET_BAUDRATE,
 .reqhdl = _set_baudrate_reqhdl,
 .rsphdl = _set_baudrate_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 588 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.20 srv\_she\_cancle\_cmd

ehsm\_service\_st srv\_she\_cancle\_cmd

##### Initial value:

```
= {
 .service_id = EHSM_SRV_SYS_SHE_CANCEL,
 .reqhdl = _she_cancle_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 1261 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.62.1.21 `srv_soc_boot_status`

`ehsm_service_st` `srv_soc_boot_status`

##### Initial value:

```
= {
 .service_id = EHSM_SRV_SOC_BOOT_STATUS,
 .reqhdl = _soc_boot_status_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 1022 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.62.1.22 `srv_write_otp_data`

`ehsm_service_st` `srv_write_otp_data`

##### Initial value:

```
= {
 .service_id = EHSM_SRV_EXTENDED_OTP_WRITE,
 .reqhdl = _write_otp_data_reqhdl,
 .rsphdl = _write_otp_data_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 444 of file `eHSM_Srv_Ext_Ip.c`.

## 4.63 eHSM\_Srv\_Hash\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
```

### Variables

- `ehsm_service_st` `srv_crypto_hash`

#### 4.63.1 Variable Documentation

4.63.1.1 `srv_crypto_hash`

```
ehsm_service_st srv_crypto_hash
```

**Initial value:**

```
= {
 .service_id = EHSM_SRV_EXTENDED_CRYPT_HASH,
 .reqhdl = _srv_crypto_cipher_reqhdl,
 .rsphdl = _srv_crypto_cipher_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 42 of file `eHSM_Srv_Hash_Ip.c`.

## 4.64 eHSM\_Srv\_Key\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include <string.h>
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_If_Ext_Types_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
```

**Macros**

- `#define DEFAULT_RSA_E_SIZE` 17

**Variables**

- `ehsm_service_st srv_create_dh_key`
- `ehsm_service_st srv_create_random_key`
- `ehsm_service_st srv_derive_key`
- `ehsm_service_st srv_export_key`
- `ehsm_service_st srv_get_pub_from_priv`
- `ehsm_service_st srv_import_key`
- `ehsm_service_st srv_key_remove`
- `ehsm_service_st srv_key_copy`
- `ehsm_service_st srv_key_status`
- `ehsm_service_st srv_she_load_key`
- `ehsm_service_st srv_she_load_plain_key`
- `ehsm_service_st srv_she_ram_key_export`
- `ehsm_service_st srv_certificate_parse`
- `ehsm_service_st srv_certificate_verify`

## 4.64.1 Macro Definition Documentation

### 4.64.1.1 DEFAULT\_RSA\_E\_SIZE

```
#define DEFAULT_RSA_E_SIZE 17
```

Definition at line 27 of file eHSM\_Srv\_Key\_Ip.c.

## 4.64.2 Variable Documentation

### 4.64.2.1 srv\_certificate\_parse

```
ehsm_service_st srv_certificate_parse
```

**Initial value:**

```
= {
 .service_id = EHSM_SRV_KEYMGR_CERRITIFATEPARSE,
 .reqhdl = _certificate_parse_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 864 of file eHSM\_Srv\_Key\_Ip.c.

### 4.64.2.2 srv\_certificate\_verify

```
ehsm_service_st srv_certificate_verify
```

**Initial value:**

```
= {
 .service_id = EHSM_SRV_KEYMGR_CERRITIFATEVERIFY,
 .reqhdl = _certificate_verify_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 898 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.3 srv\_create\_dh\_key

ehsm\_service\_st srv\_create\_dh\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEYEXCHANGEALCSECRET,
 .reqhdl = _create_dh_key_reqhdl,
 .rsphdl = _create_dh_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 85 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.4 srv\_create\_random\_key

ehsm\_service\_st srv\_create\_random\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEYGENERATE,
 .reqhdl = _create_random_key_reqhdl,
 .rsphdl = _create_random_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_RSA_K_CMD_TIMEOUT
}
```

Definition at line 178 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.5 srv\_derive\_key

ehsm\_service\_st srv\_derive\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_CREATE_DERIVED_KEY,
 .reqhdl = _derive_key_reqhdl,
 .rsphdl = _derive_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 318 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.6 srv\_export\_key

ehsm\_service\_st srv\_export\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEY_EXPORT,
 .reqhdl = _export_key_reqhdl,
 .rsphdl = _export_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 406 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.7 srv\_get\_pub\_from\_priv

ehsm\_service\_st srv\_get\_pub\_from\_priv

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEYEXCHANGECPUBVAL,
 .reqhdl = _get_pub_from_priv_reqhdl,
 .rsphdl = _get_pub_from_priv_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 482 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.8 srv\_import\_key

ehsm\_service\_st srv\_import\_key

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEY_IMPORT,
 .reqhdl = _import_key_reqhdl,
 .rsphdl = _import_key_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 548 of file eHSM\_Srv\_Key\_Ip.c.



#### 4.64.2.9 srv\_key\_copy

ehsm\_service\_st srv\_key\_copy

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_COPY_KEY,
 .reqhdl = _key_copy_reqhdl,
 .rsphdl = _key_copy_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 655 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.10 srv\_key\_remove

ehsm\_service\_st srv\_key\_remove

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEY_REMOVE,
 .reqhdl = _key_remove_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 621 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.11 srv\_key\_status

ehsm\_service\_st srv\_key\_status

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_KEY_STATUS,
 .reqhdl = _key_status_reqhdl,
 .rsphdl = _key_status_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 706 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.12 srv\_she\_load\_key

`ehsm_service_st` `srv_she_load_key`

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_LOAD_KEY,
 .reqhdl = _she_load_key_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 767 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.13 srv\_she\_load\_plain\_key

`ehsm_service_st` `srv_she_load_plain_key`

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_LOAD_PLAIN_KEY,
 .reqhdl = _she_load_plain_key_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 801 of file eHSM\_Srv\_Key\_Ip.c.

#### 4.64.2.14 srv\_she\_ram\_key\_export

`ehsm_service_st` `srv_she_ram_key_export`

##### Initial value:

```
= {
 .service_id = EHSM_SRV_KEYMGR_EXPORT_RAM_KEY,
 .reqhdl = _she_ram_key_export_reqhdl,
 .rsphdl = _general_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 827 of file eHSM\_Srv\_Key\_Ip.c.

## 4.65 eHSM\_Srv\_Mgr\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include <string.h>
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Dspt_lp.h"
#include "eHSM_Compt_Bitmap.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
#include "eHSM_Exclusive_Area.h"
#include "eHSM_Mailbox_Ip.h"
```

## Classes

- struct [ehsm\\_cmd\\_req\\_buffer](#)

## Macros

- #define [COMMAND\\_REQ\\_QUANTITY](#)

## Typedefs

- typedef struct [ehsm\\_cmd\\_req\\_buffer](#) [ehsm\\_cmd\\_req\\_buffer\\_st](#)

## Functions

- void [hw\\_interrupt\\_disable](#) ()
- void [hw\\_interrupt\\_enable](#) ()
- void [ehsm\\_set\\_address\\_pointer](#) ([ehsm\\_uint8\\_t](#) \*addr\_array, const [ehsm\\_uint8\\_t](#) \*data\_pointer)
- [ehsm\\_uint32\\_t](#) [ehsm\\_process\\_asr\\_service](#) ([ehsm\\_uint32\\_t](#) service\_id, void \*param, [ehsm\\_service\\_info\\_st](#) \*service\_info)
- [ehsm\\_uint32\\_t](#) [ehsm\\_register\\_service](#) ([ehsm\\_service\\_st](#) \*service)
- [ehsm\\_uint32\\_t](#) [ehsm\\_service\\_init](#) (void)
- [ehsm\\_uint32\\_t](#) [ehsm\\_cancel\\_single\\_service](#) (void \*service\_ctx)
- [ehsm\\_uint32\\_t](#) [ehsm\\_process\\_sync\\_service](#) ([ehsm\\_uint32\\_t](#) service\_id, void \*param, [ehsm\\_api\\_type\\_e](#) api\_type)

## Variables

- [ehsm\\_service\\_st](#) [srv\\_crypto\\_randomgenerate](#)
- [ehsm\\_service\\_st](#) [srv\\_crypto\\_ske](#)
- [ehsm\\_service\\_st](#) [srv\\_crypto\\_hash](#)
- [ehsm\\_service\\_st](#) [srv\\_crypto\\_pke](#)
- [ehsm\\_service\\_st](#) [srv\\_she\\_load\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_she\\_load\\_plain\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_she\\_ram\\_key\\_export](#)
- [ehsm\\_service\\_st](#) [srv\\_get\\_she\\_status](#)
- [ehsm\\_service\\_st](#) [srv\\_get\\_she\\_id](#)
- [ehsm\\_service\\_st](#) [srv\\_she\\_cancel\\_cmd](#)
- [ehsm\\_service\\_st](#) [srv\\_key\\_copy](#)
- [ehsm\\_service\\_st](#) [srv\\_asr\\_cancel\\_cmd](#)
- [ehsm\\_service\\_st](#) [srv\\_certificate\\_parse](#)
- [ehsm\\_service\\_st](#) [srv\\_certificate\\_verify](#)
- [ehsm\\_service\\_st](#) [srv\\_create\\_random\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_derive\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_create\\_dh\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_export\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_get\\_pub\\_from\\_priv](#)
- [ehsm\\_service\\_st](#) [srv\\_import\\_key](#)
- [ehsm\\_service\\_st](#) [srv\\_key\\_remove](#)
- [ehsm\\_service\\_st](#) [srv\\_key\\_status](#)
- [ehsm\\_service\\_st](#) [srv\\_get\\_module\\_status](#)
- [ehsm\\_service\\_st](#) [srv\\_get\\_challenge](#)
- [ehsm\\_service\\_st](#) [srv\\_debug\\_auth](#)
- [ehsm\\_service\\_st](#) [srv\\_close\\_debug](#)
- [ehsm\\_service\\_st](#) [srv\\_image\\_upgrade](#)

- [ehsm\\_service\\_st srv\\_image\\_verify](#)
- [ehsm\\_service\\_st g\\_srv\\_soc\\_image\\_verify](#)
- [ehsm\\_service\\_st srv\\_low\\_power](#)
- [ehsm\\_service\\_st srv\\_set\\_baudrate](#)
- [ehsm\\_service\\_st srv\\_soc\\_boot\\_status](#)
- [ehsm\\_service\\_st srv\\_read\\_otp\\_data](#)
- [ehsm\\_service\\_st srv\\_write\\_otp\\_data](#)
- [ehsm\\_service\\_st srv\\_fw\\_get\\_random\\_key](#)
- [ehsm\\_service\\_st srv\\_fw\\_encrypt\\_key](#)
- [ehsm\\_service\\_st srv\\_change\\_lifecycle](#)
- [ehsm\\_service\\_st srv\\_change\\_control\\_field](#)
- [ehsm\\_service\\_st srv\\_bootloader\\_cmd](#)
- [ehsm\\_service\\_st \\* service\\_table \[EHSM\\_SRV\\_END\] = {NULL}](#)
- [ehsm\\_cmd\\_req\\_buffer\\_st \\* g\\_cmd\\_req\\_buffer = NULL](#)

## 4.65.1 Macro Definition Documentation

### 4.65.1.1 COMMAND\_REQ\_QUANTITY

```
#define COMMAND_REQ_QUANTITY
```

**Value:**

```
(CONFIG_EHSM_ARCH_V_CRYPTOBJ_SKE_QUEUE_SIZE +
 CONFIG_EHSM_ARCH_V_CRYPTOBJ_PKE_QUEUE_SIZE + \
 CONFIG_EHSM_ARCH_V_CRYPTOBJ_TRNG_QUEUE_SIZE +
 CONFIG_EHSM_ARCH_V_CRYPTOBJ_HASH_QUEUE_SIZE + \
 CONFIG_EHSM_ARCH_V_CRYPTOBJ_K_QUEUE_SIZE +
 CONFIG_EHSM_ARCH_V_CRYPTOBJ_SYSMGR_QUEUE_SIZE + \
 CONFIG_EHSM_ARCH_V_CMD_QUEUE_SIZE *
 CRYPTO_OBJECT_TYPE_MAX)
```

Definition at line 25 of file eHSM\_Srv\_Mgr\_Ip.c.

## 4.65.2 Typedef Documentation

### 4.65.2.1 ehsm\_cmd\_req\_buffer\_st

```
typedef struct ehsm_cmd_req_buffer ehsm_cmd_req_buffer_st
```

## 4.65.3 Function Documentation

#### 4.65.3.1 ehsm\_cancel\_single\_service()

```
ehsm_uint32_t ehsm_cancel_single_service (
 void * service_ctx)
```

Definition at line 426 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.3.2 ehsm\_process\_asr\_service()

```
ehsm_uint32_t ehsm_process_asr_service (
 ehsm_uint32_t service_id,
 void * param,
 ehsm_service_info_st * service_info)
```

Definition at line 252 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.3.3 ehsm\_process\_sync\_service()

```
ehsm_uint32_t ehsm_process_sync_service (
 ehsm_uint32_t service_id,
 void * param,
 ehsm_api_type_e api_type)
```

Definition at line 467 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.3.4 ehsm\_register\_service()

```
ehsm_uint32_t ehsm_register_service (
 ehsm_service_st * service)
```

Definition at line 331 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.3.5 ehsm\_service\_init()

```
ehsm_uint32_t ehsm_service_init (
 void)
```

Definition at line 343 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.3.6 ehsm\_set\_address\_pointer()

```
void ehsm_set_address_pointer (
 ehsm_uint8_t * addr_array,
 const ehsm_uint8_t * data_pointer)
```

Definition at line 246 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.3.7 hw\_interrupt\_disable()

```
void hw_interrupt_disable ()
```

#### 4.65.3.8 hw\_interrupt\_enable()

```
void hw_interrupt_enable ()
```

### 4.65.4 Variable Documentation

#### 4.65.4.1 g\_cmd\_req\_buffer

```
ehsm_cmd_req_buffer_st* g_cmd_req_buffer = NULL
```

Definition at line 124 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.4.2 g\_srv\_soc\_image\_verify

```
ehsm_service_st g_srv_soc_image_verify
```

Definition at line 295 of file eHSM\_Srv\_Ext\_Ip.c.

#### 4.65.4.3 service\_table

```
ehsm_service_st* service_table[EHSM_SRV_END] = {NULL}
```

Definition at line 119 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.65.4.4 `srv_asr_cancle_cmd`

`ehsm_service_st` `srv_asr_cancle_cmd`

Definition at line 1233 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.5 `srv_bootloader_cmd`

`ehsm_service_st` `srv_bootloader_cmd`

Definition at line 875 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.6 `srv_certificate_parse`

`ehsm_service_st` `srv_certificate_parse`

Definition at line 864 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.7 `srv_certificate_verify`

`ehsm_service_st` `srv_certificate_verify`

Definition at line 898 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.8 `srv_change_control_field`

`ehsm_service_st` `srv_change_control_field`

Definition at line 670 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.9 `srv_change_lifecycle`

`ehsm_service_st` `srv_change_lifecycle`

Definition at line 629 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.10 `srv_close_debug`

`ehsm_service_st` `srv_close_debug`

Definition at line 1052 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.11 `srv_create_dh_key`

`ehsm_service_st` `srv_create_dh_key`

Definition at line 85 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.12 `srv_create_random_key`

`ehsm_service_st` `srv_create_random_key`

Definition at line 178 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.13 `srv_crypto_hash`

`ehsm_service_st` `srv_crypto_hash`

Definition at line 42 of file `eHSM_Srv_Hash_Ip.c`.

#### 4.65.4.14 `srv_crypto_pke`

`ehsm_service_st` `srv_crypto_pke`

Definition at line 41 of file `eHSM_Srv_AsymCper_Ip.c`.

#### 4.65.4.15 `srv_crypto_randomgenerate`

`ehsm_service_st` `srv_crypto_randomgenerate`

Definition at line 42 of file `eHSM_Srv_Rng_Ip.c`.



#### 4.65.4.16 `srv_crypto_ske`

`ehsm_service_st` `srv_crypto_ske`

Definition at line 201 of file `eHSM_Srv_Ciper_Ip.c`.

#### 4.65.4.17 `srv_debug_auth`

`ehsm_service_st` `srv_debug_auth`

Definition at line 967 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.18 `srv_derive_key`

`ehsm_service_st` `srv_derive_key`

Definition at line 49 of file `eHSM_Srv_Mgr_Ip.c`.

#### 4.65.4.19 `srv_export_key`

`ehsm_service_st` `srv_export_key`

Definition at line 406 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.20 `srv_fw_encrypt_key`

`ehsm_service_st` `srv_fw_encrypt_key`

Definition at line 122 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.21 `srv_fw_get_random_key`

`ehsm_service_st` `srv_fw_get_random_key`

Definition at line 168 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.22 `srv_get_challenge`

`ehsm_service_st` `srv_get_challenge`

Definition at line 914 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.23 `srv_get_module_status`

`ehsm_service_st` `srv_get_module_status`

Definition at line 815 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.24 `srv_get_pub_from_priv`

`ehsm_service_st` `srv_get_pub_from_priv`

Definition at line 482 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.25 `srv_get_she_id`

`ehsm_service_st` `srv_get_she_id`

Definition at line 763 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.26 `srv_get_she_status`

`ehsm_service_st` `srv_get_she_status`

Definition at line 715 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.27 `srv_image_upgrade`

`ehsm_service_st` `srv_image_upgrade`

Definition at line 212 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.28 `srv_image_verify`

`ehsm_service_st` `srv_image_verify`

Definition at line 261 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.29 `srv_import_key`

`ehsm_service_st` `srv_import_key`

Definition at line 548 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.30 `srv_key_copy`

`ehsm_service_st` `srv_key_copy`

Definition at line 655 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.31 `srv_key_remove`

`ehsm_service_st` `srv_key_remove`

Definition at line 621 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.32 `srv_key_status`

`ehsm_service_st` `srv_key_status`

Definition at line 706 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.33 `srv_low_power`

`ehsm_service_st` `srv_low_power`

Definition at line 534 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.34 `srv_read_otp_data`

`ehsm_service_st` `srv_read_otp_data`

Definition at line 399 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.35 `srv_set_baudrate`

`ehsm_service_st` `srv_set_baudrate`

Definition at line 588 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.36 `srv_she_cancle_cmd`

`ehsm_service_st` `srv_she_cancle_cmd`

Definition at line 1261 of file `eHSM_Srv_Ext_Ip.c`.

#### 4.65.4.37 `srv_she_load_key`

`ehsm_service_st` `srv_she_load_key`

Definition at line 767 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.38 `srv_she_load_plain_key`

`ehsm_service_st` `srv_she_load_plain_key`

Definition at line 801 of file `eHSM_Srv_Key_Ip.c`.

#### 4.65.4.39 `srv_she_ram_key_export`

`ehsm_service_st` `srv_she_ram_key_export`

Definition at line 827 of file `eHSM_Srv_Key_Ip.c`.

4.65.4.40 `srv_soc_boot_status`

`ehsm_service_st` `srv_soc_boot_status`

Definition at line 1022 of file `eHSM_Srv_Ext_Ip.c`.

4.65.4.41 `srv_write_otp_data`

`ehsm_service_st` `srv_write_otp_data`

Definition at line 444 of file `eHSM_Srv_Ext_Ip.c`.

## 4.66 eHSM\_Srv\_Mgr\_Ip.h File Reference

```
#include "eHSM_Srv_CmdReq_Ip.h"
```

### Classes

- struct `ehsm_service`
- struct `ehsm_service_info`

### Typedefs

- typedef int(\* `service_reqhdl`) (void \*para, `ehsm_cmd_req_st` \*req)
- typedef int(\* `service_rsphdl`) (void \*para, `ehsm_cmd_req_st` \*req)
- typedef struct `ehsm_service` `ehsm_service_st`
- typedef struct `ehsm_service_info` `ehsm_service_info_st`

### Enumerations

- enum `ehsm_cmd_ext_type_e` {  
`EHSM_SRV_START`, `EHSM_SRV_CRYPTO_RANDOMGENERATE`, `EHSM_SRV_CRYPTO_RANDOMSEED`,  
`EHSM_SRV_KEYMGR_KEYGENERATE`,  
`EHSM_SRV_KEYMGR_CREATE_DERIVED_KEY`, `EHSM_SRV_KEYMGR_KEYEXCHANGEALCPUBVAL`, `EHSM_SRV_KEYMGR_KEYEXCHANGEALCSECRET`, `EHSM_SRV_KEYMGR_CERRITIFATEPARSE`,  
`EHSM_SRV_KEYMGR_CERRITIFATEVERIFY`, `EHSM_SRV_KEYMGR_KEYSETVALID`, `EHSM_SRV_KEYMGR_KEY_IMPORT`, `EHSM_SRV_KEYMGR_KEY_EXPORT`,  
`EHSM_SRV_KEYMGR_COPY_KEY`, `EHSM_SRV_KEYMGR_KEYELEMENTIDSGET`, `EHSM_SRV_KEYMGR_KEYSETINVALID`, `EHSM_SRV_KEYMGR_KEY_REMOVE`,  
`EHSM_SRV_KEYMGR_KEY_STATUS`, `EHSM_SRV_KEYMGR_LOAD_KEY`, `EHSM_SRV_KEYMGR_LOAD_PLAIN_KEY`, `EHSM_SRV_KEYMGR_EXPORT_RAM_KEY`,  
`EHSM_SRV_KEYMGR_GENERATER_SM9_KEY`, `EHSM_SRV_KEYMGR_EXCHG_SM9_KEY`, `EHSM_SRV_KEYMGR_SM9_GEN_TMP_KEY`, `EHSM_SRV_KEYMGR_SM9_WRAP_KEY`,  
`EHSM_SRV_KEYMGR_SM9_UNWRAP_KEY`, `EHSM_SRV_KEYMGR_SM9_EXPORT_KEY`, `EHSM_SRV_KEYMGR_SM9_IMPORT_KEY`, `EHSM_SRV_KEYMGR_SM9_GEN_MAST_PUBKEY`,  
`EHSM_SRV_KEYMGR_SM9_GEN_TMP_PUBKEY`, `EHSM_SRV_KEYMGR_SM9_REMOVE_KEY`, `EHSM_SRV_V_RNG_INIT`, `EHSM_SRV_RNG_EXTEND_SEED`,  
`EHSM_SRV_ADMIN_SECURE_BOOT`, `EHSM_SRV_ADMIN_BOOT_STATUS`, `EHSM_SRV_ADMIN_UPGRADE`,  
`EHSM_SRV_ADMIN_VERIFY_PROGRAM`,

```

EHSM_SRV_DEBUG_GET_CHALLENGE, EHSM_SRV_DEBUG_AUTH, EHSM_SRV_DEBUG_SHE_AUTH, E↵
HSM_SRV_TIMER,
EHSM_SRV_COUNTER, EHSM_SRV_SYS_SELF_TEST, EHSM_SRV_SYS_ASR_CANCEL, EHSM_SRV_SY↵
S_SHE_CANCEL,
EHSM_SRV_SYS_GET_STATUS, EHSM_SRV_SYS_GET_ID, EHSM_SRV_SYS_HASH_FINISH_EXTEND, E↵
HSM_SRV_EXTENDED_FW_GET_RANDOM_KEY,
EHSM_SRV_EXTENDED_FW_ENCRYPT_KEY, EHSM_SRV_EXTENDED_GET_CHALLENGE, EHSM_SRV_↵
EXTENDED_DEBUG_AUTHENTICATION, EHSM_SRV_EXTENDED_IMAGE_UPGRADE,
EHSM_SRV_EXTENDED_IMAGE_VERIFY, EHSM_SRV_EXTENDED_SOC_IMAGE_VERIFY, EHSM_SRV_E↵
XTENDED_CRYPTO_SKE, EHSM_SRV_EXTENDED_CRYPTO_PKE,
EHSM_SRV_EXTENDED_CRYPTO_HASH, EHSM_SRV_EXTENDED_OTP_READ, EHSM_SRV_EXTENDED↵
_OTP_WRITE, EHSM_SRV_DEBUG_LOAD_KEY,
EHSM_SRV_LOW_POWER, EHSM_SRV_SET_BAUDRATE, EHSM_SRV_CHANGE_LIFECYCLE, EHSM_SR↵
V_CHANGE_CONTROLFIELD,
EHSM_SRV_GET_SHE_STATUS, EHSM_SRV_GET_SHE_ID, EHSM_SRV_GET_EMU_STATUS, EHSM_SR↵
V_MODULE_STATUS,
EHSM_SRV_BOOTLOADER_CMD, EHSM_SRV_SOC_BOOT_STATUS, EHSM_SRV_SYS_CLOSE_DEBUG,
EHSM_SRV_SENSOR_RESP_INIT,
EHSM_SRV_END }

```

## Functions

- `ehsm_uint32_t ehsm_process_asr_service` (`ehsm_uint32_t service_id`, `void *param`, `ehsm_service_info_↵`  
`st *service_info`)
- `ehsm_uint32_t ehsm_process_sync_service` (`ehsm_uint32_t service_id`, `void *param`, `ehsm_api_type_e api_type`)
- `ehsm_uint32_t ehsm_register_service` (`ehsm_service_st *service`)
- `ehsm_uint32_t ehsm_service_init` (`void`)
- `void ehsm_set_address_pointer` (`ehsm_uint8_t *addr_array`, `const ehsm_uint8_t *addr`)
- `ehsm_uint32_t ehsm_cancel_single_service` (`void *service_ctx`)

### 4.66.1 Typedef Documentation

#### 4.66.1.1 ehsm\_service\_info\_st

```
typedef struct ehsm_service_info ehsm_service_info_st
```

#### 4.66.1.2 ehsm\_service\_st

```
typedef struct ehsm_service ehsm_service_st
```

#### 4.66.1.3 service\_reqhdl

```
typedef int(* service_reqhdl) (void *para, ehsm_cmd_req_st *req)
```

Definition at line 22 of file eHSM\_Srv\_Mgr\_Ip.h.

## 4.66.1.4 service\_rsphdl

```
typedef int(* service_rsphdl) (void *para, ehsm_cmd_req_st *req)
```

Definition at line 23 of file eHSM\_Srv\_Mgr\_Ip.h.

## 4.66.2 Enumeration Type Documentation

## 4.66.2.1 ehsm\_cmd\_ext\_type\_e

```
enum ehsm_cmd_ext_type_e
```

## Enumerator

|                                      |  |
|--------------------------------------|--|
| EHSM_SRV_START                       |  |
| EHSM_SRV_CRYPTO_RANDOMGENERATE       |  |
| EHSM_SRV_CRYPTO_RANDOMSEED           |  |
| EHSM_SRV_KEYMGR_KEYGENERATE          |  |
| EHSM_SRV_KEYMGR_CREATE_DERIVED_KEY   |  |
| EHSM_SRV_KEYMGR_KEYEXCHANGEALCPUBVAL |  |
| EHSM_SRV_KEYMGR_KEYEXCHANGEALCSECRET |  |
| EHSM_SRV_KEYMGR_CERRITIFATEPARSE     |  |
| EHSM_SRV_KEYMGR_CERRITIFATEVERIFY    |  |
| EHSM_SRV_KEYMGR_KEYSETVALID          |  |
| EHSM_SRV_KEYMGR_KEY_IMPORT           |  |
| EHSM_SRV_KEYMGR_KEY_EXPORT           |  |
| EHSM_SRV_KEYMGR_COPY_KEY             |  |
| EHSM_SRV_KEYMGR_KEYELEMENTIDSGET     |  |
| EHSM_SRV_KEYMGR_KEYSETINVALID        |  |
| EHSM_SRV_KEYMGR_KEY_REMOVE           |  |
| EHSM_SRV_KEYMGR_KEY_STATUS           |  |
| EHSM_SRV_KEYMGR_LOAD_KEY             |  |
| EHSM_SRV_KEYMGR_LOAD_PLAIN_KEY       |  |
| EHSM_SRV_KEYMGR_EXPORT_RAM_KEY       |  |
| EHSM_SRV_KEYMGR_GENERATER_SM9_KEY    |  |
| EHSM_SRV_KEYMGR_EXCHG_SM9_KEY        |  |
| EHSM_SRV_KEYMGR_SM9_GEN_TMP_KEY      |  |
| EHSM_SRV_KEYMGR_SM9_WRAP_KEY         |  |
| EHSM_SRV_KEYMGR_SM9_UNWRAP_KEY       |  |
| EHSM_SRV_KEYMGR_SM9_EXPORT_KEY       |  |
| EHSM_SRV_KEYMGR_SM9_IMPORT_KEY       |  |
| EHSM_SRV_KEYMGR_SM9_GEN_MAST_PUBKEY  |  |
| EHSM_SRV_KEYMGR_SM9_GEN_TMP_PUBKEY   |  |
| EHSM_SRV_KEYMGR_SM9_REMOVE_KEY       |  |
| EHSM_SRV_RNG_INIT                    |  |
| EHSM_SRV_RNG_EXTEND_SEED             |  |
| EHSM_SRV_ADMIN_SECURE_BOOT           |  |
| EHSM_SRV_ADMIN_BOOT_STATUS           |  |
| EHSM_SRV_ADMIN_UPGRADE               |  |
| EHSM_SRV_ADMIN_VERIFY_PROGRAM        |  |

## Enumerator

|                                        |  |
|----------------------------------------|--|
| EHSM_SRV_DEBUG_GET_CHALLENGE           |  |
| EHSM_SRV_DEBUG_AUTH                    |  |
| EHSM_SRV_DEBUG_SHE_AUTH                |  |
| EHSM_SRV_TIMER                         |  |
| EHSM_SRV_COUNTER                       |  |
| EHSM_SRV_SYS_SELF_TEST                 |  |
| EHSM_SRV_SYS_ASR_CANCEL                |  |
| EHSM_SRV_SYS_SHE_CANCEL                |  |
| EHSM_SRV_SYS_GET_STATUS                |  |
| EHSM_SRV_SYS_GET_ID                    |  |
| EHSM_SRV_SYS_HASH_FINISH_EXTEND        |  |
| EHSM_SRV_EXTENDED_FW_GET_RANDOM_KEY    |  |
| EHSM_SRV_EXTENDED_FW_ENCRYPT_KEY       |  |
| EHSM_SRV_EXTENDED_GET_CHALLENGE        |  |
| EHSM_SRV_EXTENDED_DEBUG_AUTHENTICATION |  |
| EHSM_SRV_EXTENDED_IMAGE_UPGRADE        |  |
| EHSM_SRV_EXTENDED_IMAGE_VERIFY         |  |
| EHSM_SRV_EXTENDED_SOC_IMAGE_VERIFY     |  |
| EHSM_SRV_EXTENDED_CRYPTO_SKE           |  |
| EHSM_SRV_EXTENDED_CRYPTO_PKE           |  |
| EHSM_SRV_EXTENDED_CRYPTO_HASH          |  |
| EHSM_SRV_EXTENDED_OTP_READ             |  |
| EHSM_SRV_EXTENDED_OTP_WRITE            |  |
| EHSM_SRV_DEBUG_LOAD_KEY                |  |
| EHSM_SRV_LOW_POWER                     |  |
| EHSM_SRV_SET_BAUDRATE                  |  |
| EHSM_SRV_CHANGE_LIFECYCLE              |  |
| EHSM_SRV_CHANGE_CONTROLFIELD           |  |
| EHSM_SRV_GET_SHE_STATUS                |  |
| EHSM_SRV_GET_SHE_ID                    |  |
| EHSM_SRV_GET_EMU_STATUS                |  |
| EHSM_SRV_MODULE_STATUS                 |  |
| EHSM_SRV_BOOTLOADER_CMD                |  |
| EHSM_SRV_SOC_BOOT_STATUS               |  |
| EHSM_SRV_SYS_CLOSE_DEBUG               |  |
| EHSM_SRV_SENSOR_RESP_INIT              |  |
| EHSM_SRV_END                           |  |

Definition at line 25 of file eHSM\_Srv\_Mgr\_Ip.h.

### 4.66.3 Function Documentation

#### 4.66.3.1 ehsm\_cancel\_single\_service()

```
ehsm_uint32_t ehsm_cancel_single_service (
 void * service_ctx)
```

Definition at line 426 of file eHSM\_Srv\_Mgr\_Ip.c.



#### 4.66.3.2 ehsm\_process\_asr\_service()

```
ehsm_uint32_t ehsm_process_asr_service (
 ehsm_uint32_t service_id,
 void * param,
 ehsm_service_info_st * service_info)
```

Definition at line 252 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.66.3.3 ehsm\_process\_sync\_service()

```
ehsm_uint32_t ehsm_process_sync_service (
 ehsm_uint32_t service_id,
 void * param,
 ehsm_api_type_e api_type)
```

Definition at line 467 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.66.3.4 ehsm\_register\_service()

```
ehsm_uint32_t ehsm_register_service (
 ehsm_service_st * service)
```

Definition at line 331 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.66.3.5 ehsm\_service\_init()

```
ehsm_uint32_t ehsm_service_init (
 void)
```

Definition at line 343 of file eHSM\_Srv\_Mgr\_Ip.c.

#### 4.66.3.6 ehsm\_set\_address\_pointer()

```
void ehsm_set_address_pointer (
 ehsm_uint8_t * addr_array,
 const ehsm_uint8_t * addr)
```

Definition at line 246 of file eHSM\_Srv\_Mgr\_Ip.c.

## 4.67 eHSM\_Srv\_Rng\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Srv_Mgr_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Srv_CmdReq_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_Srv_Cipher_Ip.h"
#include "eHSM_If_Asr_KeyCfg_Ip.h"
```

### Variables

- [ehsm\\_service\\_st srv\\_crypto\\_randomgenerate](#)
- [ehsm\\_service\\_st srv\\_rng\\_init](#)
- [ehsm\\_service\\_st srv\\_rng\\_extend\\_seed](#)
- [ehsm\\_service\\_st srv\\_crypto\\_randomseed](#)

### 4.67.1 Variable Documentation

#### 4.67.1.1 srv\_crypto\_randomgenerate

[ehsm\\_service\\_st](#) [srv\\_crypto\\_randomgenerate](#)

##### Initial value:

```
= {
 .service_id = EHSM_SRV_CRYPTO_RANDOMGENERATE,
 .reqhdl = _srv_crypto_cipher_reqhdl,
 .rsphdl = _srv_crypto_cipher_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 42 of file eHSM\_Srv\_Rng\_Ip.c.

#### 4.67.1.2 srv\_crypto\_randomseed

[ehsm\\_service\\_st](#) [srv\\_crypto\\_randomseed](#)

##### Initial value:

```
= {
 .service_id = EHSM_SRV_CRYPTO_RANDOMSEED,
 .reqhdl = _srv_crypto_randomseed_reqhdl,
 .rsphdl = _srv_crypto_randomseed_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 98 of file eHSM\_Srv\_Rng\_Ip.c.

#### 4.67.1.3 srv\_rng\_extend\_seed

ehsm\_service\_st srv\_rng\_extend\_seed

##### Initial value:

```
= {
 .service_id = EHSM_SRV_RNG_EXTEND_SEED,
 .reqhdl = _rng_extend_seed_reqhdl,
 .rsphdl = _rng_extend_seed_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 78 of file eHSM\_Srv\_Rng\_Ip.c.

#### 4.67.1.4 srv\_rng\_init

ehsm\_service\_st srv\_rng\_init

##### Initial value:

```
= {
 .service_id = EHSM_SRV_RNG_INIT,
 .reqhdl = _rng_init_reqhdl,
 .rsphdl = _rng_init_rsphdl,
 .timeout = CONFIG_EHSM_ARCH_V_DEFAULT_CMD_TIMEOUT
}
```

Definition at line 59 of file eHSM\_Srv\_Rng\_Ip.c.

## 4.68 eHSM\_Srv\_Timer\_Ip.c File Reference

```
#include "eHSM_Config_Ip.h"
#include "eHSM_IntCfg_Ip.h"
```

## 4.69 eHSM\_Types\_Ip.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
```

### Macros

- #define false 0
- #define true 1
- #define NULL ((void \*)0)
- #define EHSM\_ARRAY\_SIZE(a) (sizeof(a) / sizeof((a)[0]))

## Typedefs

- typedef signed char [ehsm\\_int8\\_t](#)
- typedef unsigned char [ehsm\\_uint8\\_t](#)
- typedef int [ehsm\\_int32\\_t](#)
- typedef unsigned int [ehsm\\_uint32\\_t](#)
- typedef short [ehsm\\_int16\\_t](#)
- typedef unsigned short [ehsm\\_uint16\\_t](#)
- typedef long long int [ehsm\\_int64\\_t](#)
- typedef unsigned long long int [ehsm\\_uint64\\_t](#)
- typedef signed char [ehsm\\_bool\\_t](#)
- typedef [ehsm\\_uint32\\_t](#) [ehsm\\_addr\\_t](#)
- typedef unsigned int [ehsm\\_handle\\_t](#)

## 4.69.1 Macro Definition Documentation

### 4.69.1.1 EHSM\_ARRAY\_SIZE

```
#define EHSM_ARRAY_SIZE(
 a) (sizeof(a) / sizeof((a)[0]))
```

Definition at line 27 of file eHSM\_Types\_Ip.h.

### 4.69.1.2 false

```
#define false 0
```

Definition at line 16 of file eHSM\_Types\_Ip.h.

### 4.69.1.3 NULL

```
#define NULL ((void *)0)
```

Definition at line 24 of file eHSM\_Types\_Ip.h.

### 4.69.1.4 true

```
#define true 1
```

Definition at line 20 of file eHSM\_Types\_Ip.h.

## 4.69.2 Typedef Documentation

### 4.69.2.1 ehsm\_addr\_t

```
typedef ehsm_uint32_t ehsm_addr_t
```

Definition at line 41 of file eHSM\_Types\_Ip.h.

### 4.69.2.2 ehsm\_bool\_t

```
typedef signed char ehsm_bool_t
```

Definition at line 40 of file eHSM\_Types\_Ip.h.

### 4.69.2.3 ehsm\_handle\_t

```
typedef unsigned int ehsm_handle_t
```

Definition at line 43 of file eHSM\_Types\_Ip.h.

### 4.69.2.4 ehsm\_int16\_t

```
typedef short ehsm_int16_t
```

Definition at line 36 of file eHSM\_Types\_Ip.h.

### 4.69.2.5 ehsm\_int32\_t

```
typedef int ehsm_int32_t
```

Definition at line 34 of file eHSM\_Types\_Ip.h.

### 4.69.2.6 ehsm\_int64\_t

```
typedef long long int ehsm_int64_t
```

Definition at line 38 of file eHSM\_Types\_Ip.h.

#### 4.69.2.7 ehsm\_int8\_t

```
typedef signed char ehsm_int8_t
```

Definition at line 32 of file eHSM\_Types\_lp.h.

#### 4.69.2.8 ehsm\_uint16\_t

```
typedef unsigned short ehsm_uint16_t
```

Definition at line 37 of file eHSM\_Types\_lp.h.

#### 4.69.2.9 ehsm\_uint32\_t

```
typedef unsigned int ehsm_uint32_t
```

Definition at line 35 of file eHSM\_Types\_lp.h.

#### 4.69.2.10 ehsm\_uint64\_t

```
typedef unsigned long long int ehsm_uint64_t
```

Definition at line 39 of file eHSM\_Types\_lp.h.

#### 4.69.2.11 ehsm\_uint8\_t

```
typedef unsigned char ehsm_uint8_t
```

Definition at line 33 of file eHSM\_Types\_lp.h.

## 4.70 Hsm\_Hal.c File Reference

This file provides HSM integration functions.

```
#include "Hsm_Hal.h"
#include "Fls_Hal.h"
```

## Classes

- struct [HSM\\_KeyHandleInfoType](#)  
*Specifies the Key handle info content,size is 40Bytes.*
- struct [HSM\\_KeySlotInfoType](#)  
*Specifies the Key slot info content,size is 8Bytes.*
- struct [HSM\\_KeyIndexInfoType](#)  
*Specifies the Key index info content,size is 8Bytes.*
- struct [HSM\\_FlashKeyType](#)  
*Specifies the Key info content,size is 56Bytes.*
- struct [HSM\\_FlashKeyPageType](#)  
*Specifies the flash Key content,size is DFLASH\_PAGE\_SIZE.*
- struct [HSM\\_RamKeyInfoType](#)  
*Specifies the Ram Key info.*
- struct [HSM\\_RamKeyHeadType](#)  
*Specifies the Ram Key space header.*

## Functions

- [ISR](#) (HSM\_IRQHandler)
- void [HSM\\_Hal\\_Lock](#) (void)  
*Lock access hsm path.*
- void [HSM\\_Hal\\_Unlock](#) (void)  
*unlock access hsm path.*
- boolean [HSM\\_Hal\\_GetLockState](#) (void)  
*get access hsm path lock state.*
- Hal\_StatusType [HSM\\_Hal\\_Init](#) (boolean HsmIrqEn, uint8 HsmIrqPri)  
*hsm module init function.*
- void [HSM\\_Hal\\_Deinit](#) (void)  
*hsm module deinit function.*
- Hal\_StatusType [HSM\\_Hal\\_GetRnd](#) ([HSM\\_RndAlgo](#) Algo, uint32 RndSize, uint8 \*Rnd)  
*get random.*
- Hal\_StatusType [HSM\\_Hal\\_AesCipher](#) (const [HSM\\_SymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Aes Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_Sm4Cipher](#) (const [HSM\\_SymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Sm4 Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_CipherMac](#) (const [HSM\\_CMacCfgType](#) \*CfgPtr, [HSM\\_InOutMacType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Cipher Mac Generate/Verify mac function.*
- Hal\_StatusType [HSM\\_Hal\\_Hash](#) (const [HSM\\_HashAlgoType](#) Algo, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*generate message digest(hash value) function.*
- Hal\_StatusType [HSM\\_Hal\\_HashMac](#) (const [HSM\\_HMacCfgType](#) \*CfgPtr, const [HSM\\_InOutMacType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*generate message hash mac function use hash algo*
- Hal\_StatusType [HSM\\_Hal\\_EccSign](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutSignType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*ECC Sign/Verify function.*
- Hal\_StatusType [HSM\\_Hal\\_RsaSign](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutSignType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Rsa Sign/Verify function.*

- Hal\_StatusType [HSM\\_Hal\\_Sm2Sign](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutSignType](#) \*InOutPtr, [HSM\\_ProcessMode](#) ProMode)  
*SM2 Sign/Verify function.*
- Hal\_StatusType [HSM\\_Hal\\_RsaCipher](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_ProcessMode](#) ProMode)  
*Rsa Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_Sm2Cipher](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_ProcessMode](#) ProMode)  
*Sm2 Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_RemoveKey](#) ([HSM\\_KeyId](#) KeyId)  
*Remove Key.*
- Hal\_StatusType [HSM\\_Hal\\_SetPlainKey](#) (const [HSM\\_PlainKeyCfgType](#) \*CfgPtr)  
*set plain key to hsm mem.*
- Hal\_StatusType [HSM\\_Hal\\_GenerateKey](#) (const [HSM\\_GenKeyCfgType](#) \*CfgPtr)  
*generate key for algo, eg.aes/sm2.*
- Hal\_StatusType [HSM\\_Hal\\_DeriveKey](#) ([HSM\\_KeyId](#) ParentKeyId, [HSM\\_GenKeyAlgo](#) ParentKeyAlgo, const [HSM\\_DeriveKeyCfgType](#) \*TargetKeyPtr)  
*drive key*
- Hal\_StatusType [HSM\\_Hal\\_GetSecretkey](#) ([HSM\\_KeyId](#) TpKeyId, [HSM\\_KeyId](#) AuthKeyId, [HSM\\_KeyId](#) TargetKeyId, const [HSM\\_SecretKeyCfgType](#) \*ExportKeyCfg)  
*get secret key to hsm*
- Hal\_StatusType [HSM\\_Hal\\_SetSecretKey](#) ([HSM\\_KeyId](#) TpKeyId, [HSM\\_KeyId](#) AuthKeyId, const [HSM\\_SecretKeyCfgType](#) \*ImportKeyPtr, [HSM\\_KeyId](#) KeyId)  
*set secret key to hsm*
- Hal\_StatusType [HSM\\_Hal\\_GetPubKeyFromPrvKey](#) ([HSM\\_KeyId](#) KeyId, uint8 \*PubKey, uint32 \*PubKeySize, [HSM\\_GenKeyAlgo](#) Algo)  
*get pubkey form private key*
- Hal\_StatusType [HSM\\_Hal\\_GenerateDHKey](#) ([HSM\\_KeyId](#) LocalKeyId, const uint8 \*RemotePubKey, uint32 RemotePubKeySize, [HSM\\_GenKeyAlgo](#) ParKeyAlgo, const [HSM\\_GenKeyCfgType](#) \*TargetKeyPtr)  
*generate DH key.*
- Hal\_StatusType [HSM\\_Hal\\_GetKeyStatus](#) (const [HSM\\_KeyStatusType](#) \*Status)  
*get key status for specific key id*
- [HSM\\_LifeCycleType](#) [HSM\\_Hal\\_GetLifeCycle](#) (void)  
*Get LifeCycle.*
- uint32 [HSM\\_Hal\\_SetLifeCycle](#) ([HSM\\_LifeCycleType](#) LcIndex)  
*Set LifeCycle.*
- Hal\_StatusType [HSM\\_Hal\\_DebugAuth](#) ([HSM\\_DebugAuthConfigType](#) \*CfgPtr)  
*Debug authentication.*
- Hal\_StatusType [HSM\\_Hal\\_GetChallenge](#) ([HSM\\_ChallengeType](#) Type, uint8 \*ChBuf, uint32 ChSize)  
*get challenge*
- uint32 [HSM\\_Hal\\_OtpRead](#) (uint32 OtpAddr, uint32 BytesLen, uint8 \*Data)  
*read otp*
- uint32 [HSM\\_Hal\\_OtpWrite](#) (uint32 OtpAddr, uint32 BytesLen, uint8 \*Data)  
*write otp*
- Hal\_StatusType [HSM\\_Hal\\_EnableSecureBoot](#) (void)  
*enable secureboot*
- Hal\_StatusType [HSM\\_Hal\\_DisableSecureBoot](#) (void)  
*disable secureboot*
- void [HSM\\_Hal\\_InstallCallback](#) (const [Hal\\_CallbackType](#) Callback, void \*Args)  
*irq callback.*
- Hal\_StatusType [HSM\\_Hal\\_HostImageSecureUpgrade](#) (const [HSM\\_SecureUpgradeType](#) \*HostUpgradePara)  
*Host image upgrade.*
- Hal\_StatusType [HSM\\_Hal\\_HostImageSecureVerify](#) (const [HSM\\_ImageVerifyType](#) \*HostVerifyPara)  
*Host image upgrade verify.*



- uint32 [HSM\\_Hal\\_GetHsmFwVersion](#) (const uint32 \*Version)  
*get hsm fw version*
- Hal\_StatusType [HSM\\_Hal\\_SetOtpKeyCipherAlgo](#) (HSM\_OtpKeyCipherAlgo Algo)  
*Otp Hw Ctrl.*
- Hal\_StatusType [HSM\\_Hal\\_SetImageSecureUpgradeAlgo](#) (HSM\_OtpCtrlAlgoType Algo, HSM\_ImageType Type)  
*Otp Hw Ctrl.*
- Hal\_StatusType [HSM\\_Hal\\_SetImageSecureVerifyAlgo](#) (HSM\_OtpCtrlAlgoType Algo, HSM\_ImageType Type)  
*Otp Hw Ctrl.*
- uint32 [HSM\\_Hal\\_GetRndKey](#) (HSM\_OtpKeyLevel Level, HSM\_RndOtpKeyType KeyType, const uint8 \*KeyOut)  
*generate random key in bootloader.*
- uint32 [HSM\\_Hal\\_SetOtpRndKey](#) (HSM\_OtpKeyId KeyId, HSM\_OtpKeyLevel KeyLevel, HSM\_RndOtpKeyType KeyType, uint32 KeyAttr)  
*set random key to otp key slot*
- uint32 [HSM\\_Hal\\_SetOtpExternalKey](#) (HSM\_OtpKeyId KeyId, const uint8 \*ExternKey, uint32 ExternKeyLen, HSM\_OtpKeyLevel KeyLevel, uint32 KeyAttr)  
*set external key or hash value(raw dta key) to otp key slot*

### 4.70.1 Detailed Description

This file provides HSM integration functions.

### 4.70.2 Function Documentation

#### 4.70.2.1 HSM\_Hal\_AesCipher()

```
Hal_StatusType HSM_Hal_AesCipher (
 const HSM_SymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Aes Decrypt/Encrypt function.

#### Note

Function ID: DES\_HSM\_API\_003

#### Parameters

|    |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | Aes Algo config KeyId: Key Index. Iv: init vector. IvLen: iv Len Padding:data padding type. HSM_NOPADDING = 0U, HSM_PSASSA_PSS = 1U, HSM_PKCS7 = 2U, HSM_ONEWITHZEROS = 3U CipherDir :request opeartion, eg.decrypt/encrypt. HSM_ENCRYPTION = 0U HSM_DECRYPTION = 1U CMode:for sym algo and CMAC algo, eg.ECB/CBC. ECB_MODE = 1U CBC_MODE = 3U CFB_MODE = 4U OFB_MODE = 5U CTR_MODE = 6U SymAlgo: symmetric algo type, eg.AES_128/AES_256 HSM_AES_128 = 5U, HSM_AES_256 = 7U, |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length                                                                                                                                                                                                                                                                                                                                                    |
| in | <i>ProMode</i>  | process mode. START_MODE = (0x01U), UPDATE_MODE = (0x02U), FINISH_MODE = (0x04U), FW_ONEPASS_MODE = (0x07U),                                                                                                                                                                                                                                                                                                                                                                  |

Returns

Hal\_StatusType: STATUS\_ERROR: error happend in Cipher operation; STATUS\_SUCCESS: successfull

Definition at line 2328 of file Hsm\_Hal.c.

4.70.2.2 HSM\_Hal\_CipherMac()

```
Hal_StatusType HSM_Hal_CipherMac (
 const HSM_CMacCfgType * CfgPtr,
 HSM_InOutMacType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Cipher Mac Generate/Verify mac function.

Note

Function ID: DES\_HSM\_API\_010

Parameters

|    |          |                                                                                                                                                                                                              |
|----|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | CfgPtr   | Algo config KeyId: Key Index. MacDir :request opeartion, eg.decrypt/encrypt. HSM_MAC_GEN = 0U HSM_MAC_VERIFY = 1U SymAlgo: symmetric algo type, eg.AES_128/SM4 HSM_AES_128 = 6U, HSM_SM4 = 8U,               |
| in | InOutPtr | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length MacInBuf: store Mac value when verify MacInBufLen: Mac length Vry: verify result. |
| in | ProMode  | process mode. START_MODE = (0x01U), UPDATE_MODE = (0x02U), FINISH_MODE = (0x04U), FW_ONEPASS_MODE = (0x07U),                                                                                                 |

Returns

Hal\_StatusType: STATUS\_ERROR: error happend in Cipher operation; STATUS\_SUCCESS: successfull

Definition at line 2373 of file Hsm\_Hal.c.

4.70.2.3 HSM\_Hal\_DebugAuth()

```
Hal_StatusType HSM_Hal_DebugAuth (
 HSM_DebugAuthConfigType * CfgPtr)
```

Debug authentication.

Note

Function ID: DES\_HSM\_API\_028

## Parameters

|    |               |                                      |
|----|---------------|--------------------------------------|
| in | <i>CfgPtr</i> | Debug Auth config information struct |
|----|---------------|--------------------------------------|

## Returns

op status

Definition at line 3263 of file Hsm\_Hal.c.

## 4.70.2.4 HSM\_Hal\_Deinit()

```
void HSM_Hal_Deinit (
 void)
```

hsm module deinit function.

## Note

Function ID: DES\_HSM\_API\_002

## Returns

op status

Definition at line 2307 of file Hsm\_Hal.c.

## 4.70.2.5 HSM\_Hal\_DeriveKey()

```
Hal_StatusType HSM_Hal_DeriveKey (
 HSM_KeyId ParentKeyId,
 HSM_GenKeyAlgo ParentKeyAlgo,
 const HSM_DeriveKeyCfgType * TargetKeyPtr)
```

drive key

## Note

Function ID: DES\_HSM\_API\_015

## Parameters

|    |                     |                    |
|----|---------------------|--------------------|
| in | <i>ParentKeyPtr</i> | parent key config. |
| in | <i>TargetKeyPtr</i> | Target key config. |

**Returns**

op status

Definition at line 2968 of file Hsm\_Hal.c.

**4.70.2.6 HSM\_Hal\_DisableSecureBoot()**

```
Hal_StatusType HSM_Hal_DisableSecureBoot (
 void)
```

disable secureboot

**Note**

Function ID: DES\_HSM\_API\_024

**Returns**

op status

Definition at line 3312 of file Hsm\_Hal.c.

**4.70.2.7 HSM\_Hal\_EccSign()**

```
Hal_StatusType HSM_Hal_EccSign (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutSignType * InOutPtr,
 HSM_ProcessMode ProMode)
```

ECC Sign/Verify function.

**Note**

Function ID: DES\_HSM\_API\_007

**Parameters**

|    |                 |                                                                                                                                                                                                                            |
|----|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | ecc Algo config.                                                                                                                                                                                                           |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length SignInBuf: store signature value when verify SignInBufLen: signature length Vry: verify result. |
| in | <i>ProMode</i>  | process mode.                                                                                                                                                                                                              |

**Returns**

op status

Definition at line 2622 of file Hsm\_Hal.c.

4.70.2.8 HSM\_Hal\_EnableSecureBoot()

```
Hal_StatusType HSM_Hal_EnableSecureBoot (
 void)
```

enable secureboot

Note

Function ID: DES\_HSM\_API\_023

Returns

op status

Definition at line 3304 of file Hsm\_Hal.c.

4.70.2.9 HSM\_Hal\_GenerateDHKey()

```
Hal_StatusType HSM_Hal_GenerateDHKey (
 HSM_KeyId LocalKeyId,
 const uint8 * RemotePubKey,
 uint32 RemotePubKeySize,
 HSM_GenKeyAlgo ParKeyAlgo,
 const HSM_GenKeyCfgType * TargetKeyPtr)
```

generate DH key.

Note

Function ID: DES\_HSM\_API\_019

Parameters

|     |                         |                                    |
|-----|-------------------------|------------------------------------|
| in  | <i>LocalKeyId</i>       | local key id.                      |
| out | <i>RemotePubKey</i>     | the pubkey key buffer from remote. |
| out | <i>RemotePubKeySize</i> | the remote pubkey key size.        |
| in  | <i>TargetKeyPtr</i>     | generate dh key config.            |

Returns

op status

Definition at line 3164 of file Hsm\_Hal.c.

4.70.2.10 HSM\_Hal\_GenerateKey()

```
Hal_StatusType HSM_Hal_GenerateKey (
 const HSM_GenKeyCfgType * CfgPtr)
```

generate key for algo, eg.aes/sm2.

Note

Function ID: DES\_HSM\_API\_014

Parameters

|    |        |                      |
|----|--------|----------------------|
| in | CfgPtr | generate key config. |
|----|--------|----------------------|

Returns

op status

Definition at line 2926 of file Hsm\_Hal.c.

4.70.2.11 HSM\_Hal\_GetChallenge()

```
Hal_StatusType HSM_Hal_GetChallenge (
 HSM_ChallengeType Type,
 uint8 * ChBuf,
 uint32 ChSize)
```

get challenge

Note

Function ID: DES\_HSM\_API\_029

Parameters

|     |        |                    |
|-----|--------|--------------------|
| in  | Type   | challenge type     |
| out | ChBuf  | challenge buf      |
| in  | ChSize | challenge buf size |

Returns

op status

Definition at line 3279 of file Hsm\_Hal.c.

4.70.2.12 HSM\_Hal\_GetHsmFwVersion()

```
uint32 HSM_Hal_GetHsmFwVersion (
 const uint32 * Version)
```

get hsm fw version

Returns

operate status

Definition at line 3377 of file Hsm\_Hal.c.

4.70.2.13 HSM\_Hal\_GetKeyStatus()

```
Hal_StatusType HSM_Hal_GetKeyStatus (
 const HSM_KeyStatusType * Status)
```

get key status for specific key id

Note

Function ID: DES\_HSM\_API\_027

Parameters

|    |        |                                                                                                                                                                                                                                                |
|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | Status | key status struct TargetKeyId: the target id get it status CertificationKeyId: certification key id for status information CertificationAuthSize: size CertificationAuth:auth value KeyStatusSize: status size for target id KeyStatus: status |
|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Returns

op status

Definition at line 3217 of file Hsm\_Hal.c.

4.70.2.14 HSM\_Hal\_GetLifeCycle()

```
HSM_LifeCycleType HSM_Hal_GetLifeCycle (
 void)
```

Get LifeCycle.

Note

Function ID: DES\_HSM\_API\_040

Returns

op status

Definition at line 3233 of file Hsm\_Hal.c.

#### 4.70.2.15 HSM\_Hal\_GetLockState()

```
boolean HSM_Hal_GetLockState (
 void)
```

get access hsm path lock state.

##### Note

Function ID: DES\_HSM\_API\_005

##### Returns

op status

Definition at line 2246 of file Hsm\_Hal.c.

#### 4.70.2.16 HSM\_Hal\_GetPubKeyFromPrvKey()

```
Hal_StatusType HSM_Hal_GetPubKeyFromPrvKey (
 HSM_KeyId KeyId,
 uint8 * PubKey,
 uint32 * PubKeySize,
 HSM_GenKeyAlgo Algo)
```

get pubkey form private key

##### Note

Function ID: DES\_HSM\_API\_026

##### Parameters

|     |                   |                        |
|-----|-------------------|------------------------|
| in  | <i>KeyId</i>      | Target key id.         |
| out | <i>PubKey</i>     | the pubkey key buffer. |
| out | <i>PubKeySize</i> | the pubkey key size.   |
| in  | <i>Algo</i>       | target key algo.       |

##### Returns

op status

Definition at line 3142 of file Hsm\_Hal.c.

#### 4.70.2.17 HSM\_Hal\_GetRnd()

```
Hal_StatusType HSM_Hal_GetRnd (
 HSM_RndAlgo Algo,
```



```
uint32 RndSize,
uint8 * Rnd)
```

get random.

#### Note

Function ID: DES\_HSM\_API\_021

#### Parameters

|     |                |                                    |
|-----|----------------|------------------------------------|
| in  | <i>Algo</i>    | trng or prng use algo type         |
| in  | <i>RndSize</i> | random size, max size is 128 bytes |
| out | <i>Rnd</i>     | random information                 |

#### Returns

op status

Definition at line 2319 of file Hsm\_Hal.c.

#### 4.70.2.18 HSM\_Hal\_GetRndKey()

```
uint32 HSM_Hal_GetRndKey (
 HSM_OtpKeyLevel Level,
 HSM_RndOtpKeyType KeyType,
 const uint8 * KeyOut)
```

generate random key in bootloader.

#### Note

Function ID: DES\_HSM\_API\_018

#### Parameters

|     |                |                   |
|-----|----------------|-------------------|
| in  | <i>Level</i>   | key level.        |
| in  | <i>KeyType</i> | support key type. |
| out | <i>KeyOut</i>  | key out buffer.   |

#### Returns

op status

Definition at line 3535 of file Hsm\_Hal.c.

#### 4.70.2.19 HSM\_Hal\_GetSecretkey()

```
Hal_StatusType HSM_Hal_GetSecretkey (
 HSM_KeyId TpKeyId,
 HSM_KeyId AuthKeyId,
 HSM_KeyId TargetKeyId,
 const HSM_SecretKeyCfgType * ExportKeyCfg)
```

get secret key to hsm

##### Note

Function ID: DES\_HSM\_API\_017

##### Parameters

|    |                     |                        |
|----|---------------------|------------------------|
| in | <i>TpKeyId</i>      | transport key id.      |
| in | <i>AuthKeyId</i>    | auth key id.           |
| in | <i>TargetKeyId</i>  | Target key id.         |
| in | <i>ExportKeyCfg</i> | export key blob config |

##### Returns

op status

Definition at line 3016 of file Hsm\_Hal.c.

#### 4.70.2.20 HSM\_Hal\_Hash()

```
Hal_StatusType HSM_Hal_Hash (
 const HSM_HashAlgoType Algo,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

generate message digest(hash value) function.

##### Note

Function ID: DES\_HSM\_API\_011

##### Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | hash Algo config.        |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

##### Returns

op status

Definition at line 2458 of file Hsm\_Hal.c.

4.70.2.21 HSM\_Hal\_HashMac()

```
Hal_StatusType HSM_Hal_HashMac (
 const HSM_HMacCfgType * CfgPtr,
 const HSM_InOutMacType * InOutPtr,
 HSM_ProcessMode ProMode)
```

generate message hash mac function use hash algo

Note

Function ID: DES\_HSM\_API\_012

Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | hash Algo config.        |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

Returns

op status

Definition at line 2505 of file Hsm\_Hal.c.

4.70.2.22 HSM\_Hal\_HostImageSecureUpgrade()

```
Hal_StatusType HSM_Hal_HostImageSecureUpgrade (
 const HSM_SecureUpgradeType * HostUpgradePara)
```

Host image upgrade.

Parameters

|    |                        |                                         |
|----|------------------------|-----------------------------------------|
| in | <i>HostUpgradePara</i> | Pointer to host image upgrade parameter |
|----|------------------------|-----------------------------------------|

Returns

operate status

Definition at line 3334 of file Hsm\_Hal.c.

4.70.2.23 HSM\_Hal\_HostImageSecureVerify()

```
Hal_StatusType HSM_Hal_HostImageSecureVerify (
 const HSM_ImageVerifyType * HostVerifyPara)
```

Host image upgrade verify.

Parameters

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

Returns

operate status

Definition at line 3356 of file Hsm\_Hal.c.

4.70.2.24 HSM\_Hal\_Init()

```
Hal_StatusType HSM_Hal_Init (
 boolean HsmIrqEn,
 uint8 HsmIrqPri)
```

hsm module init function.

Note

Function ID: DES\_HSM\_API\_001

Parameters

|    |                  |                                                    |
|----|------------------|----------------------------------------------------|
| in | <i>HsmIrqEn</i>  | hsm irq enable/disable 1:enable 0:disable(default) |
| in | <i>HsmIrqPri</i> | hsm irq priority.                                  |

Returns

op status

Definition at line 2258 of file Hsm\_Hal.c.

4.70.2.25 HSM\_Hal\_InstallCallback()

```
void HSM_Hal_InstallCallback (
 const Hal_CallbackType Callback,
 void * Args)
```

irq callback.

Note

Function ID: DES\_HSM\_API\_025

**Parameters**

|    |                 |                   |
|----|-----------------|-------------------|
| in | <i>Callback</i> | callback function |
| in | <i>Args</i>     | args              |

**Returns**

op status

Definition at line 3320 of file Hsm\_Hal.c.

**4.70.2.26 HSM\_Hal\_Lock()**

```
void HSM_Hal_Lock (
 void)
```

Lock access hsm path.

**Note**

Function ID: DES\_HSM\_API\_043

**Returns**

op status

Definition at line 2236 of file Hsm\_Hal.c.

**4.70.2.27 HSM\_Hal\_OtpRead()**

```
uint32 HSM_Hal_OtpRead (
 uint32 OtpAddr,
 uint32 BytesLen,
 uint8 * Data)
```

read otp

**Note**

Function ID: DES\_HSM\_API\_030

**Parameters**

|     |                 |                   |
|-----|-----------------|-------------------|
| in  | <i>OtpAddr</i>  | otp address value |
| in  | <i>BytesLen</i> | read bytes len    |
| out | <i>Data</i>     | Secureboot Config |

Returns

op status

Definition at line 3290 of file Hsm\_Hal.c.

4.70.2.28 HSM\_Hal\_OtpWrite()

```
uint32 HSM_Hal_OtpWrite (
 uint32 OtpAddr,
 uint32 BytesLen,
 uint8 * Data)
```

write otp

Note

Function ID: DES\_HSM\_API\_042

Parameters

|     |                 |                   |
|-----|-----------------|-------------------|
| in  | <i>OtpAddr</i>  | otp address value |
| in  | <i>BytesLen</i> | read bytes len    |
| out | <i>Data</i>     | Secureboot Config |

Returns

op status

Definition at line 3297 of file Hsm\_Hal.c.

4.70.2.29 HSM\_Hal\_RemoveKey()

```
Hal_StatusType HSM_Hal_RemoveKey (
 HSM_KeyId KeyId)
```

Remove Key.

Note

Function ID: DES\_HSM\_API\_020

Parameters

|    |                    |           |
|----|--------------------|-----------|
| in | <i>Key↔<br/>Id</i> | Key Index |
|----|--------------------|-----------|

**Returns**

op status

Definition at line 2745 of file Hsm\_Hal.c.

**4.70.2.30 HSM\_Hal\_RsaCipher()**

```
Hal_StatusType HSM_Hal_RsaCipher (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Rsa Decrypt/Encrypt function.

**Note**

Function ID:DES\_HSM\_API\_039

**Parameters**

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | rsa Algo config.         |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

**Returns**

op status

Definition at line 2671 of file Hsm\_Hal.c.

**4.70.2.31 HSM\_Hal\_RsaSign()**

```
Hal_StatusType HSM_Hal_RsaSign (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutSignType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Rsa Sign/Verify function.

**Note**

Function ID: DES\_HSM\_API\_008

**Parameters**

|    |                 |                              |
|----|-----------------|------------------------------|
| in | <i>CfgPtr</i>   | rsa config.                  |
| in | <i>InOutPtr</i> | rsa Input and output config. |
| in | <i>ProMode</i>  | process mode.                |

**Returns**

op status

Definition at line 2639 of file Hsm\_Hal.c.

**4.70.2.32 HSM\_Hal\_SetImageSecureUpgradeAlgo()**

```
Hal_StatusType HSM_Hal_SetImageSecureUpgradeAlgo (
 HSM_OtpCtrlAlgoType Algo,
 HSM_ImageType Type)
```

Otp Hw Ctrl.

Otp Hw Ctrl, set image secure upgrade algo type.

**Parameters**

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

**Returns**

operate status

Definition at line 3438 of file Hsm\_Hal.c.

**4.70.2.33 HSM\_Hal\_SetImageSecureVerifyAlgo()**

```
Hal_StatusType HSM_Hal_SetImageSecureVerifyAlgo (
 HSM_OtpCtrlAlgoType Algo,
 HSM_ImageType Type)
```

Otp Hw Ctrl.

Otp Hw Ctrl,set image secure verify algo type.

**Parameters**

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

**Returns**

operate status

Definition at line 3489 of file Hsm\_Hal.c.



4.70.2.34 HSM\_Hal\_SetLifeCycle()

```
uint32 HSM_Hal_SetLifeCycle (
 HSM_LifeCycleType LcIndex)
```

Set LifeCycle.

Note

Function ID: DES\_HSM\_API\_041

Parameters

|    |                |                |
|----|----------------|----------------|
| in | <i>LcIndex</i> | lifecycle type |
|----|----------------|----------------|

Returns

op status

Definition at line 3252 of file Hsm\_Hal.c.

4.70.2.35 HSM\_Hal\_SetOtpExternalKey()

```
uint32 HSM_Hal_SetOtpExternalKey (
 HSM_OtpKeyId KeyId,
 const uint8 * ExternKey,
 uint32 ExternKeyLen,
 HSM_OtpKeyLevel KeyLevel,
 uint32 KeyAttr)
```

set external key or hash value(raw dta key) to otp key slot

Note

Function ID: DES\_HSM\_API\_034

Parameters

|    |                     |                                    |
|----|---------------------|------------------------------------|
| in | <i>KeyId</i>        | otp key id.                        |
| in | <i>ExternKey</i>    | the key data or hash value.        |
| in | <i>ExternKeyLen</i> | the key data or hash value length. |
| in | <i>KeyLevel</i>     | otp key level.                     |
| in | <i>KeyAttr</i>      | otp key attribute.                 |

Returns

operate status

Definition at line 3602 of file Hsm\_Hal.c.

4.70.2.36 HSM\_Hal\_SetOtpKeyCipherAlgo()

```
Hal_StatusType HSM_Hal_SetOtpKeyCipherAlgo (
 HSM_OtpKeyCipherAlgo Algo)
```

Otp Hw Ctrl.

Otp Hw Ctrl, Set encrypt otp key algo.

Parameters

|    |                |                                        |
|----|----------------|----------------------------------------|
| in | HostVerifyPara | Pointer to host image verify parameter |
|----|----------------|----------------------------------------|

Returns

operate status

Definition at line 3400 of file Hsm\_Hal.c.

4.70.2.37 HSM\_Hal\_SetOtpRndKey()

```
uint32 HSM_Hal_SetOtpRndKey (
 HSM_OtpKeyId KeyId,
 HSM_OtpKeyLevel KeyLevel,
 HSM_RndOtpKeyType KeyType,
 uint32 KeyAttr)
```

set random key to otp key slot

Note

Function ID: DES\_HSM\_API\_035

Parameters

|    |          |                          |
|----|----------|--------------------------|
| in | KeyId    | otp key id.              |
| in | KeyLevel | otp key level.           |
| in | KeyType  | support random key type. |
| in | KeyAttr  | otp key attribute.       |

Returns

operate status

Definition at line 3574 of file Hsm\_Hal.c.

#### 4.70.2.38 HSM\_Hal\_SetPlainKey()

```
Hal_StatusType HSM_Hal_SetPlainKey (
 const HSM_PlainKeyCfgType * CfgPtr)
```

set plain key to hsm mem.

##### Note

Function ID: DES\_HSM\_API\_013

##### Parameters

|    |               |                   |
|----|---------------|-------------------|
| in | <i>CfgPtr</i> | plain key config. |
|----|---------------|-------------------|

##### Returns

op status

Definition at line 2754 of file Hsm\_Hal.c.

#### 4.70.2.39 HSM\_Hal\_SetSecretKey()

```
Hal_StatusType HSM_Hal_SetSecretKey (
 HSM_KeyId TpKeyId,
 HSM_KeyId AuthKeyId,
 const HSM_SecretKeyCfgType * ImportKeyPtr,
 HSM_KeyId KeyId)
```

set secret key to hsm

##### Note

Function ID: DES\_HSM\_API\_016

##### Parameters

|    |                     |                    |
|----|---------------------|--------------------|
| in | <i>TpKeyId</i>      | transport key id.  |
| in | <i>AuthKeyId</i>    | auth key id.       |
| in | <i>ImportKeyPtr</i> | Target key config. |
| in | <i>KeyId</i>        | Target key id.     |

##### Returns

op status

Definition at line 3061 of file Hsm\_Hal.c.

4.70.2.40 HSM\_Hal\_Sm2Cipher()

```
Hal_StatusType HSM_Hal_Sm2Cipher (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Sm2 Decrypt/Encrypt function.

Note

Function ID:

Parameters

|    |          |                          |
|----|----------|--------------------------|
| in | CfgPtr   | sm2 Algo config.         |
| in | InOutPtr | Input and output config. |

Returns

op status

Definition at line 2703 of file Hsm\_Hal.c.

4.70.2.41 HSM\_Hal\_Sm2Sign()

```
Hal_StatusType HSM_Hal_Sm2Sign (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutSignType * InOutPtr,
 HSM_ProcessMode ProMode)
```

SM2 Sign/Verify function.

Note

Function ID: DES\_HSM\_API\_009

Parameters

|    |          |                          |
|----|----------|--------------------------|
| in | CfgPtr   | Sm2 Algo config.         |
| in | InOutPtr | Input and output config. |

Returns

op status

Definition at line 2656 of file Hsm\_Hal.c.

## 4.70.2.42 HSM\_Hal\_Sm4Cipher()

```
Hal_StatusType HSM_Hal_Sm4Cipher (
 const HSM_SymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Sm4 Decrypt/Encrypt function.

## Note

Function ID: DES\_HSM\_API\_004

## Parameters

|    |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | sm4 Algo config KeyId: Key Index. Iv: init vector. IvLen: iv Len Padding:data padding type. HSM_NOPADDING = 0U, HSM_PSASSA_PSS = 1U, HSM_PKCS7 = 2U, HSM_ONEWITHZEROS = 3U CipherDir :request opeartion, eg.decrypt/encrypt. HSM_ENCRYPTION = 0U, HSM_DECRYPTION = 1U, CMode:for sym algo and CMAC algo, eg.ECB/CBC. ECB_MODE = 1U CBC_MODE = 3U CFB_MODE = 4U OFB_MODE = 5U CTR_MODE = 6U SymAlgo: symmetric algo type, eg.AES_128/AES_256 HSM_SM4 = 8U, |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length Vry:ignore.                                                                                                                                                                                                                                                                                                                    |
| in | <i>ProMode</i>  | process mode. START_MODE = (0x01U), UPDATE_MODE = (0x02U), FINISH_MODE = (0x04U), FW_ONEPASS_MODE = (0x07U),                                                                                                                                                                                                                                                                                                                                              |

## Returns

Hal\_StatusType: STATUS\_ERROR: error happend in Cipher operation; STATUS\_SUCCESS: successfull

Definition at line 2350 of file Hsm\_Hal.c.

## 4.70.2.43 HSM\_Hal\_Unlock()

```
void HSM_Hal_Unlock (
 void)
```

unlock access hsm path.

## Note

Function ID: DES\_HSM\_API\_044

## Returns

op status

Definition at line 2241 of file Hsm\_Hal.c.

## 4.70.2.44 ISR()

```
ISR (
 HSM_IRQHandler)
```

Definition at line 3666 of file Hsm\_Hal.c.

## 4.71 Hsm\_Hal.h File Reference

This file provides hsm integration functions interface.

```
#include "Device_Register.h"
#include "eHSM_IntCfg_Ip.h"
#include "eHSM_Config_Ip.h"
#include "eHSM_Types_Ip.h"
#include "eHSM_Debug_Ip.h"
#include "eHSM_Mailbox_Prtcl_Ip.h"
#include "eHSM_Err_Code_Ip.h"
#include "eHSM_If_Evita_Ip.h"
#include "eHSM_If_Evita_Types_Ip.h"
#include "eHSM_If_Evita_ErrCode_Ip.h"
#include "eHSM_Mgr_Ctx_Ip.h"
#include "eHSM_Com_Struct_Ip.h"
#include "eHSM_Mailbox_Reg_Ip.h"
#include "eHSM_If_Ext_Ip.h"
#include "AC784xx_Hsm_Reg.h"
#include "Core_Hal.h"
#include "string.h"
```

## Classes

- struct [HSM\\_BootCfgType](#)  
*Specifies the secureboot information.*
- struct [HSM\\_SecureUpgradeType\\_\\_](#)  
*Host image secure upgrade information.*
- struct [HSM\\_ImageVerifyType\\_\\_](#)  
*Host image verify information.*
- struct [HSM\\_AsymCfgType](#)  
*Specifies the asymmetric algo config struct.*
- struct [HSM\\_SymCfgType](#)  
*Specifies the symmetric algo config struct.*
- struct [HSM\\_HMacCfgType](#)  
*Specifies the Hash Mac algo config struct.*
- struct [HSM\\_CMacCfgType](#)  
*Specifies the Cipher Mac algo config struct.*
- struct [HSM\\_InOutType](#)  
*the basic Input and output struct.*
- struct [HSM\\_InOutMacType](#)  
*the Input and output struct about gen/ver Mac interface.*
- struct [HSM\\_InOutSignType](#)  
*the Input and output struct about gen/ver signature interface.*
- struct [HSM\\_KeyFlagsElementType](#)

- the key flags.*
- struct [HSM\\_KeyUsagesType](#)
  - the key attribute config struct.*
- struct [HSM\\_KeyActUseFlagsType](#)
  - the key active attribute config struct.*
- struct [HSM\\_EccPubKeyType](#)
  - the ecc pubkey struct.*
- struct [HSM\\_RsaPubKeyType](#)
  - the rsa pubkey struct.*
- struct [HSM\\_DhParamType](#)
  - the Dh algo param.*
- struct [Hsm\\_DhPubKeyType\\_](#)
  - the Dh pubkey struct.*
- union [Hsm\\_PubKeyDataType\\_](#)
  - the public key struct for all algo.*
- struct [HSM\\_DhPriKeyType](#)
  - the Dh private key struct.*
- struct [Hsm\\_RsaCrtType\\_](#)
  - the rsa private key struct when CRT.*
- union [Hsm\\_PriKeyDataType\\_](#)
  - the private key struct for all algo.*
- struct [HSM\\_SecretKeyCfgType](#)
  - the Secret Key config struct for key import or export by secret.*
- struct [HSM\\_DeriveKeyCfgType](#)
  - the derive Key config struct.*
- struct [HSM\\_GenKeyCfgType](#)
  - the generate random Key config struct by random generator in hsm.*
- struct [HSM\\_PlainKeyCfgType](#)
  - the set plain key config struct.*
- struct [HSM\\_DebugAuthConfigType](#)
  - the deug auth config struct.*
- struct [HSM\\_KeyStatusType](#)
  - the key status in hsm.*

## Macros

- #define [HSM\\_SUCCESS](#) 0xA55AU
- #define [HSM\\_SECURE\\_BOOT\\_ENABLE](#) (0x6EBA5B4BU)
- #define [HSM\\_SECURE\\_BOOT\\_DISABLE](#) (0U)
- #define [HSM\\_DISABLE](#) (0xACC94D73U)
- #define [HSM\\_UNNORMAL\\_MODE](#) 0xA55AA55AU
- #define [HSM\\_TEST\\_MODE](#) 0xFFFFFFFFU
- #define [HSM\\_DEVELOP\\_MODE](#) 0xBD7E7BEBU
- #define [HSM\\_MANU\\_MODE](#) 0xB93E5BE9U
- #define [HSM\\_USER\\_MODE](#) 0xA83E1369U
- #define [HSM\\_DEBUG\\_MODE](#) 0x283A0321U
- #define [HSM\\_DESTROY\\_MODE](#) 0x00000000U
- #define [HSM\\_FLASH\\_KEY\\_SLOT\\_LEN](#) 56u
- #define [HSM\\_FLASH\\_KEY\\_MAX\\_SLOT\\_NUM](#) (DFLASH\_PAGE\_SIZE / [HSM\\_FLASH\\_KEY\\_SLOT\\_LEN](#))
- #define [HSM\\_FLASH\\_KEY\\_PAGE\\_NUM](#) 2u
- #define [HSM\\_FLASH\\_PAGE\\_VALID\\_LEN](#) 8u
- #define [HSM\\_FLASH\\_PAGE\\_NUM](#) 2u

- #define `HSM_FLASH_PAGE_REVERSE_LEN` (DFLASH\_PAGE\_SIZE - (HSM\_FLASH\_KEY\_SLOT\_LEN \* HSM\_FLASH\_KEY\_MAX\_SLOT\_NUM) - HSM\_FLASH\_PAGE\_VALID\_LEN)
- #define `HSM_PAGE_VALID_TAG` (0xAAU)
- #define `HSM_SLOT_INVALID_TAG` (0xA5A5A5A5U)
- #define `HSM_KEY_DATA_VALID_TAG` (0x52525252U)
- #define `HSM_FLASH_PAGE_NUM_OFFSET` (125U)
- #define `HSM_FLASH_KEY_BASE` 0x01100000U + (HSM\_FLASH\_PAGE\_NUM\_OFFSET \* DFLASH\_PAGE\_SIZE)
- #define `HSM_FLASH_KEY_OFFSET` (HSM\_FLASH\_KEY\_BASE - DFLASH\_BASE)
- #define `HSM_RAM_KEY_SYM_MAX_NUM` 10u
- #define `HSM_RAM_KEY_ECC_MAX_NUM` 5u
- #define `HSM_RAM_KEY_RSA_MAX_NUM` 4u
- #define `HSM_RAM_KEY_MAX_NUM` HSM\_RAM\_KEY\_SYM\_MAX\_NUM + HSM\_RAM\_KEY\_ECC\_MAX\_NUM + HSM\_RAM\_KEY\_RSA\_MAX\_NUM
- #define `HSM_RAM_KEY_MEM_SIZE` 800u
- #define `OTP_KEY_CIPHER_AES_CIPHER` 0xFFFFDFFFU  
*otp value*
- #define `OTP_KEY_CIPHER_SM4_CIPHER` 0xFFFFCFFFU
- #define `RSA_2048_WITH_AES128_CBC_UPGRADE` 0xFFFFF00U
- #define `SM2_WITH_SM4_CBC_UPGRADE` 0xFFFFF01U
- #define `AES128_CMAC_WITH_AES128_CBC_UPGRADE` 0xFFFFF100U
- #define `SM4_CMAC_WITH_SM4_CBC_UPGRADE` 0xFFFFF101U
- #define `RSA_2048_WITH_AES128_CBC_VERIFY` 0xFFFFF00U
- #define `SM2_WITH_SM4_CBC_VERIFY` 0xFFFFF01U
- #define `AES128_CMAC_WITH_AES128_CBC_VERIFY` 0xFFFFF10U
- #define `SM4_CMAC_WITH_SM4_CBC_VERIFY` 0xFFFFF11U

## Typedefs

- typedef struct `HSM_SecureUpgradeType` `HSM_SecureUpgradeType`  
*Host image secure upgrade information.*
- typedef struct `HSM_ImageVerifyType` `HSM_ImageVerifyType`  
*Host image verify information.*
- typedef struct `Hsm_DhPubKeyType` `HSM_DhPubKeyType`  
*the Dh pubkey struct.*
- typedef union `Hsm_PubKeyDataType` `HSM_PubKeyDataType`  
*the public key struct for all algo.*
- typedef struct `Hsm_RsaCrtType` `HSM_RsaPrvCrtType`  
*the rsa private key struct when CRT.*
- typedef union `Hsm_PriKeyDataType` `HSM_PriKeyDataType`  
*the private key struct for all algo.*

## Enumerations

- enum `HSM_OtpKeyLevel` { `HSM_OTP_KEY_LEVEL1` = 1, `HSM_OTP_KEY_LEVEL2` = 2 }
  - enum `HSM_RndOtpKeyType` { `HSM_RND_OTP_KEY_SYM` = 1, `HSM_RND_OTP_KEY_SM2_PRIVATE` = 2, `HSM_RND_OTP_KEY_SECP256R1_PRIVATE` = 4 }
- Specifies the otp randome key type.*



- enum `HSM_OtpKeyId` {  
`CHIP_ROOT_KEY` = 0U, `DEVICE_ROOT_KEY`, `USER_ROOT_KEY`, `HSM_DEBUG_KEY`,  
`HSM_FW_VERIFY_KEY`, `HSM_ENCRYPT_KEY`, `HSM_UPGRADE_ENCRYPT_KEY`, `HSM_UPGRADE_VERIFY_KEY`,  
`HSM_PRIVATE_KEY`, `HOST_DEBUG_KEY` = 9U, `HOST_FW_VERIFY_KEY`, `HOST_ENCRYPT_KEY`,  
`HOST_UPGRADE_ENCRYPT_KEY`, `HOST_UPGRADE_VERIFY_KEY`, `HOST_PRIVATE_KEY`, `HSM_SECRET_KEY` = 15U,  
`USER_AUTH_KEY` }  
*Specifies the otp Key Index, please RM document description.*
- enum `HSM_KeyId` {  
`HSM_FLASH_KEY_SYM_0` = 0, `HSM_FLASH_KEY_SYM_1`, `HSM_FLASH_KEY_SYM_2`, `HSM_FLASH_KEY_SYM_3`,  
`HSM_FLASH_KEY_SYM_4`, `HSM_FLASH_KEY_SYM_5`, `HSM_FLASH_KEY_SYM_6`, `HSM_FLASH_KEY_SYM_7`,  
`HSM_FLASH_KEY_SYM_8`, `HSM_FLASH_KEY_SYM_9` = 9, `HSM_FLASH_KEY_ECC_0` = 10, `HSM_FLASH_KEY_ECC_1`,  
`HSM_FLASH_KEY_ECC_2`, `HSM_FLASH_KEY_ECC_3`, `HSM_FLASH_KEY_ECC_4`, `HSM_FLASH_KEY_ECC_5`,  
`HSM_FLASH_KEY_ECC_6`, `HSM_FLASH_KEY_ECC_7`, `HSM_FLASH_KEY_ECC_8`, `HSM_FLASH_KEY_ECC_9` = 19,  
`HSM_FLASH_KEY_SM2_0` = 20, `HSM_FLASH_KEY_SM2_1`, `HSM_FLASH_KEY_SM2_2`, `HSM_FLASH_KEY_SM2_3`,  
`HSM_FLASH_KEY_SM2_4`, `HSM_FLASH_KEY_SM2_5`, `HSM_FLASH_KEY_SM2_6`, `HSM_FLASH_KEY_SM2_7`,  
`HSM_FLASH_KEY_SM2_8`, `HSM_FLASH_KEY_SM2_9` = 29, `HSM_FLASH_KEY_RSA_0` = 30, `HSM_FLASH_KEY_RSA_1`,  
`HSM_FLASH_KEY_RSA_2`, `HSM_FLASH_KEY_RSA_3` = 33, `HSM_RAM_KEY_SYM_0` = 34, `HSM_RAM_KEY_SYM_1`,  
`HSM_RAM_KEY_SYM_2`, `HSM_RAM_KEY_SYM_3`, `HSM_RAM_KEY_SYM_4`, `HSM_RAM_KEY_SYM_5`,  
`HSM_RAM_KEY_SYM_6`, `HSM_RAM_KEY_SYM_7`, `HSM_RAM_KEY_SYM_8`, `HSM_RAM_KEY_SYM_9` = 43,  
`HSM_RAM_KEY_ECC_0` = 44, `HSM_RAM_KEY_ECC_1`, `HSM_RAM_KEY_ECC_2`, `HSM_RAM_KEY_ECC_3`,  
`HSM_RAM_KEY_ECC_4` = 48, `HSM_RAM_KEY_SM2_0` = 49, `HSM_RAM_KEY_SM2_1`, `HSM_RAM_KEY_SM2_2`,  
`HSM_RAM_KEY_SM2_3`, `HSM_RAM_KEY_SM2_4` = 53, `HSM_RAM_KEY_RSA_0` = 54, `HSM_RAM_KEY_RSA_1`,  
`HSM_RAM_KEY_RSA_2`, `HSM_RAM_KEY_RSA_3` = 57 }  
*Specifies the Flash Key Index.*
- enum `HSM_OtpKeyCipherAlgo` { `OTP_KEY_CIPHER_AES` = (0U), `OTP_KEY_CIPHER_SM4` = (1U) }  
*config otp ctrl bit for otp key encrypt algo*
- enum `HSM_RndAlgo` { `HSM_SM4_CTRDRBG` = (0U), `HSM_AES_CTRDRBG` = (1U) }  
*prnd algo Type*
- enum `HSM_ProcessMode` { `START_MODE` = (0x01U), `UPDATE_MODE` = (0x02U), `FINISH_MODE` = (0x04U),  
`FW_ONEPASS_MODE` = (0x07U) }  
*Process Mode Type.*
- enum `HSM_CipherDirection` { `HSM_ENCRYPTION` = 0U, `HSM_DECRYPTION` = 1U }  
*Specifies the boot type for the BOOT\_DEFINE command.*
- enum `HSM_MacDirection` { `HSM_MAC_GEN` = 0U, `HSM_MAC_VERIFY` = 1U, `HSM_MAC_TIMESTAMPED_SIGN` = 2U }  
*Specifies the gen/ver Mac value request type.*
- enum `HSM_SignDirection` { `HSM_SIGN` = 0U, `HSM_VERIFY` = 1U }  
*Specifies the gen/ver signature value request type.*
- enum `HSM_CipherMode` {  
`ECB_MODE` = 1U, `CBC_MODE` = 3U, `CFB_MODE` = 4U, `OFB_MODE` = 5U,  
`CTR_MODE` = 6U, `CMAC_MODE` = 7U }  
*Specifies the symmetric algo cipher mode, support aes/sm4.*
- enum `HSM_PaddingType` { `HSM_NOPADDING` = 0U, `HSM_PSASSA_PSS` = 1U, `HSM_PKCS7` = 2U, `HSM_ONEWITHZEROS` = 3U }  
*Specifies the sym/asym algo padding type.*

- enum `HSM_KeyAlgoType` {  
`HSM_ALGO_TYPE_NONE`, `HSM_ALGO_TYPE_RANDOM`, `HSM_ALGO_TYPE_SYM`, `HSM_ALGO_TYPE_SM2`,  
`HSM_ALGO_TYPE_ECC`, `HSM_ALGO_TYPE_DH`, `HSM_ALGO_TYPE_RSA_CRT`, `HSM_ALGO_TYPE_RSA_COMM`,  
`HSM_ALGO_TYPE_X25519`, `HSM_ALGO_TYPE_ED25519`, `HSM_ALGO_TYPE_END` }  
*Specifies the Key algo type.*
- enum `HSM_GenKeyAlgo` {  
`HSM_ALG_RANDOM` = 0U, `HSM_ALG_DES` = 1, `HSM_ALG_TDES_128` = 2, `HSM_ALG_TDES_192` = 3,  
`HSM_ALG_AES_128` = 4, `HSM_ALG_AES_192` = 5, `HSM_ALG_AES_256` = 6, `HSM_ALG_SM4` = 7,  
`HSM_ALG_SM2` = 8, `HSM_ALG_DH` = 17, `HSM_ALG_BRAINPOOLP160R1` = 18, `HSM_ALG_BRAINPOOLP192R1` = 19,  
`HSM_ALG_BRAINPOOLP224R1` = 20, `HSM_ALG_BRAINPOOLP256R1` = 21, `HSM_ALG_BRAINPOOLP320R1` = 22,  
`HSM_ALG_BRAINPOOLP384R1` = 23, `HSM_ALG_BRAINPOOLP512R1` = 24, `HSM_ALG_SECP192R1` = 25, `HSM_ALG_SECP224R1` = 26, `HSM_ALG_SECP256R1` = 27,  
`HSM_ALG_SECP384R1` = 28, `HSM_ALG_SECP521R1` = 29, `HSM_ALG_ED25519` = 30, `HSM_ALG_END` }  
*Specifies the set or generate Key for algo type.*
- enum `HSM_KeyStorageType` { `HSM_KEY_TYPE_NVM`, `HSM_KEY_TYPE_RAM` }  
*Specifies the Key storage type.*
- enum `HSM_HashAlgoType` {  
`HSM_SM3` = 0U, `HSM_SHA256` = 2U, `HSM_SHA384` = 3U, `HSM_SHA512` = 4U,  
`HSM_SHA1` = 5U, `HSM_SHA224` = 6U, `HSM_SHA512_224` = 7U, `HSM_SHA512_256` = 8U,  
`HSM_INVALID_ALG` = 0xFFU }  
*Specifies the hash type.*
- enum `HSM_SymAlgoType` { `HSM_AES_128` = 5U, `HSM_AES_256` = 7U, `HSM_SM4` = 8U }  
*Specifies the symmetric algo type.*
- enum `HSM_AsymAlgoType` { `HSM_ASYM_SM2` = 0U, `HSM_ASYM_ECDSA` = 1U, `HSM_ASYM_RSA` = 2U }  
*Specifies the asymmetric algo type.*
- enum `HSM_KdfType` { `HSM_KEY_DERIVE_KDFX963` = 0U, `HSM_KEY_DERIVE_PBKDF2` = 1U }  
*Specifies the derive algo type.*
- enum `HSM_LifeCycleType` {  
`HSM_LIFE_CYCLE_UNNORMAL_MODE` = 0x00, `HSM_LIFE_CYCLE_TEST_MODE` = 0x01, `HSM_LIFE_CYCLE_DEV_MODE` = 0x02,  
`HSM_LIFE_CYCLE_MANU_MODE` = 0x03, `HSM_LIFE_CYCLE_USER_MODE` = 0x04, `HSM_LIFE_CYCLE_DEBUG_MODE` = 0x05,  
`HSM_LIFE_CYCLE_DESTROY_MODE` = 0x06 }  
*LifeCycle type.*
- enum `HSM_IrqNum` {  
`HSM_CLR_S2H_NOTE_HOST_IRQ` = 0, `S2H_NOTE_COLLISION_HOST_IRQ`, `HOST_SET_S2H_NOTE_HSM_IRQ`,  
`S2H_NOTE_COLLISION_HSM_IRQ`, `HSM_SET_H2S_NOTE_HOST_IRQ` = 4, `H2S_NOTE_COLLISION_HOST_IRQ`, `SOC_CLEAR_H2S_NOTE_HSM_IRQ`,  
`H2S_NOTE_COLLISION_HSM_IRQ`, `HSM_IRQ_NUM` }  
*the irq.*
- enum `HSM_DebugAuthAlgoType` { `HSM_DEBUG_AUTH_ALG_SM2_WITH_SM3` = 1, `HSM_DEBUG_AUTH_ALG_ECCSECP256R1_WITH_SHA256` = 2,  
`HSM_DEBUG_AUTH_ALG_SM4_CMAC` = 3, `HSM_DEBUG_AUTH_ALG_AES128_CMAC` = 4 }  
*debug authentication algo type.*
- enum `HSM_ChallengeType` {  
`HSM_CHALLENGE_TYPE_INVALID` = 0, `HSM_CHALLENGE_TYPE_TIME_SYNC` = 1, `HSM_CHALLENGE_TYPE_EHSM_DEBUG` = 2,  
`HSM_CHALLENGE_TYPE_SHE_DEBUG` = 3, `HSM_CHALLENGE_TYPE_SOC_DEBUG` = 4, `HSM_CHALLENGE_TYPE_USER_AUTH` = 5, `HSM_CHALLENGE_TYPE_MAX` }  
*the get challenge used type.*
- enum `HSM_OtpCtrlAlgoType` { `RSA_2048_WITH_AES128_CBC` = 0, `SM2_WITH_SM4_CBC`, `AES128_CMAC_WITH_AES128_CBC`,  
`SM4_CMAC_WITH_SM4_CBC` }  
*the hsm and host upgrade algo or verify algo.*
- enum `HSM_ImageType` { `HSM_IMAGE_HOST_TYPE`, `HSM_IMAGE_HSM_TYPE` }  
*Specifies the image type host image or hsm firmware image.*

## Functions

- Hal\_StatusType [HSM\\_Hal\\_Init](#) (boolean HsmIrqEn, uint8 HsmIrqPri)  
*hsm module init function.*
- void [HSM\\_Hal\\_Deinit](#) (void)  
*hsm module deinit function.*
- Hal\_StatusType [HSM\\_Hal\\_AesCipher](#) (const [HSM\\_SymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Aes Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_Sm4Cipher](#) (const [HSM\\_SymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Sm4 Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_CipherMac](#) (const [HSM\\_CMacCfgType](#) \*CfgPtr, [HSM\\_InOutMacType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Cipher Mac Generate/Verify mac function.*
- Hal\_StatusType [HSM\\_Hal\\_RsaCipher](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Rsa Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_Sm2Cipher](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Sm2 Decrypt/Encrypt function.*
- Hal\_StatusType [HSM\\_Hal\\_EccSign](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutSignType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*ECC Sign/Verify function.*
- Hal\_StatusType [HSM\\_Hal\\_RsaSign](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutSignType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*Rsa Sign/Verify function.*
- Hal\_StatusType [HSM\\_Hal\\_Sm2Sign](#) (const [HSM\\_AsymCfgType](#) \*CfgPtr, const [HSM\\_InOutSignType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*SM2 Sign/Verify function.*
- Hal\_StatusType [HSM\\_Hal\\_Hash](#) (const [HSM\\_HashAlgoType](#) Algo, const [HSM\\_InOutType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*generate message digest(hash value) function.*
- Hal\_StatusType [HSM\\_Hal\\_HashMac](#) (const [HSM\\_HMacCfgType](#) \*CfgPtr, const [HSM\\_InOutMacType](#) \*InOutPtr, [HSM\\_↔\\_ProcessMode](#) ProMode)  
*generate message hash mac function use hash algo*
- Hal\_StatusType [HSM\\_Hal\\_SetPlainKey](#) (const [HSM\\_PlainKeyCfgType](#) \*CfgPtr)  
*set plain key to hsm mem.*
- Hal\_StatusType [HSM\\_Hal\\_GenerateKey](#) (const [HSM\\_GenKeyCfgType](#) \*CfgPtr)  
*generate key for algo, eg.aes/sm2.*
- Hal\_StatusType [HSM\\_Hal\\_DeriveKey](#) ([HSM\\_KeyId](#) ParentKeyId, [HSM\\_GenKeyAlgo](#) ParentKeyAlgo, const [HSM\\_↔\\_DeriveKeyCfgType](#) \*TargetKeyPtr)  
*drive key*
- Hal\_StatusType [HSM\\_Hal\\_SetSecretKey](#) ([HSM\\_KeyId](#) TpKeyId, [HSM\\_KeyId](#) AuthKeyId, const [HSM\\_SecretKey↔\\_CfgType](#) \*ImportKeyPtr, [HSM\\_KeyId](#) KeyId)  
*set secret key to hsm*
- Hal\_StatusType [HSM\\_Hal\\_GetSecretkey](#) ([HSM\\_KeyId](#) TpKeyId, [HSM\\_KeyId](#) AuthKeyId, [HSM\\_KeyId](#) TargetKeyId, const [HSM\\_SecretKeyCfgType](#) \*ExportKeyCfg)  
*get secret key to hsm*
- Hal\_StatusType [HSM\\_Hal\\_GetPubKeyFromPrvKey](#) ([HSM\\_KeyId](#) KeyId, uint8 \*PubKey, uint32 \*PubKeySize, [HSM\\_↔\\_GenKeyAlgo](#) Algo)  
*get pubkey form private key*
- Hal\_StatusType [HSM\\_Hal\\_GenerateDHKey](#) ([HSM\\_KeyId](#) LocalKeyId, const uint8 \*RemotePubKey, uint32 RemotePubKeySize, [HSM\\_GenKeyAlgo](#) ParKeyAlgo, const [HSM\\_GenKeyCfgType](#) \*TargetKeyPtr)  
*generate DH key.*

- Hal\_StatusType [HSM\\_Hal\\_RemoveKey](#) (HSM\_KeyId KeyId)  
*Remove Key.*
- Hal\_StatusType [HSM\\_Hal\\_GetRnd](#) (HSM\_RndAlgo Algo, uint32 RndSize, uint8 \*Rnd)  
*get random.*
- Hal\_StatusType [HSM\\_Hal\\_SetSecureBootCfg](#) (HSM\_BootCfgType \*CfgPtr)  
*set secure boot config*
- Hal\_StatusType [HSM\\_Hal\\_EnableSecureBoot](#) (void)  
*enable secureboot*
- Hal\_StatusType [HSM\\_Hal\\_DisableSecureBoot](#) (void)  
*disable secureboot*
- HSM\_LifeCycleType [HSM\\_Hal\\_GetLifeCycle](#) (void)  
*Get LifeCycle.*
- uint32 [HSM\\_Hal\\_SetLifeCycle](#) (HSM\_LifeCycleType LcIndex)  
*Set LifeCycle.*
- Hal\_StatusType [HSM\\_Hal\\_DebugAuth](#) (HSM\_DebugAuthConfigType \*CfgPtr)  
*Debug authentication.*
- Hal\_StatusType [HSM\\_Hal\\_GetChallenge](#) (HSM\_ChallengeType Type, uint8 \*ChBuf, uint32 ChSize)  
*get challenge*
- uint32 [HSM\\_Hal\\_OtpRead](#) (uint32 OtpAddr, uint32 BytesLen, uint8 \*Data)  
*read otp*
- uint32 [HSM\\_Hal\\_OtpWrite](#) (uint32 OtpAddr, uint32 BytesLen, uint8 \*Data)  
*write otp*
- void [HSM\\_Hal\\_InstallCallback](#) (const Hal\_CallbackType Callback, void \*Args)  
*irq callback.*
- boolean [HSM\\_Hal\\_GetLockState](#) (void)  
*get access hsm path lock state.*
- void [HSM\\_Hal\\_Lock](#) (void)  
*Lock access hsm path.*
- void [HSM\\_Hal\\_Unlock](#) (void)  
*unlock access hsm path.*
- Hal\_StatusType [HSM\\_Hal\\_GetKeyStatus](#) (const HSM\_KeyStatusType \*Status)  
*get key status for specific key id*
- uint32 [HSM\\_Hal\\_GetRndKey](#) (HSM\_OtpKeyLevel Level, HSM\_RndOtpKeyType KeyType, const uint8 \*KeyOut)  
*generate random key in bootloader.*
- Hal\_StatusType [HSM\\_Hal\\_SetOtpKeyCipherAlgo](#) (HSM\_OtpKeyCipherAlgo Algo)  
*Otp Hw Ctrl, Set encrypt otp key algo.*
- Hal\_StatusType [HSM\\_Hal\\_SetImageSecureUpgradeAlgo](#) (HSM\_OtpCtrlAlgoType Algo, HSM\_ImageType Type)  
*Otp Hw Ctrl, set image secure upgrade algo type.*
- Hal\_StatusType [HSM\\_Hal\\_SetImageSecureVerifyAlgo](#) (HSM\_OtpCtrlAlgoType Algo, HSM\_ImageType Type)  
*Otp Hw Ctrl, set image secure verify algo type.*
- Hal\_StatusType [HSM\\_Hal\\_HostImageSecureUpgrade](#) (const HSM\_SecureUpgradeType \*HostUpgradePara)  
*Host image upgrade.*
- Hal\_StatusType [HSM\\_Hal\\_HostImageSecureVerify](#) (const HSM\_ImageVerifyType \*HostVerifyPara)  
*Host image upgrade verify.*
- uint32 [HSM\\_Hal\\_SetOtpExternalKey](#) (HSM\_OtpKeyId KeyId, const uint8 \*ExternKey, uint32 ExternKeyLen, HSM\_OtpKeyLevel KeyLevel, uint32 KeyAttr)  
*set external key or hash value(raw dta key) to otp key slot*
- uint32 [HSM\\_Hal\\_SetOtpRndKey](#) (HSM\_OtpKeyId KeyId, HSM\_OtpKeyLevel KeyLevel, HSM\_RndOtpKeyType KeyType, uint32 KeyAttr)  
*set random key to otp key slot*
- uint32 [HSM\\_Hal\\_GetHsmFwVersion](#) (const uint32 \*Version)  
*get hsm fw version*

### 4.71.1 Detailed Description

This file provides hsm integration functions interface.

### 4.71.2 Macro Definition Documentation

#### 4.71.2.1 AES128\_CMAC\_WITH\_AES128\_CBC\_UPGRADE

```
#define AES128_CMAC_WITH_AES128_CBC_UPGRADE 0xFFFFF100U
```

Definition at line 115 of file Hsm\_Hal.h.

#### 4.71.2.2 AES128\_CMAC\_WITH\_AES128\_CBC\_VERIFY

```
#define AES128_CMAC_WITH_AES128_CBC_VERIFY 0xFFFFF10U
```

Definition at line 120 of file Hsm\_Hal.h.

#### 4.71.2.3 HSM\_DEBUG\_MODE

```
#define HSM_DEBUG_MODE 0x283A0321U
```

Definition at line 81 of file Hsm\_Hal.h.

#### 4.71.2.4 HSM\_DESTROY\_MODE

```
#define HSM_DESTROY_MODE 0x00000000U
```

Definition at line 82 of file Hsm\_Hal.h.

#### 4.71.2.5 HSM\_DEVELOP\_MODE

```
#define HSM_DEVELOP_MODE 0xBD7E7BEBU
```

Definition at line 78 of file Hsm\_Hal.h.

#### 4.71.2.6 HSM\_DISABLE

```
#define HSM_DISABLE (0xACC94D73U)
```

Definition at line 74 of file Hsm\_Hal.h.

#### 4.71.2.7 HSM\_FLASH\_KEY\_BASE

```
#define HSM_FLASH_KEY_BASE 0x01100000U + (HSM_FLASH_PAGE_NUM_OFFSET * DFLASH_PAGE_SIZE)
```

Definition at line 98 of file Hsm\_Hal.h.

#### 4.71.2.8 HSM\_FLASH\_KEY\_MAX\_SLOT\_NUM

```
#define HSM_FLASH_KEY_MAX_SLOT_NUM (DFLASH_PAGE_SIZE / HSM_FLASH_KEY_SLOT_LEN)
```

Definition at line 85 of file Hsm\_Hal.h.

#### 4.71.2.9 HSM\_FLASH\_KEY\_OFFSET

```
#define HSM_FLASH_KEY_OFFSET (HSM_FLASH_KEY_BASE - DFLASH_BASE)
```

Definition at line 99 of file Hsm\_Hal.h.

#### 4.71.2.10 HSM\_FLASH\_KEY\_PAGE\_NUM

```
#define HSM_FLASH_KEY_PAGE_NUM 2u
```

Definition at line 86 of file Hsm\_Hal.h.

#### 4.71.2.11 HSM\_FLASH\_KEY\_SLOT\_LEN

```
#define HSM_FLASH_KEY_SLOT_LEN 56u
```

Definition at line 84 of file Hsm\_Hal.h.

#### 4.71.2.12 HSM\_FLASH\_PAGE\_NUM

```
#define HSM_FLASH_PAGE_NUM 2u
```

Definition at line 89 of file Hsm\_Hal.h.

#### 4.71.2.13 HSM\_FLASH\_PAGE\_NUM\_OFFSET

```
#define HSM_FLASH_PAGE_NUM_OFFSET (125U)
```

default key page num offset for flash key

Definition at line 97 of file Hsm\_Hal.h.

#### 4.71.2.14 HSM\_FLASH\_PAGE\_REVERSE\_LEN

```
#define HSM_FLASH_PAGE_REVERSE_LEN (DFLASH_PAGE_SIZE - (HSM_FLASH_KEY_SLOT_LEN * HSM_FLASH_KEY_MAX_↵
SLOT_NUM) - HSM_FLASH_PAGE_VALID_LEN)
```

Definition at line 90 of file Hsm\_Hal.h.

#### 4.71.2.15 HSM\_FLASH\_PAGE\_VALID\_LEN

```
#define HSM_FLASH_PAGE_VALID_LEN 8u
```

Definition at line 88 of file Hsm\_Hal.h.

#### 4.71.2.16 HSM\_KEY\_DATA\_VALID\_TAG

```
#define HSM_KEY_DATA_VALID_TAG (0x52525252U)
```

Key data valid

Definition at line 95 of file Hsm\_Hal.h.

#### 4.71.2.17 HSM\_MANU\_MODE

```
#define HSM_MANU_MODE 0xB93E5BE9U
```

Definition at line 79 of file Hsm\_Hal.h.

#### 4.71.2.18 HSM\_PAGE\_VALID\_TAG

```
#define HSM_PAGE_VALID_TAG (0xAAU)
```

Page valid tag

Definition at line 93 of file Hsm\_Hal.h.

#### 4.71.2.19 HSM\_RAM\_KEY\_ECC\_MAX\_NUM

```
#define HSM_RAM_KEY_ECC_MAX_NUM 5u
```

Definition at line 102 of file Hsm\_Hal.h.

#### 4.71.2.20 HSM\_RAM\_KEY\_MAX\_NUM

```
#define HSM_RAM_KEY_MAX_NUM HSM_RAM_KEY_SYM_MAX_NUM + HSM_RAM_KEY_ECC_MAX_NUM + HSM_RAM_KEY_RSA_MAX_NUM
```

Definition at line 104 of file Hsm\_Hal.h.

#### 4.71.2.21 HSM\_RAM\_KEY\_MEM\_SIZE

```
#define HSM_RAM_KEY_MEM_SIZE 800u
```

Definition at line 105 of file Hsm\_Hal.h.

#### 4.71.2.22 HSM\_RAM\_KEY\_RSA\_MAX\_NUM

```
#define HSM_RAM_KEY_RSA_MAX_NUM 4u
```

Definition at line 103 of file Hsm\_Hal.h.

#### 4.71.2.23 HSM\_RAM\_KEY\_SYM\_MAX\_NUM

```
#define HSM_RAM_KEY_SYM_MAX_NUM 10u
```

Definition at line 101 of file Hsm\_Hal.h.



#### 4.71.2.24 HSM\_SECURE\_BOOT\_DISABLE

```
#define HSM_SECURE_BOOT_DISABLE (0U)
```

Definition at line 73 of file Hsm\_Hal.h.

#### 4.71.2.25 HSM\_SECURE\_BOOT\_ENABLE

```
#define HSM_SECURE_BOOT_ENABLE (0x6EBA5B4BU)
```

Definition at line 72 of file Hsm\_Hal.h.

#### 4.71.2.26 HSM\_SLOT\_INVALID\_TAG

```
#define HSM_SLOT_INVALID_TAG (0xA5A5A5A5U)
```

Slot invalid tag

Definition at line 94 of file Hsm\_Hal.h.

#### 4.71.2.27 HSM\_SUCCESS

```
#define HSM_SUCCESS 0xA55AU
```

Definition at line 70 of file Hsm\_Hal.h.

#### 4.71.2.28 HSM\_TEST\_MODE

```
#define HSM_TEST_MODE 0xFFFFFFFFU
```

Definition at line 77 of file Hsm\_Hal.h.

#### 4.71.2.29 HSM\_UNNORMAL\_MODE

```
#define HSM_UNNORMAL_MODE 0xA55AA55AU
```

Definition at line 76 of file Hsm\_Hal.h.

#### 4.71.2.30 HSM\_USER\_MODE

```
#define HSM_USER_MODE 0xA83E1369U
```

Definition at line 80 of file Hsm\_Hal.h.

#### 4.71.2.31 OTP\_KEY\_CIPHER\_AES\_CIPHER

```
#define OTP_KEY_CIPHER_AES_CIPHER 0xFFFFDFFFU
```

otp value

Definition at line 110 of file Hsm\_Hal.h.

#### 4.71.2.32 OTP\_KEY\_CIPHER\_SM4\_CIPHER

```
#define OTP_KEY_CIPHER_SM4_CIPHER 0xFFFFCFFFU
```

Definition at line 111 of file Hsm\_Hal.h.

#### 4.71.2.33 RSA\_2048\_WITH\_AES128\_CBC\_UPGRADE

```
#define RSA_2048_WITH_AES128_CBC_UPGRADE 0xFFFFF00U
```

Definition at line 113 of file Hsm\_Hal.h.

#### 4.71.2.34 RSA\_2048\_WITH\_AES128\_CBC\_VERIFY

```
#define RSA_2048_WITH_AES128_CBC_VERIFY 0xFFFFF00U
```

Definition at line 118 of file Hsm\_Hal.h.

#### 4.71.2.35 SM2\_WITH\_SM4\_CBC\_UPGRADE

```
#define SM2_WITH_SM4_CBC_UPGRADE 0xFFFFF01U
```

Definition at line 114 of file Hsm\_Hal.h.

#### 4.71.2.36 SM2\_WITH\_SM4\_CBC\_VERIFY

```
#define SM2_WITH_SM4_CBC_VERIFY 0xFFFFF01U
```

Definition at line 119 of file Hsm\_Hal.h.

#### 4.71.2.37 SM4\_CMAC\_WITH\_SM4\_CBC\_UPGRADE

```
#define SM4_CMAC_WITH_SM4_CBC_UPGRADE 0xFFFFF101U
```

Definition at line 116 of file Hsm\_Hal.h.

#### 4.71.2.38 SM4\_CMAC\_WITH\_SM4\_CBC\_VERIFY

```
#define SM4_CMAC_WITH_SM4_CBC_VERIFY 0xFFFFF11U
```

Definition at line 121 of file Hsm\_Hal.h.

### 4.71.3 Typedef Documentation

#### 4.71.3.1 HSM\_DhPubKeyType

```
typedef struct Hsm_DhPubKeyType_ HSM_DhPubKeyType
```

the Dh pubkey struct.

#### 4.71.3.2 HSM\_ImageVerifyType

```
typedef struct HSM_ImageVerifyType__ HSM_ImageVerifyType
```

Host image verify information.

#### 4.71.3.3 HSM\_PriKeyDataType

```
typedef union Hsm_PriKeyDataType_ HSM_PriKeyDataType
```

the private key struct for all algo.

#### 4.71.3.4 HSM\_PubKeyDataType

```
typedef union Hsm_PubKeyDataType_ HSM_PubKeyDataType
```

the public key struct for all algo.

#### 4.71.3.5 HSM\_RsaPrvCrtType

```
typedef struct Hsm_RsaCrtType_ HSM_RsaPrvCrtType
```

the rsa private key struct when CRT.

#### 4.71.3.6 HSM\_SecureUpgradeType

```
typedef struct HSM_SecureUpgradeType__ HSM_SecureUpgradeType
```

Host image secure upgrade information.

### 4.71.4 Enumeration Type Documentation

#### 4.71.4.1 HSM\_AsymAlgoType

```
enum HSM_AsymAlgoType
```

Specifies the asymmetric algo type.

##### Enumerator

|                |  |
|----------------|--|
| HSM_ASYM_SM2   |  |
| HSM_ASYM_ECDSA |  |
| HSM_ASYM_RSA   |  |

Definition at line 507 of file Hsm\_Hal.h.

#### 4.71.4.2 HSM\_ChallengeType

```
enum HSM_ChallengeType
```

the get challenge used type.

## Enumerator

|                               |  |
|-------------------------------|--|
| HSM_CHALLENGE_TYPE_INVALID    |  |
| HSM_CHALLENGE_TYPE_TIME_SYNC  |  |
| HSM_CHALLENGE_TYPE_EHSM_DEBUG |  |
| HSM_CHALLENGE_TYPE_SHE_DEBUG  |  |
| HSM_CHALLENGE_TYPE_SOC_DEBUG  |  |
| HSM_CHALLENGE_TYPE_USER_AUTH  |  |
| HSM_CHALLENGE_TYPE_MAX        |  |

Definition at line 873 of file Hsm\_Hal.h.

## 4.71.4.3 HSM\_CipherDirection

enum [HSM\\_CipherDirection](#)

Specifies the boot type for the BOOT\_DEFINE command.

## Enumerator

|                |                 |
|----------------|-----------------|
| HSM_ENCRYPTION | encrypt request |
| HSM_DECRYPTION | decrypt request |

Definition at line 358 of file Hsm\_Hal.h.

## 4.71.4.4 HSM\_CipherMode

enum [HSM\\_CipherMode](#)

Specifies the symmetric algo cipher mode, support aes/sm4.

## Enumerator

|           |                                               |
|-----------|-----------------------------------------------|
| ECB_MODE  | AES or sm4 encrypt/decrypt algo ECB_MODE      |
| CBC_MODE  | AES or sm4 encrypt/decrypt algo CBC_MODE      |
| CFB_MODE  | AES or sm4 encrypt/decrypt algo CFB_MODE      |
| OFB_MODE  | AES or sm4 encrypt/decrypt algo OFB_MODE      |
| CTR_MODE  | AES or sm4 encrypt/decrypt algo CTR_MODE      |
| CMAC_MODE | AES or sm4 generate/verify mac algo CMAC_MODE |

Definition at line 386 of file Hsm\_Hal.h.

## 4.71.4.5 HSM\_DebugAuthAlgoType

enum [HSM\\_DebugAuthAlgoType](#)

debug authentication algo type.

#### Enumerator

|                                             |  |
|---------------------------------------------|--|
| HSM_DEBUG_AUTH_ALG_SM2_WITH_SM3             |  |
| HSM_DEBUG_AUTH_ALG_ECCSECP256R1_WITH_SHA256 |  |
| HSM_DEBUG_AUTH_ALG_SM4_CM4C                 |  |
| HSM_DEBUG_AUTH_ALG_AES128_CM4C              |  |

Definition at line 862 of file Hsm\_Hal.h.

#### 4.71.4.6 HSM\_GenKeyAlgo

enum [HSM\\_GenKeyAlgo](#)

Specifies the set or generate Key for algo type.

#### Enumerator

|                         |  |
|-------------------------|--|
| HSM_ALG_RANDOM          |  |
| HSM_ALG_DES             |  |
| HSM_ALG_TDES_128        |  |
| HSM_ALG_TDES_192        |  |
| HSM_ALG_AES_128         |  |
| HSM_ALG_AES_192         |  |
| HSM_ALG_AES_256         |  |
| HSM_ALG_SM4             |  |
| HSM_ALG_SM2             |  |
| HSM_ALG_DH              |  |
| HSM_ALG_BRAINPOOLP160R1 |  |
| HSM_ALG_BRAINPOOLP192R1 |  |
| HSM_ALG_BRAINPOOLP224R1 |  |
| HSM_ALG_BRAINPOOLP256R1 |  |
| HSM_ALG_BRAINPOOLP320R1 |  |
| HSM_ALG_BRAINPOOLP384R1 |  |
| HSM_ALG_BRAINPOOLP512R1 |  |
| HSM_ALG_SECP192R1       |  |
| HSM_ALG_SECP224R1       |  |
| HSM_ALG_SECP256R1       |  |
| HSM_ALG_SECP384R1       |  |
| HSM_ALG_SECP521R1       |  |
| HSM_ALG_ED25519         |  |
| HSM_ALG_END             |  |

Definition at line 429 of file Hsm\_Hal.h.

#### 4.71.4.7 HSM\_HashAlgoType

enum [HSM\\_HashAlgoType](#)

Specifies the hash type.

#### Enumerator

|                 |  |
|-----------------|--|
| HSM_SM3         |  |
| HSM_SHA256      |  |
| HSM_SHA384      |  |
| HSM_SHA512      |  |
| HSM_SHA1        |  |
| HSM_SHA224      |  |
| HSM_SHA512_224  |  |
| HSM_SHA512_256  |  |
| HSM_INVALID_ALG |  |

Definition at line 481 of file Hsm\_Hal.h.

#### 4.71.4.8 HSM\_ImageType

```
enum HSM_ImageType
```

Specifies the image type host image or hsm firmware image.

#### Enumerator

|                     |  |
|---------------------|--|
| HSM_IMAGE_HOST_TYPE |  |
| HSM_IMAGE_HSM_TYPE  |  |

Definition at line 923 of file Hsm\_Hal.h.

#### 4.71.4.9 HSM\_IrqNum

```
enum HSM_IrqNum
```

the irq.

#### Enumerator

|                             |  |
|-----------------------------|--|
| HSM_CLR_S2H_NOTE_HOST_IRQ   |  |
| S2H_NOTE_COLLISION_HOST_IRQ |  |
| HOST_SET_S2H_NOTE_HSM_IRQ   |  |
| S2H_NOTE_COLLISION_HSM_IRQ  |  |
| HSM_SET_H2S_NOTE_HOST_IRQ   |  |
| H2S_NOTE_COLLISION_HOST_IRQ |  |
| SOC_CLEAR_H2S_NOTE_HSM_IRQ  |  |
| H2S_NOTE_COLLISION_HSM_IRQ  |  |
| HSM_IRQ_NUM                 |  |

Definition at line 844 of file Hsm\_Hal.h.

#### 4.71.4.10 HSM\_KdfType

enum [HSM\\_KdfType](#)

Specifies the derive algo type.

##### Enumerator

|                        |  |
|------------------------|--|
| HSM_KEY_DERIVE_KDFX963 |  |
| HSM_KEY_DERIVE_PBKDF2  |  |

Definition at line 517 of file Hsm\_Hal.h.

#### 4.71.4.11 HSM\_KeyAlgoType

enum [HSM\\_KeyAlgoType](#)

Specifies the Key algo type.

##### Enumerator

|                        |  |
|------------------------|--|
| HSM_ALGO_TYPE_NONE     |  |
| HSM_ALGO_TYPE_RANDOM   |  |
| HSM_ALGO_TYPE_SYM      |  |
| HSM_ALGO_TYPE_SM2      |  |
| HSM_ALGO_TYPE_ECC      |  |
| HSM_ALGO_TYPE_DH       |  |
| HSM_ALGO_TYPE_RSA_CRT  |  |
| HSM_ALGO_TYPE_RSA_COMM |  |
| HSM_ALGO_TYPE_X25519   |  |
| HSM_ALGO_TYPE_ED25519  |  |
| HSM_ALGO_TYPE_END      |  |

Definition at line 410 of file Hsm\_Hal.h.

#### 4.71.4.12 HSM\_KeyId

enum [HSM\\_KeyId](#)

Specifies the Flash Key Index.



## Enumerator

|                          |  |
|--------------------------|--|
| HSM_FLASH_KEY_SYM↔<br>_0 |  |
| HSM_FLASH_KEY_SYM↔<br>_1 |  |
| HSM_FLASH_KEY_SYM↔<br>_2 |  |
| HSM_FLASH_KEY_SYM↔<br>_3 |  |
| HSM_FLASH_KEY_SYM↔<br>_4 |  |
| HSM_FLASH_KEY_SYM↔<br>_5 |  |
| HSM_FLASH_KEY_SYM↔<br>_6 |  |
| HSM_FLASH_KEY_SYM↔<br>_7 |  |
| HSM_FLASH_KEY_SYM↔<br>_8 |  |
| HSM_FLASH_KEY_SYM↔<br>_9 |  |
| HSM_FLASH_KEY_ECC↔<br>_0 |  |
| HSM_FLASH_KEY_ECC↔<br>_1 |  |
| HSM_FLASH_KEY_ECC↔<br>_2 |  |
| HSM_FLASH_KEY_ECC↔<br>_3 |  |
| HSM_FLASH_KEY_ECC↔<br>_4 |  |
| HSM_FLASH_KEY_ECC↔<br>_5 |  |
| HSM_FLASH_KEY_ECC↔<br>_6 |  |
| HSM_FLASH_KEY_ECC↔<br>_7 |  |
| HSM_FLASH_KEY_ECC↔<br>_8 |  |
| HSM_FLASH_KEY_ECC↔<br>_9 |  |
| HSM_FLASH_KEY_SM2↔<br>_0 |  |
| HSM_FLASH_KEY_SM2↔<br>_1 |  |
| HSM_FLASH_KEY_SM2↔<br>_2 |  |
| HSM_FLASH_KEY_SM2↔<br>_3 |  |
| HSM_FLASH_KEY_SM2↔<br>_4 |  |
| HSM_FLASH_KEY_SM2↔<br>_5 |  |
| HSM_FLASH_KEY_SM2↔<br>_6 |  |

## Enumerator

|                          |  |
|--------------------------|--|
| HSM_FLASH_KEY_SM2↔<br>_7 |  |
| HSM_FLASH_KEY_SM2↔<br>_8 |  |
| HSM_FLASH_KEY_SM2↔<br>_9 |  |
| HSM_FLASH_KEY_RSA↔<br>_0 |  |
| HSM_FLASH_KEY_RSA↔<br>_1 |  |
| HSM_FLASH_KEY_RSA↔<br>_2 |  |
| HSM_FLASH_KEY_RSA↔<br>_3 |  |
| HSM_RAM_KEY_SYM_0        |  |
| HSM_RAM_KEY_SYM_1        |  |
| HSM_RAM_KEY_SYM_2        |  |
| HSM_RAM_KEY_SYM_3        |  |
| HSM_RAM_KEY_SYM_4        |  |
| HSM_RAM_KEY_SYM_5        |  |
| HSM_RAM_KEY_SYM_6        |  |
| HSM_RAM_KEY_SYM_7        |  |
| HSM_RAM_KEY_SYM_8        |  |
| HSM_RAM_KEY_SYM_9        |  |
| HSM_RAM_KEY_ECC_0        |  |
| HSM_RAM_KEY_ECC_1        |  |
| HSM_RAM_KEY_ECC_2        |  |
| HSM_RAM_KEY_ECC_3        |  |
| HSM_RAM_KEY_ECC_4        |  |
| HSM_RAM_KEY_SM2_0        |  |
| HSM_RAM_KEY_SM2_1        |  |
| HSM_RAM_KEY_SM2_2        |  |
| HSM_RAM_KEY_SM2_3        |  |
| HSM_RAM_KEY_SM2_4        |  |
| HSM_RAM_KEY_RSA_0        |  |
| HSM_RAM_KEY_RSA_1        |  |
| HSM_RAM_KEY_RSA_2        |  |
| HSM_RAM_KEY_RSA_3        |  |

Definition at line 170 of file Hsm\_Hal.h.

## 4.71.4.13 HSM\_KeyStorageType

```
enum HSM_KeyStorageType
```

Specifies the Key storage type.

## Enumerator

|                  |                |
|------------------|----------------|
| HSM_KEY_TYPE_NVM | store in flash |
| HSM_KEY_TYPE_RAM | store in ram   |

Definition at line 472 of file Hsm\_Hal.h.

#### 4.71.4.14 HSM\_LifeCycleType

enum [HSM\\_LifeCycleType](#)

LifeCycle type.

##### Enumerator

|                              |  |
|------------------------------|--|
| HSM_LIFE_CYCLE_UNNORMAL_MODE |  |
| HSM_LIFE_CYCLE_TEST_MODE     |  |
| HSM_LIFE_CYCLE_DEV_MODE      |  |
| HSM_LIFE_CYCLE_MANU_MODE     |  |
| HSM_LIFE_CYCLE_USER_MODE     |  |
| HSM_LIFE_CYCLE_DEBUG_MODE    |  |
| HSM_LIFE_CYCLE_DESTORY_MODE  |  |

Definition at line 526 of file Hsm\_Hal.h.

#### 4.71.4.15 HSM\_MacDirection

enum [HSM\\_MacDirection](#)

Specifies the gen/ver Mac value request type.

##### Enumerator

|                          |                     |
|--------------------------|---------------------|
| HSM_MAC_GEN              | genrate mac request |
| HSM_MAC_VERIFY           | verify mac request  |
| HSM_MAC_TIMESTAMPED_SIGN |                     |

Definition at line 367 of file Hsm\_Hal.h.

#### 4.71.4.16 HSM\_OtpCtrlAlgoType

enum [HSM\\_OtpCtrlAlgoType](#)

the hsm and host upgrade algo or verify algo.

##### Enumerator

|                             |                                                                          |
|-----------------------------|--------------------------------------------------------------------------|
| RSA_2048_WITH_AES128_CBC    | *indicate rsa 1024 for signature(e,n) and AES128 CBC for encrypt/decrypt |
| SM2_WITH_SM4_CBC            | *indicate sm2 for signature and sm4 CBC for encrypt/decrypt              |
| AES128_CMAC_WITH_AES128_CBC | *indicate aes128_cmac for signature and aes128 CBC for encrypt/decrypt   |
| SM4_CMAC_WITH_SM4_CBC       | *indicate sm4_cmac for signature and sm4 CBC for encrypt/decrypt         |

Definition at line 913 of file Hsm\_Hal.h.

#### 4.71.4.17 HSM\_OtpKeyCipherAlgo

enum [HSM\\_OtpKeyCipherAlgo](#)

config otp ctrl bit for otp key encrypt algo

##### Enumerator

|                    |                                |
|--------------------|--------------------------------|
| OTP_KEY_CIPHER_AES | the encrypt key use AES128 ECB |
| OTP_KEY_CIPHER_SM4 | the encrypt key use SM4 ECB    |

Definition at line 329 of file Hsm\_Hal.h.

#### 4.71.4.18 HSM\_OtpKeyId

enum [HSM\\_OtpKeyId](#)

Specifies the otp Key Index, please RM document description.

##### Enumerator

|                          |  |
|--------------------------|--|
| CHIP_ROOT_KEY            |  |
| DEVICE_ROOT_KEY          |  |
| USER_ROOT_KEY            |  |
| HSM_DEBUG_KEY            |  |
| HSM_FW_VERIFY_KEY        |  |
| HSM_ENCRYPT_KEY          |  |
| HSM_UPGRADE_ENCRYPT_KEY  |  |
| HSM_UPGRADE_VERIFY_KEY   |  |
| HSM_PRIVATE_KEY          |  |
| HOST_DEBUG_KEY           |  |
| HOST_FW_VERIFY_KEY       |  |
| HOST_ENCRYPT_KEY         |  |
| HOST_UPGRADE_ENCRYPT_KEY |  |
| HOST_UPGRADE_VERIFY_KEY  |  |
| HOST_PRIVATE_KEY         |  |
| HSM_SECRET_KEY           |  |
| USER_AUTH_KEY            |  |

Definition at line 146 of file Hsm\_Hal.h.

#### 4.71.4.19 HSM\_OtpKeyLevel

enum [HSM\\_OtpKeyLevel](#)

Specifies the otp Key level.

**Enumerator**

|                    |                              |
|--------------------|------------------------------|
| HSM_OTP_KEY_LEVEL1 | key levele maybe use hsm fw  |
| HSM_OTP_KEY_LEVEL2 | key levele maybe use host fw |

Definition at line 127 of file Hsm\_Hal.h.

**4.71.4.20 HSM\_PaddingType**

enum [HSM\\_PaddingType](#)

Specifies the sym/asym algo padding type.

**Enumerator**

|                  |  |
|------------------|--|
| HSM_NOPADDING    |  |
| HSM_PSASSA_PSS   |  |
| HSM_PKCS7        |  |
| HSM_ONEWITHZEROS |  |

Definition at line 399 of file Hsm\_Hal.h.

**4.71.4.21 HSM\_ProcessMode**

enum [HSM\\_ProcessMode](#)

Process Mode Type.

**Enumerator**

|                 |                                                   |
|-----------------|---------------------------------------------------|
| START_MODE      | init phase                                        |
| UPDATE_MODE     | update phase, will more loop to update input data |
| FINISH_MODE     | finish phase                                      |
| FW_ONEPASS_MODE | one phase flow, for input data size < 1024 bytes  |

Definition at line 347 of file Hsm\_Hal.h.

**4.71.4.22 HSM\_RndAlgo**

enum [HSM\\_RndAlgo](#)

prnd algo Type

## Enumerator

|                 |            |
|-----------------|------------|
| HSM_SM4_CTRDRBG | use SM4    |
| HSM_AES_CTRDRBG | use AES128 |

Definition at line 338 of file Hsm\_Hal.h.

## 4.71.4.23 HSM\_RndOtpKeyType

enum [HSM\\_RndOtpKeyType](#)

Specifies the otp randome key type.

## Enumerator

|                                   |                                              |
|-----------------------------------|----------------------------------------------|
| HSM_RND_OTP_KEY_SYM               | random key is sym key                        |
| HSM_RND_OTP_KEY_SM2_PRIVATE       | random key is sm2 private key                |
| HSM_RND_OTP_KEY_SECP256R1_PRIVATE | random key is ecc secp256r1 type private key |

Definition at line 136 of file Hsm\_Hal.h.

## 4.71.4.24 HSM\_SignDirection

enum [HSM\\_SignDirection](#)

Specifies the gen/ver signature value request type.

## Enumerator

|            |                           |
|------------|---------------------------|
| HSM_SIGN   | genrate signature request |
| HSM_VERIFY | verify signature request  |

Definition at line 377 of file Hsm\_Hal.h.

## 4.71.4.25 HSM\_SymAlgoType

enum [HSM\\_SymAlgoType](#)

Specifies the symmetric algo type.

## Enumerator

|             |  |
|-------------|--|
| HSM_AES_128 |  |
| HSM_AES_256 |  |
| HSM_SM4     |  |

Definition at line 497 of file Hsm\_Hal.h.

#### 4.71.5 Function Documentation

##### 4.71.5.1 HSM\_Hal\_AesCipher()

```
Hal_StatusType HSM_Hal_AesCipher (
 const HSM_SymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Aes Decrypt/Encrypt function.

##### Note

Function ID: DES\_HSM\_API\_003

##### Parameters

|    |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | Aes Algo config KeyId: Key Index. Iv: init vector. IvLen: iv Len Padding:data padding type. HSM_NOPADDING = 0U, HSM_PSASSA_PSS = 1U, HSM_PKCS7 = 2U, HSM_ONEWITHZEROS = 3U CipherDir :request opeartion, eg.decrypt/encrypt. HSM_ENCRYPTION = 0U HSM_DECRYPTION = 1U CMode:for sym algo and CMAC algo, eg.ECB/CBC. ECB_MODE = 1U CBC_MODE = 3U CFB_MODE = 4U OFB_MODE = 5U CTR_MODE = 6U SymAlgo: symmetric algo type, eg.AES_128/AES_256 HSM_AES_128 = 5U, HSM_AES_256 = 7U, |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length                                                                                                                                                                                                                                                                                                                                                    |
| in | <i>ProMode</i>  | process mode. START_MODE = (0x01U), UPDATE_MODE = (0x02U), FINISH_MODE = (0x04U), FW_ONEPASS_MODE = (0x07U),                                                                                                                                                                                                                                                                                                                                                                  |

##### Returns

Hal\_StatusType: STATUS\_ERROR: error happend in Cipher operation; STATUS\_SUCCESS: successfull

Definition at line 2328 of file Hsm\_Hal.c.

##### 4.71.5.2 HSM\_Hal\_CipherMac()

```
Hal_StatusType HSM_Hal_CipherMac (
 const HSM_CMacCfgType * CfgPtr,
 HSM_InOutMacType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Cipher Mac Generate/Verify mac function.

##### Note

Function ID: DES\_HSM\_API\_010

## Parameters

|    |                 |                                                                                                                                                                                                              |
|----|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | Algo config KeyId: Key Index. MacDir :request opeartion, eg.decrypt/encrypt. HSM_MAC_GEN = 0U HSM_MAC_VERIFY = 1U SymAlgo: symmetric algo type, eg.AES_128/SM4 HSM_AES_128 = 6U, HSM_SM4 = 8U,               |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length MacInBuf: store Mac value when verify MacInBufLen: Mac length Vry: verify result. |
| in | <i>ProMode</i>  | process mode. START_MODE = (0x01U), UPDATE_MODE = (0x02U), FINISH_MODE = (0x04U), FW_ONEPASS_MODE = (0x07U),                                                                                                 |

## Returns

Hal\_StatusType: STATUS\_ERROR: error happend in Cipher operation; STATUS\_SUCCESS: successfull

Definition at line 2373 of file Hsm\_Hal.c.

## 4.71.5.3 HSM\_Hal\_DebugAuth()

```
Hal_StatusType HSM_Hal_DebugAuth (
 HSM_DebugAuthConfigType * CfgPtr)
```

Debug authentication.

## Note

Function ID: DES\_HSM\_API\_028

## Parameters

|    |               |                                      |
|----|---------------|--------------------------------------|
| in | <i>CfgPtr</i> | Debug Auth config information struct |
|----|---------------|--------------------------------------|

## Returns

op status

Definition at line 3263 of file Hsm\_Hal.c.

## 4.71.5.4 HSM\_Hal\_Deinit()

```
void HSM_Hal_Deinit (
 void)
```

hsm module deinit function.

## Note

Function ID: DES\_HSM\_API\_002



Returns

op status

Definition at line 2307 of file Hsm\_Hal.c.

4.71.5.5 HSM\_Hal\_DeriveKey()

```
Hal_StatusType HSM_Hal_DeriveKey (
 HSM_KeyId ParentKeyId,
 HSM_GenKeyAlgo ParentKeyAlgo,
 const HSM_DeriveKeyCfgType * TargetKeyPtr)
```

drive key

Note

Function ID: DES\_HSM\_API\_015

Parameters

|    |                     |                    |
|----|---------------------|--------------------|
| in | <i>ParentKeyPtr</i> | parent key config. |
| in | <i>TargetKeyPtr</i> | Target key config. |

Returns

op status

Definition at line 2968 of file Hsm\_Hal.c.

4.71.5.6 HSM\_Hal\_DisableSecureBoot()

```
Hal_StatusType HSM_Hal_DisableSecureBoot (
 void)
```

disable secureboot

Note

Function ID: DES\_HSM\_API\_024

Returns

op status

Definition at line 3312 of file Hsm\_Hal.c.

#### 4.71.5.7 HSM\_Hal\_EccSign()

```
Hal_StatusType HSM_Hal_EccSign (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutSignType * InOutPtr,
 HSM_ProcessMode ProMode)
```

ECC Sign/Verify function.

##### Note

Function ID: DES\_HSM\_API\_007

##### Parameters

|    |                 |                                                                                                                                                                                                                            |
|----|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | ecc Algo config.                                                                                                                                                                                                           |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length SignInBuf: store signature value when verify SignInBufLen: signature length Vry: verify result. |
| in | <i>ProMode</i>  | process mode.                                                                                                                                                                                                              |

##### Returns

op status

Definition at line 2622 of file Hsm\_Hal.c.

#### 4.71.5.8 HSM\_Hal\_EnableSecureBoot()

```
Hal_StatusType HSM_Hal_EnableSecureBoot (
 void)
```

enable secureboot

##### Note

Function ID: DES\_HSM\_API\_023

##### Returns

op status

Definition at line 3304 of file Hsm\_Hal.c.

#### 4.71.5.9 HSM\_Hal\_GenerateDHKey()

```
Hal_StatusType HSM_Hal_GenerateDHKey (
 HSM_KeyId LocalKeyId,
 const uint8 * RemotePubKey,
 uint32 RemotePubKeySize,
 HSM_GenKeyAlgo ParKeyAlgo,
 const HSM_GenKeyCfgType * TargetKeyPtr)
```

generate DH key.

##### Note

Function ID: DES\_HSM\_API\_019

**Parameters**

|     |                         |                                    |
|-----|-------------------------|------------------------------------|
| in  | <i>LocalKeyId</i>       | local key id.                      |
| out | <i>RemotePubKey</i>     | the pubkey key buffer from remote. |
| out | <i>RemotePubKeySize</i> | the remote pubkey key size.        |
| in  | <i>TargetKeyPtr</i>     | generate dh key config.            |

**Returns**

op status

Definition at line 3164 of file Hsm\_Hal.c.

**4.71.5.10 HSM\_Hal\_GenerateKey()**

```
Hal_StatusType HSM_Hal_GenerateKey (
 const HSM_GenKeyCfgType * CfgPtr)
```

generate key for algo, eg.aes/sm2.

**Note**

Function ID: DES\_HSM\_API\_014

**Parameters**

|    |               |                      |
|----|---------------|----------------------|
| in | <i>CfgPtr</i> | generate key config. |
|----|---------------|----------------------|

**Returns**

op status

Definition at line 2926 of file Hsm\_Hal.c.

**4.71.5.11 HSM\_Hal\_GetChallenge()**

```
Hal_StatusType HSM_Hal_GetChallenge (
 HSM_ChallengeType Type,
 uint8 * ChBuf,
 uint32 ChSize)
```

get challenge

**Note**

Function ID: DES\_HSM\_API\_029

**Parameters**

|     |               |                    |
|-----|---------------|--------------------|
| in  | <i>Type</i>   | challenge type     |
| out | <i>ChBuf</i>  | challenge buf      |
| in  | <i>ChSize</i> | challenge buf size |

**Returns**

op status

Definition at line 3279 of file Hsm\_Hal.c.

**4.71.5.12 HSM\_Hal\_GetHsmFwVersion()**

```
uint32 HSM_Hal_GetHsmFwVersion (
 const uint32 * Version)
```

get hsm fw version

**Note**

Function ID: DES\_HSM\_API\_038

**Parameters**

|     |                |                 |
|-----|----------------|-----------------|
| out | <i>Version</i> | hsm fw version. |
|-----|----------------|-----------------|

**Returns**

operate status  
operate status

Definition at line 3377 of file Hsm\_Hal.c.

**4.71.5.13 HSM\_Hal\_GetKeyStatus()**

```
Hal_StatusType HSM_Hal_GetKeyStatus (
 const HSM_KeyStatusType * Status)
```

get key status for specific key id

**Note**

Function ID: DES\_HSM\_API\_027

Parameters

|    |        |                                                                                                                                                                                                                                                |
|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | Status | key status struct TargetKeyId: the target id get it status CertificationKeyId: certification key id for status information CertificationAuthSize: size CertificationAuth:auth value KeyStatusSize: status size for target id KeyStatus: status |
|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Returns

op status

Definition at line 3217 of file Hsm\_Hal.c.

4.71.5.14 HSM\_Hal\_GetLifeCycle()

```
HSM_LifeCycleType HSM_Hal_GetLifeCycle (
 void)
```

Get LifeCycle.

Note

Function ID: DES\_HSM\_API\_040

Returns

op status

Definition at line 3233 of file Hsm\_Hal.c.

4.71.5.15 HSM\_Hal\_GetLockState()

```
boolean HSM_Hal_GetLockState (
 void)
```

get access hsm path lock state.

Note

Function ID: DES\_HSM\_API\_005

Returns

op status

Definition at line 2246 of file Hsm\_Hal.c.

4.71.5.16 HSM\_Hal\_GetPubKeyFromPrvKey()

```
Hal_StatusType HSM_Hal_GetPubKeyFromPrvKey (
 HSM_KeyId KeyId,
 uint8 * PubKey,
 uint32 * PubKeySize,
 HSM_GenKeyAlgo Algo)
```

get pubkey form private key

Note

Function ID: DES\_HSM\_API\_026

**Parameters**

|     |                   |                        |
|-----|-------------------|------------------------|
| in  | <i>KeyId</i>      | Target key id.         |
| out | <i>PubKey</i>     | the pubkey key buffer. |
| out | <i>PubKeySize</i> | the pubkey key size.   |
| in  | <i>Algo</i>       | target key algo.       |

**Returns**

op status

Definition at line 3142 of file Hsm\_Hal.c.

**4.71.5.17 HSM\_Hal\_GetRnd()**

```
Hal_StatusType HSM_Hal_GetRnd (
 HSM_RndAlgo Algo,
 uint32 RndSize,
 uint8 * Rnd)
```

get random.

**Note**

Function ID: DES\_HSM\_API\_021

**Parameters**

|     |                |                                    |
|-----|----------------|------------------------------------|
| in  | <i>Algo</i>    | trng or prng use algo type         |
| in  | <i>RndSize</i> | random size, max size is 128 bytes |
| out | <i>Rnd</i>     | random information                 |

**Returns**

op status

Definition at line 2319 of file Hsm\_Hal.c.

**4.71.5.18 HSM\_Hal\_GetRndKey()**

```
uint32 HSM_Hal_GetRndKey (
 HSM_OtpKeyLevel Level,
 HSM_RndOtpKeyType KeyType,
 const uint8 * KeyOut)
```

generate random key in bootloader.

**Note**

Function ID: DES\_HSM\_API\_018

## Parameters

|     |                |                   |
|-----|----------------|-------------------|
| in  | <i>Level</i>   | key level.        |
| in  | <i>KeyType</i> | support key type. |
| out | <i>KeyOut</i>  | key out buffer.   |

## Returns

op status

Definition at line 3535 of file Hsm\_Hal.c.

## 4.71.5.19 HSM\_Hal\_GetSecretkey()

```
Hal_StatusType HSM_Hal_GetSecretkey (
 HSM_KeyId TpKeyId,
 HSM_KeyId AuthKeyId,
 HSM_KeyId TargetKeyId,
 const HSM_SecretKeyCfgType * ExportKeyCfg)
```

get secret key to hsm

## Note

Function ID: DES\_HSM\_API\_017

## Parameters

|    |                     |                        |
|----|---------------------|------------------------|
| in | <i>TpKeyId</i>      | transport key id.      |
| in | <i>AuthKeyId</i>    | auth key id.           |
| in | <i>TargetKeyId</i>  | Target key id.         |
| in | <i>ExportKeyCfg</i> | export key blob config |

## Returns

op status

Definition at line 3016 of file Hsm\_Hal.c.

## 4.71.5.20 HSM\_Hal\_Hash()

```
Hal_StatusType HSM_Hal_Hash (
 const HSM_HashAlgoType Algo,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

generate message digest(hash value) function.

## Note

Function ID: DES\_HSM\_API\_011

Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | hash Algo config.        |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

Returns

op status

Definition at line 2458 of file Hsm\_Hal.c.

4.71.5.21 HSM\_Hal\_HashMac()

```
Hal_StatusType HSM_Hal_HashMac (
 const HSM_HMacCfgType * CfgPtr,
 const HSM_InOutMacType * InOutPtr,
 HSM_ProcessMode ProMode)
```

generate message hash mac function use hash algo

Note

Function ID: DES\_HSM\_API\_012

Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | hash Algo config.        |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

Returns

op status

Definition at line 2505 of file Hsm\_Hal.c.

4.71.5.22 HSM\_Hal\_HostImageSecureUpgrade()

```
Hal_StatusType HSM_Hal_HostImageSecureUpgrade (
 const HSM_SecureUpgradeType * HostUpgradePara)
```

Host image upgrade.

Note

Function ID: DES\_HSM\_API\_031



Parameters

|    |                        |                                         |
|----|------------------------|-----------------------------------------|
| in | <i>HostUpgradePara</i> | Pointer to host image upgrade parameter |
|----|------------------------|-----------------------------------------|

Returns

operate status

Parameters

|    |                        |                                         |
|----|------------------------|-----------------------------------------|
| in | <i>HostUpgradePara</i> | Pointer to host image upgrade parameter |
|----|------------------------|-----------------------------------------|

Returns

operate status

Definition at line 3334 of file Hsm\_Hal.c.

4.71.5.23 HSM\_Hal\_HostImageSecureVerify()

```
Hal_StatusType HSM_Hal_HostImageSecureVerify (
 const HSM_ImageVerifyType * HostVerifyPara)
```

Host image uggrade verify.

Note

Function ID: DES\_HSM\_API\_032

Parameters

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

Returns

operate status

Parameters

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

Returns

operate status

Definition at line 3356 of file Hsm\_Hal.c.

#### 4.71.5.24 HSM\_Hal\_Init()

```
Hal_StatusType HSM_Hal_Init (
 boolean HsmIrqEn,
 uint8 HsmIrqPri)
```

hsm module init function.

##### Note

Function ID: DES\_HSM\_API\_001

##### Parameters

|    |                  |                                                    |
|----|------------------|----------------------------------------------------|
| in | <i>HsmIrqEn</i>  | hsm irq enable/disable 1:enable 0:disable(default) |
| in | <i>HsmIrqPri</i> | hsm irq priority.                                  |

##### Returns

op status

Definition at line 2258 of file Hsm\_Hal.c.

#### 4.71.5.25 HSM\_Hal\_InstallCallback()

```
void HSM_Hal_InstallCallback (
 const Hal_CallbackType Callback,
 void * Args)
```

irq callback.

##### Note

Function ID: DES\_HSM\_API\_025

##### Parameters

|    |                 |                   |
|----|-----------------|-------------------|
| in | <i>Callback</i> | callback function |
| in | <i>Args</i>     | args              |

##### Returns

op status

Definition at line 3320 of file Hsm\_Hal.c.

4.71.5.26 HSM\_Hal\_Lock()

```
void HSM_Hal_Lock (
 void)
```

Lock access hsm path.

Note

Function ID: DES\_HSM\_API\_043

Returns

op status

Definition at line 2236 of file Hsm\_Hal.c.

4.71.5.27 HSM\_Hal\_OtpRead()

```
uint32 HSM_Hal_OtpRead (
 uint32 OtpAddr,
 uint32 BytesLen,
 uint8 * Data)
```

read otp

Note

Function ID: DES\_HSM\_API\_030

Parameters

|     |                 |                   |
|-----|-----------------|-------------------|
| in  | <i>OtpAddr</i>  | otp address value |
| in  | <i>BytesLen</i> | read bytes len    |
| out | <i>Data</i>     | Secureboot Config |

Returns

op status

Definition at line 3290 of file Hsm\_Hal.c.

4.71.5.28 HSM\_Hal\_OtpWrite()

```
uint32 HSM_Hal_OtpWrite (
 uint32 OtpAddr,
 uint32 BytesLen,
 uint8 * Data)
```

write otp

Note

Function ID: DES\_HSM\_API\_042

Parameters

|     |                 |                   |
|-----|-----------------|-------------------|
| in  | <i>OtpAddr</i>  | otp address value |
| in  | <i>BytesLen</i> | read bytes len    |
| out | <i>Data</i>     | Secureboot Config |

Returns

op status

Definition at line 3297 of file Hsm\_Hal.c.

4.71.5.29 HSM\_Hal\_RemoveKey()

```
Hal_StatusType HSM_Hal_RemoveKey (
 HSM_KeyId KeyId)
```

Remove Key.

Note

Function ID: DES\_HSM\_API\_020

Parameters

|    |                    |           |
|----|--------------------|-----------|
| in | <i>Key↔<br/>Id</i> | Key Index |
|----|--------------------|-----------|

Returns

op status

Definition at line 2745 of file Hsm\_Hal.c.

4.71.5.30 HSM\_Hal\_RsaCipher()

```
Hal_StatusType HSM_Hal_RsaCipher (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Rsa Decrypt/Encrypt function.

Note

Function ID:DES\_HSM\_API\_039

## Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | rsa Algo config.         |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

## Returns

op status

Definition at line 2671 of file Hsm\_Hal.c.

#### 4.71.5.31 HSM\_Hal\_RsaSign()

```
Hal_StatusType HSM_Hal_RsaSign (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutSignType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Rsa Sign/Verify function.

## Note

Function ID: DES\_HSM\_API\_008

## Parameters

|    |                 |                              |
|----|-----------------|------------------------------|
| in | <i>CfgPtr</i>   | rsa config.                  |
| in | <i>InOutPtr</i> | rsa Input and output config. |
| in | <i>ProMode</i>  | process mode.                |

## Returns

op status

Definition at line 2639 of file Hsm\_Hal.c.

#### 4.71.5.32 HSM\_Hal\_SetImageSecureUpgradeAlgo()

```
Hal_StatusType HSM_Hal_SetImageSecureUpgradeAlgo (
 HSM_OtpCtrlAlgoType Algo,
 HSM_ImageType Type)
```

Otp Hw Ctrl, set image secure upgrade algo type.

## Note

Function ID: DES\_HSM\_API\_037

**Parameters**

|    |             |                          |
|----|-------------|--------------------------|
| in | <i>Algo</i> | secure upgrade algo type |
| in | <i>Type</i> | image type.host or hsm.  |

**Returns**

operate status

Otp Hw Ctrl, set image secure upgrade algo type.

**Parameters**

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

**Returns**

operate status

Definition at line 3438 of file Hsm\_Hal.c.

**4.71.5.33 HSM\_Hal\_SetImageSecureVerifyAlgo()**

```
Hal_StatusType HSM_Hal_SetImageSecureVerifyAlgo (
 HSM_OtpCtrlAlgoType Algo,
 HSM_ImageType Type)
```

Otp Hw Ctrl,set image secure verify algo type.

**Note**

Function ID: DES\_HSM\_API\_036

**Parameters**

|    |             |                          |
|----|-------------|--------------------------|
| in | <i>Algo</i> | secure upgrade algo type |
| in | <i>Type</i> | image type.host or hsm.  |

**Returns**

operate status

Otp Hw Ctrl,set image secure verify algo type.

**Parameters**

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

Returns

operate status

Definition at line 3489 of file Hsm\_Hal.c.

4.71.5.34 HSM\_Hal\_SetLifeCycle()

```
uint32 HSM_Hal_SetLifeCycle (
 HSM_LifeCycleType LcIndex)
```

Set LifeCycle.

Note

Function ID: DES\_HSM\_API\_041

Parameters

|    |                |                |
|----|----------------|----------------|
| in | <i>LcIndex</i> | lifecycle type |
|----|----------------|----------------|

Returns

op status

Definition at line 3252 of file Hsm\_Hal.c.

4.71.5.35 HSM\_Hal\_SetOtpExternalKey()

```
uint32 HSM_Hal_SetOtpExternalKey (
 HSM_OtpKeyId KeyId,
 const uint8 * ExternKey,
 uint32 ExternKeyLen,
 HSM_OtpKeyLevel KeyLevel,
 uint32 KeyAttr)
```

set external key or hash value(raw dta key) to otp key slot

Note

Function ID: DES\_HSM\_API\_034

Parameters

|    |                     |                                    |
|----|---------------------|------------------------------------|
| in | <i>KeyId</i>        | otp key id.                        |
| in | <i>ExternKey</i>    | the key data or hash value.        |
| in | <i>ExternKeyLen</i> | the key data or hash value length. |
| in | <i>KeyLevel</i>     | otp key level.                     |
| in | <i>KeyAttr</i>      | otp key attribute.                 |

**Returns**

operate status

Definition at line 3602 of file Hsm\_Hal.c.

**4.71.5.36 HSM\_Hal\_SetOtpKeyCipherAlgo()**

```
Hal_StatusType HSM_Hal_SetOtpKeyCipherAlgo (
 HSM_OtpKeyCipherAlgo Algo)
```

Otp Hw Ctrl, Set encrypt otp key algo.

**Note**

Function ID: DES\_HSM\_API\_033

**Parameters**

|    |             |                    |
|----|-------------|--------------------|
| in | <i>Algo</i> | encrypt algo type. |
|----|-------------|--------------------|

**Returns**

operate status

Otp Hw Ctrl, Set encrypt otp key algo.

**Parameters**

|    |                       |                                        |
|----|-----------------------|----------------------------------------|
| in | <i>HostVerifyPara</i> | Pointer to host image verify parameter |
|----|-----------------------|----------------------------------------|

**Returns**

operate status

Definition at line 3400 of file Hsm\_Hal.c.

**4.71.5.37 HSM\_Hal\_SetOtpRndKey()**

```
uint32 HSM_Hal_SetOtpRndKey (
 HSM_OtpKeyId KeyId,
 HSM_OtpKeyLevel KeyLevel,
 HSM_RndOtpKeyType KeyType,
 uint32 KeyAttr)
```

set random key to otp key slot

**Note**

Function ID: DES\_HSM\_API\_035



Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>KeyId</i>    | otp key id.              |
| in | <i>KeyLevel</i> | otp key level.           |
| in | <i>KeyType</i>  | support random key type. |
| in | <i>KeyAttr</i>  | otp key attribute.       |

Returns

operate status

Definition at line 3574 of file Hsm\_Hal.c.

4.71.5.38 HSM\_Hal\_SetPlainKey()

```
Hal_StatusType HSM_Hal_SetPlainKey (
 const HSM_PlainKeyCfgType * CfgPtr)
```

set plain key to hsm mem.

Note

Function ID: DES\_HSM\_API\_013

Parameters

|    |               |                   |
|----|---------------|-------------------|
| in | <i>CfgPtr</i> | plain key config. |
|----|---------------|-------------------|

Returns

op status

Definition at line 2754 of file Hsm\_Hal.c.

4.71.5.39 HSM\_Hal\_SetSecretKey()

```
Hal_StatusType HSM_Hal_SetSecretKey (
 HSM_KeyId TpKeyId,
 HSM_KeyId AuthKeyId,
 const HSM_SecretKeyCfgType * ImportKeyPtr,
 HSM_KeyId KeyId)
```

set secret key to hsm

Note

Function ID: DES\_HSM\_API\_016

Parameters

|    |                     |                    |
|----|---------------------|--------------------|
| in | <i>TpKeyId</i>      | transport key id.  |
| in | <i>AuthKeyId</i>    | auth key id.       |
| in | <i>ImportKeyPtr</i> | Target key config. |
| in | <i>KeyId</i>        | Target key id.     |

Returns

op status

Definition at line 3061 of file Hsm\_Hal.c.

4.71.5.40 HSM\_Hal\_SetSecureBootCfg()

```
Hal_StatusType HSM_Hal_SetSecureBootCfg (
 HSM_BootCfgType * CfgPtr)
```

set secure boot config

Note

Function ID: DES\_HSM\_API\_022

Parameters

|    |               |                   |
|----|---------------|-------------------|
| in | <i>CfgPtr</i> | Secureboot Config |
|----|---------------|-------------------|

Returns

op status

4.71.5.41 HSM\_Hal\_Sm2Cipher()

```
Hal_StatusType HSM_Hal_Sm2Cipher (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Sm2 Decrypt/Encrypt function.

Note

Function ID: DES\_HSM\_API\_006

## Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | sm2 Algo config.         |
| in | <i>InOutPtr</i> | Input and output config. |
| in | <i>ProMode</i>  | process mode.            |

## Returns

op status

## Note

Function ID:

## Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | sm2 Algo config.         |
| in | <i>InOutPtr</i> | Input and output config. |

## Returns

op status

Definition at line 2703 of file Hsm\_Hal.c.

## 4.71.5.42 HSM\_Hal\_Sm2Sign()

```
Hal_StatusType HSM_Hal_Sm2Sign (
 const HSM_AsymCfgType * CfgPtr,
 const HSM_InOutSignType * InOutPtr,
 HSM_ProcessMode ProMode)
```

SM2 Sign/Verify function.

## Note

Function ID: DES\_HSM\_API\_009

## Parameters

|    |                 |                          |
|----|-----------------|--------------------------|
| in | <i>CfgPtr</i>   | Sm2 Algo config.         |
| in | <i>InOutPtr</i> | Input and output config. |

## Returns

op status

Definition at line 2656 of file Hsm\_Hal.c.

## 4.71.5.43 HSM\_Hal\_Sm4Cipher()

```
Hal_StatusType HSM_Hal_Sm4Cipher (
 const HSM_SymCfgType * CfgPtr,
 const HSM_InOutType * InOutPtr,
 HSM_ProcessMode ProMode)
```

Sm4 Decrypt/Encrypt function.

**Note**

Function ID: DES\_HSM\_API\_004

**Parameters**

|    |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>CfgPtr</i>   | sm4 Algo config KeyId: Key Index. Iv: init vector. IvLen: iv Len Padding:data padding type. HSM_NOPADDING = 0U, HSM_PSASSA_PSS = 1U, HSM_PKCS7 = 2U, HSM_ONEWITHZEROS = 3U CipherDir :request opeartion, eg.decrypt/encrypt. HSM_ENCRYPTION = 0U, HSM_DECRYPTION = 1U, CMode:for sym algo and CMAC algo, eg.ECB/CBC. ECB_MODE = 1U CBC_MODE = 3U CFB_MODE = 4U OFB_MODE = 5U CTR_MODE = 6U SymAlgo: symmetric algo type, eg.AES_128/AES_256 HSM_SM4 = 8U, |
| in | <i>InOutPtr</i> | Input and output config. InBuf: input buffer InLen: input buffer length OutBuf: output buffer OutLen: output buffer length Vry:ignore.                                                                                                                                                                                                                                                                                                                    |
| in | <i>ProMode</i>  | process mode. START_MODE = (0x01U), UPDATE_MODE = (0x02U), FINISH_MODE = (0x04U), FW_ONEPASS_MODE = (0x07U),                                                                                                                                                                                                                                                                                                                                              |

**Returns**

Hal\_StatusType: STATUS\_ERROR: error happend in Cipher operation; STATUS\_SUCCESS: successfull

Definition at line 2350 of file Hsm\_Hal.c.

## 4.71.5.44 HSM\_Hal\_Unlock()

```
void HSM_Hal_Unlock (
 void)
```

unlock access hsm path.

**Note**

Function ID: DES\_HSM\_API\_044

**Returns**

op status

Definition at line 2241 of file Hsm\_Hal.c.

## 4.72 test\_int\_config.h File Reference

```
#include "eHSM_IntCfg_Ip.h"
```

### Macros

- #define [CONFIG\\_EHSM\\_UNIT\\_TEST\\_EVITA\\_SYM](#)

### 4.72.1 Macro Definition Documentation

#### 4.72.1.1 CONFIG\_EHSM\_UNIT\_TEST\_EVITA\_SYM

```
#define CONFIG_EHSM_UNIT_TEST_EVITA_SYM
```

Definition at line 52 of file test\_int\_config.h.

## 4.73 utest\_vecs\_st.h File Reference

```
#include <stdint.h>
```

### Classes

- struct [hash\\_testvec](#)
- struct [kdf\\_testvec](#)
- struct [cipher\\_testvec](#)
- struct [aead\\_testvec](#)
- struct [akcipher\\_testvec](#)
- struct [kpp\\_testvec](#)
  - Test struct for shared secret.*
- struct [rsacipher\\_testvec](#)
- struct [sm9cipher\\_testvec](#)
- struct [ecies\\_testvec](#)

### Macros

- #define [MAX\\_TAP](#) 4

### 4.73.1 Macro Definition Documentation

#### 4.73.1.1 MAX\_TAP

```
#define MAX_TAP 4
```

Definition at line 14 of file utest\_vecs\_st.h.

# Index

- [\\_\\_attribute\\_\\_](#)
  - [eHSM\\_Mailbox\\_Ip.c, 845](#)
- [\\_srv\\_crypto\\_cipher\\_reqhdl](#)
  - [eHSM\\_Srv\\_Ciper\\_Ip.c, 884](#)
  - [eHSM\\_Srv\\_Cipher\\_Ip.h, 885](#)
- [\\_srv\\_crypto\\_cipher\\_rsphdl](#)
  - [eHSM\\_Srv\\_Ciper\\_Ip.c, 884](#)
  - [eHSM\\_Srv\\_Cipher\\_Ip.h, 885](#)
- [\\_translate\\_bool\\_array\\_to\\_use\\_flags](#)
  - [eHSM\\_If\\_Evita\\_Key\\_Ip.c, 665](#)
- [AC784xx\\_API\\_Reference\\_Manual\\_HSM.pdf, 385](#)
- [AC784xx\\_Hsm\\_Reg.h, 385](#)
  - [HSM\\_CMD\\_BOOTROM\\_SET\\_BAUDRATE, 386](#)
  - [HSM\\_CMD\\_ENCRYPT\\_KEY, 386](#)
  - [HSM\\_CMD\\_GET\\_RANDOM\\_KEY, 386](#)
  - [OTP\\_BASE\\_ADDR, 386](#)
  - [OTP\\_ERR\\_RSP\\_CTRL\\_ADDR, 386](#)
  - [OTP\\_FW\\_CTRL\\_FIELD\\_ADDR\\_H, 387](#)
  - [OTP\\_FW\\_CTRL\\_FIELD\\_ADDR\\_L, 387](#)
  - [OTP\\_HOST\\_CTRL\\_FIELD\\_ADDR\\_H, 387](#)
  - [OTP\\_HOST\\_CTRL\\_FIELD\\_ADDR\\_L, 387](#)
  - [OTP\\_HSM\\_ENABLE\\_ADDR, 387](#)
  - [OTP\\_HSM\\_VERSION\\_ADDR, 387](#)
  - [OTP\\_HW\\_CTRL\\_FIELD\\_ADDR, 388](#)
  - [OTP\\_KEY\\_ADDR, 388](#)
  - [OTP\\_KEY\\_ATTR\\_BYTE\\_LENGTH, 388](#)
  - [OTP\\_KEY\\_ATTR\\_ENCODE\\_EACH\\_LENGTH, 388](#)
  - [OTP\\_KEY\\_ATTR\\_LC, 388](#)
  - [OTP\\_KEY\\_CRC\\_SIZE, 389](#)
  - [OTP\\_KEY\\_CRC, 388](#)
  - [OTP\\_KEY\\_SIZE, 389](#)
  - [OTP\\_LIFE\\_CYCLE\\_ADDR, 389](#)
  - [OTP\\_SECURE\\_BOOT\\_ADDR, 389](#)
  - [OTP\\_SIZE, 389](#)
  - [OTP\\_SOC\\_VERSION\\_ADDR, 389](#)
  - [OTP\\_UID\\_ADDR, 390](#)
  - [OTP\\_VERSION\\_ENCODE\\_LENGTH, 390](#)
  - [OTP\\_VERSION\\_LENGTH, 390](#)
  - [SOC\\_CMD\\_GET\\_HSM\\_FW\\_VERSION, 390](#)
  - [SOC\\_CMD\\_IMAGE\\_UPGRADE\\_UPGRADE, 390](#)
  - [SOC\\_CMD\\_IMAGE\\_VERIFY, 390](#)
- [AES128\\_CMAC\\_WITH\\_AES128\\_CBC\\_UPGRADE](#)
  - [Hsm\\_Hal.h, 954](#)
- [AES128\\_CMAC\\_WITH\\_AES128\\_CBC\\_VERIFY](#)
  - [Hsm\\_Hal.h, 954](#)
- [AES\\_CTRDRBG](#)
  - [eHSM\\_If\\_She\\_Ip.h, 772](#)
- [ALIGN\\_BYTE](#)
  - [eHSM\\_If\\_Evita\\_Types\\_Ip.h, 688, 708](#)
- [AUTOSAR\\_LOG\\_DEBUG](#)
  - [eHSM\\_Debug\\_Ip.h, 486](#)
- [AUTOSAR\\_LOG\\_ERROR](#)
  - [eHSM\\_Debug\\_Ip.h, 486](#)
- [AUTOSAR\\_LOG\\_INFO](#)
  - [eHSM\\_Debug\\_Ip.h, 487](#)
- [AUTOSAR\\_LOG\\_WARN](#)
  - [eHSM\\_Debug\\_Ip.h, 487](#)
- [AUTOSAR\\_PURE\\_LOG\\_DEBUG](#)
  - [eHSM\\_Debug\\_Ip.h, 487](#)
- [aad\\_ptr](#)
  - [ehsm\\_cmd\\_aead\\_ptr\\_st, 70](#)
- [activeUseFlag](#)
  - [eHSM\\_If\\_Evita\\_Types\\_Ip.h, 708](#)
  - [ehsm\\_key\\_status\\_, 186](#)
- [addr](#)
  - [ehsm\\_storage\\_area\\_param\\_st, 279](#)
- [Aead\\_Finish](#)
  - [eHSM\\_If\\_Evita\\_Ip.h, 635](#)
  - [eHSM\\_If\\_Evita\\_SymCper\\_Ip.c, 677](#)
- [Aead\\_Init](#)
  - [eHSM\\_If\\_Evita\\_Ip.h, 636](#)
  - [eHSM\\_If\\_Evita\\_SymCper\\_Ip.c, 677](#)
- [Aead\\_Process](#)
  - [eHSM\\_If\\_Evita\\_Ip.h, 637](#)
  - [eHSM\\_If\\_Evita\\_SymCper\\_Ip.c, 678](#)
- [aead\\_data](#)
  - [cipher\\_session\\_st, 15](#)
- [aead\\_testvec, 8](#)
  - [alen, 8](#)
  - [assoc, 8](#)
  - [clen, 9](#)
  - [crypt\\_error, 9](#)
  - [ctxt, 9](#)
  - [iv, 9](#)
  - [ivlen, 9](#)
  - [key, 9](#)
  - [klen, 10](#)
  - [novrfy, 10](#)
  - [plen, 10](#)
  - [ptext, 10](#)
  - [setauthsize\\_error, 10](#)
  - [setkey\\_error, 10](#)
  - [wk, 11](#)
- [akcipher\\_testvec, 11](#)
  - [c, 12](#)
  - [c\\_size, 12](#)
  - [hash\\_alg, 12](#)
  - [key, 12](#)
  - [key\\_len, 12](#)
  - [m, 12](#)
  - [m\\_size, 12](#)
  - [param\\_len, 13](#)
  - [params, 13](#)

- public\_key\_vec, 13
- siggen\_sigver\_test, 13
- alen
  - aead\_testvec, 8
- Alg
  - HSM\_DebugAuthConfigType, 292
- alg
  - ehsm\_debug\_auth\_st, 107
- alg\_or\_crt
  - ehsm\_gen\_key\_cmd, 142
- algo\_id
  - eHSM\_If\_Evita\_Types\_Ip.h, 708
  - ehsm\_export\_pub\_key\_, 132
  - ehsm\_gen\_key\_param\_st, 145
  - ehsm\_key\_attr\_data\_, 175
  - ehsm\_key\_status\_, 186
  - ehsm\_module\_status\_cmd, 208
  - ehsm\_module\_status\_st, 210
  - ehsm\_pub\_key\_, 218
  - ehsm\_se\_key\_, 226
- algorithm
  - cipher\_session\_st, 15
  - Crypto\_PrimitiveInfoType, 50
  - ehsm\_cmd\_hdr\_pke\_st, 76
  - ehsm\_cmd\_hdr\_rng\_st, 78
  - ehsm\_cmd\_hdr\_ske\_st, 80
  - ehsm\_create\_dh\_key\_cmd, 90
  - ehsm\_crypto\_randomgenerate\_param, 104
  - ehsm\_fast\_cmac\_st, 134
  - ehsm\_rng\_generate\_cmd, 220
- algrithm
  - ehsm\_debug\_authentication\_cmd, 109
- api\_type
  - ehsm\_cmd\_req, 84
  - ehsm\_mailbox\_req, 196
  - ehsm\_mbox\_cancel\_channel\_req, 204
  - ehsm\_mbox\_cancel\_channel\_rps, 205
  - ehsm\_service\_info, 234
- Asr\_Standard\_Types.h, 391
  - CRYPTO\_E\_BUSY, 394
  - CRYPTO\_E\_ENTROPY\_EXHAUSTION, 394
  - CRYPTO\_E\_JOB\_CANCELED, 394
  - CRYPTO\_E\_KEY\_EMPTY, 394
  - CRYPTO\_E\_KEY\_NOT\_AVAILABLE, 395
  - CRYPTO\_E\_KEY\_NOT\_VALID, 395
  - CRYPTO\_E\_KEY\_READ\_FAIL, 395
  - CRYPTO\_E\_KEY\_SIZE\_MISMATCH, 395
  - CRYPTO\_E\_KEY\_WRITE\_FAIL, 395
  - CRYPTO\_E\_QUEUE\_FULL, 396
  - CRYPTO\_E\_SMALL\_BUFFER, 396
  - CRYPTO\_KE\_CERTIFICATE\_CURRENT\_TIME, 396
  - CRYPTO\_KE\_CERTIFICATE\_DATA, 396
  - CRYPTO\_KE\_CERTIFICATE\_EXTENSIONS, 396
  - CRYPTO\_KE\_CERTIFICATE\_ISSUER, 396
  - CRYPTO\_KE\_CERTIFICATE\_PARSING\_FORMAT, 397
  - CRYPTO\_KE\_CERTIFICATE\_SERIALNUMBER, 397
  - CRYPTO\_KE\_CERTIFICATE\_SIGNATURE\_ALGORITHM, 397
  - CRYPTO\_KE\_CERTIFICATE\_SIGNATURE, 397
- CRYPTO\_KE\_CERTIFICATE\_SUBJECT\_PUBLIC\_KEY, 397
- CRYPTO\_KE\_CERTIFICATE\_SUBJECT, 397
- CRYPTO\_KE\_CERTIFICATE\_VALIDITY\_NOT\_AFTER, 398
- CRYPTO\_KE\_CERTIFICATE\_VALIDITY\_NOT\_BEFORE, 398
- CRYPTO\_KE\_CERTIFICATE\_VERSION, 398
- CRYPTO\_KE\_CIPHER\_2NDKEY, 398
- CRYPTO\_KE\_CIPHER\_IV, 398
- CRYPTO\_KE\_CIPHER\_KEY, 398
- CRYPTO\_KE\_CIPHER\_PROOF, 399
- CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM, 399
- CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS, 399
- CRYPTO\_KE\_KEYDERIVATION\_PASSWD, 399
- CRYPTO\_KE\_KEYDERIVATION\_SALT, 399
- CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM, 399
- CRYPTO\_KE\_KEYEXCHANGE\_BASE, 400
- CRYPTO\_KE\_KEYEXCHANGE\_OWNPUBLICKEY, 400
- CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY, 400
- CRYPTO\_KE\_KEYGENERATE\_ALGORITHM, 400
- CRYPTO\_KE\_KEYGENERATE\_KEY, 400
- CRYPTO\_KE\_KEYGENERATE\_SEED, 400
- CRYPTO\_KE\_MAC\_KEY, 401
- CRYPTO\_KE\_MAC\_PROOF, 401
- CRYPTO\_KE\_RANDOM\_ALGORITHM, 401
- CRYPTO\_KE\_RANDOM\_SEED\_STATE, 401
- CRYPTO\_KE\_SIGNATURE\_KEY, 401
- CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE, 401
- Crypto\_AlgorithmFamilyType, 402
- Crypto\_AlgorithmModeType, 403
- Crypto\_ConfigType, 402
- Crypto\_InputOutputRedirectionConfigType, 404
- Crypto\_JobStateType, 404
- Crypto\_KeyElementReadAccessType, 405
- Crypto\_KeyElementWriteAccessType, 405
- Crypto\_OperationModeType, 405
- Crypto\_ProcessingType, 406
- Crypto\_ServiceInfoType, 406
- Crypto\_VerifyResultType, 406
- CryptoDriverStateType, 408
- CryptoKeyFormatType, 408
- E\_NOT\_OK, 402
- E\_OK, 402
- Std\_HsmReturnTypes, 402
- assoc
  - aead\_testvec, 8
- AsymAlgo
  - HSM\_AsymCfgType, 285
- attr
  - eHSM\_If\_Evita\_Types\_Ip.h, 708
  - ehsm\_internal\_key\_, 173
- auth\_flag
  - ehsm\_key\_flags\_element\_st, 181
  - HSM\_KeyFlagsElementType, 314
- auth\_sign\_data
  - eHSM\_If\_Evita\_Types\_Ip.h, 708
  - ehsm\_external\_key\_, 133
- auth\_size
  - crypto\_copy\_key\_info, 25

- crypto\_evita\_key\_info, 28
- eHSM\_If\_Evita\_Types\_lp.h, 709
- ehsm\_key\_flags\_element\_st, 181
- ehsm\_se\_key\_, 227
- HSM\_KeyFlagsElementType, 314
- auth\_value
  - crypto\_copy\_key\_info, 25
  - crypto\_evita\_key\_info, 28
  - eHSM\_If\_Evita\_Types\_lp.h, 709
  - ehsm\_key\_flags\_element\_st, 181
  - ehsm\_se\_key\_, 227
  - HSM\_KeyFlagsElementType, 314
- auth\_value\_exist\_flags
  - ehsm\_key\_flags\_element\_st, 182
  - HSM\_KeyFlagsElementType, 314
- AuthSize
  - HSM\_KeyHandleInfoType, 315
- AuthValueSize
  - HSM\_SecretKeyCfgType, 333
- AuthValue
  - HSM\_KeyHandleInfoType, 315
  - HSM\_PlainKeyCfgType, 322
  - HSM\_SecretKeyCfgType, 333
- AuthValueSize
  - HSM\_PlainKeyCfgType, 322
- authenticated\_key
  - ehsm\_sm9\_export\_key\_cmd, 253
  - ehsm\_sm9\_import\_key\_cmd, 259
- authenticated\_key\_size
  - ehsm\_sm9\_export\_key\_cmd, 253
  - ehsm\_sm9\_import\_key\_cmd, 259
- authenticity\_key\_auth\_size
  - ehsm\_export\_key\_cmd, 129
  - ehsm\_import\_key\_cmd, 171
- authenticity\_key\_auth\_value
  - ehsm\_export\_key\_cmd, 129
  - ehsm\_import\_key\_cmd, 171
- authenticity\_key\_author\_size
  - ehsm\_evita\_key\_export, 120
  - ehsm\_evita\_key\_import\_st, 122
- authenticity\_key\_author\_value
  - ehsm\_evita\_key\_export, 120
  - ehsm\_evita\_key\_import\_st, 122
- authenticity\_key\_authorization
  - crypto\_import\_evita\_key\_info, 30
  - crypto\_key\_export\_info, 45
- authenticity\_key\_authorization\_size
  - crypto\_import\_evita\_key\_info, 31
  - crypto\_key\_export\_info, 45
- authenticity\_key\_handle
  - crypto\_import\_evita\_key\_info, 31
  - crypto\_key\_export\_info, 46
  - ehsm\_evita\_key\_export, 120
  - ehsm\_evita\_key\_import\_st, 123
  - ehsm\_export\_key\_cmd, 130
  - ehsm\_import\_key\_cmd, 171
- b\_public
  - kpp\_testvec, 348
- b\_public\_size
  - kpp\_testvec, 348
- b\_secret
  - kpp\_testvec, 349
- b\_secret\_size
  - kpp\_testvec, 349
- BasicInOut
  - HSM\_InOutMacType, 308
  - HSM\_InOutSignType, 309
- baud\_div
  - ehsm\_set\_baudrate\_cmd, 236
- bit\_bytes
  - bitmap, 14
- bit\_map
  - bitmap, 14
- bit\_max
  - bitmap, 14
- bitmap, 13
  - bit\_bytes, 14
  - bit\_map, 14
  - bit\_max, 14
  - ehsm\_cmd\_req\_buffer, 87
- bitmap\_st
  - eHSM\_Compt\_Bitmap.h, 435
- block\_buf
  - ehsm\_ctx\_block\_mgr, 105
- block\_sz
  - ehsm\_ctx\_block\_mgr, 105
  - ehsm\_ctx\_session\_st, 106
- BootAddr
  - HSM\_BootCfgType, 288
- buf
  - ehsm\_get\_challenge\_st, 153
- c
  - akcipher\_testvec, 12
  - rsacipher\_testvec, 356
  - sm9cipher\_testvec, 364
- c\_size
  - akcipher\_testvec, 12
  - rsacipher\_testvec, 357
- c\_sz
  - sm9cipher\_testvec, 364
- CMD\_TAG\_BYTE\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, 852
- CMD\_TAG\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_lp.h, 852
- CMD\_TAG\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, 852
- CMode
  - HSM\_SymCfgType, 340
- CODE\_VALID\_FLAG
  - eHSM\_Com\_Struct\_lp.h, 412
- COMMAND\_REQ\_QUANTITY
  - eHSM\_Srv\_Mgr\_lp.c, 905
- COMMON\_LOG\_DEBUG
  - eHSM\_Debug\_lp.h, 487
- COMMON\_LOG\_ERROR
  - eHSM\_Debug\_lp.h, 487
- COMMON\_LOG\_INFO
  - eHSM\_Debug\_lp.h, 488
- COMMON\_LOG\_WARN
  - eHSM\_Debug\_lp.h, 488



COMMON\_PURE\_LOG\_DEBUG  
    eHSM\_Debug\_lp.h, [488](#)

CONFIG\_EHSM\_ARCH\_HOST\_MAILBOX\_POLLING  
    eHSM\_Config\_lp.h, [446](#)

CONFIG\_EHSM\_ARCH\_MAIN\_HOOK  
    eHSM\_IntCfg\_lp.h, [795](#)

CONFIG\_EHSM\_ARCH\_MULTI\_CHANNEL  
    eHSM\_IntCfg\_lp.h, [795](#)

CONFIG\_EHSM\_ARCH\_OS\_NONE  
    eHSM\_IntCfg\_lp.h, [796](#)

CONFIG\_EHSM\_ARCH\_SHARE\_MEM  
    eHSM\_IntCfg\_lp.h, [796](#)

CONFIG\_EHSM\_ARCH\_V\_CMD\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [446](#)

CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_HASH\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [447](#)

CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_K\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [447](#)

CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_PKE\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [447](#)

CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_SKE\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [447](#)

CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_SYSMGR\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [447](#)

CONFIG\_EHSM\_ARCH\_V\_CRYPT\_OBJ\_TRNG\_QUEUE\_SIZE  
    eHSM\_Config\_lp.h, [447](#)

CONFIG\_EHSM\_ARCH\_V\_DEFAULT\_CMD\_TIMEOUT  
    eHSM\_Config\_lp.h, [448](#)

CONFIG\_EHSM\_ARCH\_V\_JTAG\_TIMEOUT  
    eHSM\_Config\_lp.h, [448](#)

CONFIG\_EHSM\_ARCH\_V\_MAILBOX\_TIMEOUT  
    eHSM\_Config\_lp.h, [448](#)

CONFIG\_EHSM\_ARCH\_V\_REQ\_HASH\_MAX\_SIZE  
    eHSM\_IntCfg\_lp.h, [796](#)

CONFIG\_EHSM\_ARCH\_V\_REQ\_SKE\_MAX\_SIZE  
    eHSM\_IntCfg\_lp.h, [796](#)

CONFIG\_EHSM\_ARCH\_V\_RSA\_K\_CMD\_TIMEOUT  
    eHSM\_Config\_lp.h, [448](#)

CONFIG\_EHSM\_AUTOSAR  
    eHSM\_Config\_lp.h, [448](#)

CONFIG\_EHSM\_COUNTER\_AUTO\_INCREASE  
    eHSM\_IntCfg\_lp.h, [796](#)

CONFIG\_EHSM\_CRYPT\_AEAD  
    eHSM\_IntCfg\_lp.h, [797](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_AES  
    eHSM\_IntCfg\_lp.h, [797](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_CTRDRBG  
    eHSM\_IntCfg\_lp.h, [797](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_DH  
    eHSM\_IntCfg\_lp.h, [797](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_ECC  
    eHSM\_IntCfg\_lp.h, [797](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_ED25519  
    eHSM\_IntCfg\_lp.h, [797](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_MD5  
    eHSM\_IntCfg\_lp.h, [798](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_PBKDF2  
    eHSM\_IntCfg\_lp.h, [798](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_RSA\_1024  
    eHSM\_IntCfg\_lp.h, [798](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_RSA\_1024\_CERT  
    eHSM\_IntCfg\_lp.h, [798](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_RSA\_2048  
    eHSM\_IntCfg\_lp.h, [798](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_RSA\_2048\_CERT  
    eHSM\_IntCfg\_lp.h, [799](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_RSA  
    eHSM\_IntCfg\_lp.h, [798](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SECP256R1  
    eHSM\_IntCfg\_lp.h, [799](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SECP384R1  
    eHSM\_IntCfg\_lp.h, [799](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA1  
    eHSM\_IntCfg\_lp.h, [799](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA2  
    eHSM\_IntCfg\_lp.h, [799](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA224  
    eHSM\_IntCfg\_lp.h, [799](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA256  
    eHSM\_IntCfg\_lp.h, [800](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA384  
    eHSM\_IntCfg\_lp.h, [800](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA512  
    eHSM\_IntCfg\_lp.h, [800](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA512\_224  
    eHSM\_IntCfg\_lp.h, [800](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SHA512\_256  
    eHSM\_IntCfg\_lp.h, [800](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SM2  
    eHSM\_IntCfg\_lp.h, [800](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SM3  
    eHSM\_IntCfg\_lp.h, [801](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_SM4  
    eHSM\_IntCfg\_lp.h, [801](#)

CONFIG\_EHSM\_CRYPT\_ALGOFAM\_X963  
    eHSM\_IntCfg\_lp.h, [801](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_CBC\_MAC  
    eHSM\_IntCfg\_lp.h, [801](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_CBC  
    eHSM\_IntCfg\_lp.h, [801](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_CFB  
    eHSM\_IntCfg\_lp.h, [801](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_CMAC  
    eHSM\_IntCfg\_lp.h, [802](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_CTR  
    eHSM\_IntCfg\_lp.h, [802](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_ECB  
    eHSM\_IntCfg\_lp.h, [802](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_GMAC  
    eHSM\_IntCfg\_lp.h, [802](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_OFB  
    eHSM\_IntCfg\_lp.h, [802](#)

CONFIG\_EHSM\_CRYPT\_ALGOMODE\_RSASSA\_PSS  
    eHSM\_IntCfg\_lp.h, [802](#)

CONFIG\_EHSM\_CRYPT\_RSA\_CERT\_MODE  
    eHSM\_IntCfg\_lp.h, [803](#)

- CONFIG\_EHSM\_CRYPTO\_V\_CERT\_MAX\_PUB\_K\_SIZE
  - eHSM\_IntCfg\_lp.h, [803](#)
- CONFIG\_EHSM\_CRYPTO\_V\_CERT\_MAX\_SIZE
  - eHSM\_IntCfg\_lp.h, [803](#)
- CONFIG\_EHSM\_CRYPTO\_V\_GCM\_MAX\_AAD\_SIZE
  - eHSM\_IntCfg\_lp.h, [803](#)
- CONFIG\_EHSM\_CRYPTO\_V\_HMAC\_MAX\_KSIZE
  - eHSM\_IntCfg\_lp.h, [803](#)
- CONFIG\_EHSM\_EVITA
  - eHSM\_Config\_lp.h, [449](#)
- CONFIG\_EHSM\_FIRMWARE\_UPGRADE
  - eHSM\_IntCfg\_lp.h, [804](#)
- CONFIG\_EHSM\_HASH\_CORE\_NUM
  - eHSM\_IntCfg\_lp.h, [804](#)
- CONFIG\_EHSM\_HW\_AHB\_BYTE
  - eHSM\_IntCfg\_lp.h, [804](#)
- CONFIG\_EHSM\_HW\_BRANCH\_2\_0\_0
  - eHSM\_IntCfg\_lp.h, [804](#)
- CONFIG\_EHSM\_HW\_COUNTER
  - eHSM\_IntCfg\_lp.h, [804](#)
- CONFIG\_EHSM\_HW\_FLASH\_ECC
  - eHSM\_IntCfg\_lp.h, [805](#)
- CONFIG\_EHSM\_HW\_FLASH
  - eHSM\_IntCfg\_lp.h, [805](#)
- CONFIG\_EHSM\_HW\_GUOMI\_LEVEL1
  - eHSM\_IntCfg\_lp.h, [805](#)
- CONFIG\_EHSM\_HW\_HASH\_DMA
  - eHSM\_IntCfg\_lp.h, [805](#)
- CONFIG\_EHSM\_HW\_HASH\_LP
  - eHSM\_IntCfg\_lp.h, [806](#)
- CONFIG\_EHSM\_HW\_HASH
  - eHSM\_IntCfg\_lp.h, [805](#)
- CONFIG\_EHSM\_HW\_INSTALL\_K\_KEK
  - eHSM\_IntCfg\_lp.h, [806](#)
- CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DEBUG\_MODE
  - eHSM\_IntCfg\_lp.h, [806](#)
- CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DESTROY\_MODE
  - eHSM\_IntCfg\_lp.h, [806](#)
- CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DEVELOP\_MODE
  - eHSM\_IntCfg\_lp.h, [806](#)
- CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_MANUFACTURE\_←  
MODE
  - eHSM\_IntCfg\_lp.h, [807](#)
- CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_TEST\_MODE
  - eHSM\_IntCfg\_lp.h, [807](#)
- CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_USER\_MODE
  - eHSM\_IntCfg\_lp.h, [807](#)
- CONFIG\_EHSM\_HW\_LOW\_POWER
  - eHSM\_IntCfg\_lp.h, [807](#)
- CONFIG\_EHSM\_HW\_OTP\_MAP
  - eHSM\_IntCfg\_lp.h, [808](#)
- CONFIG\_EHSM\_HW\_OTP
  - eHSM\_IntCfg\_lp.h, [807](#)
- CONFIG\_EHSM\_HW\_PKE\_LP
  - eHSM\_IntCfg\_lp.h, [808](#)
- CONFIG\_EHSM\_HW\_PKE
  - eHSM\_IntCfg\_lp.h, [808](#)
- CONFIG\_EHSM\_HW\_SKE\_DMA
  - eHSM\_IntCfg\_lp.h, [808](#)
- CONFIG\_EHSM\_HW\_SKE\_LP
  - eHSM\_IntCfg\_lp.h, [808](#)
- CONFIG\_EHSM\_HW\_SKE\_SECURE\_PORT
  - eHSM\_IntCfg\_lp.h, [809](#)
- CONFIG\_EHSM\_HW\_TRNG
  - eHSM\_IntCfg\_lp.h, [809](#)
- CONFIG\_EHSM\_HW\_UTC\_TIME
  - eHSM\_IntCfg\_lp.h, [809](#)
- CONFIG\_EHSM\_HW\_V\_CODE\_MAX\_SIZE
  - eHSM\_IntCfg\_lp.h, [809](#)
- CONFIG\_EHSM\_HW\_V\_CUSTOMER\_OTP
  - eHSM\_IntCfg\_lp.h, [809](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_BASE\_ADDR
  - eHSM\_IntCfg\_lp.h, [810](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_CUSTOMER
  - eHSM\_IntCfg\_lp.h, [810](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_DATA\_ADDR
  - eHSM\_IntCfg\_lp.h, [810](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_DATA\_SIZE
  - eHSM\_IntCfg\_lp.h, [810](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_ERASE\_CELL\_VALUE
  - eHSM\_IntCfg\_lp.h, [810](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_FREE\_ECC\_IDX1
  - eHSM\_IntCfg\_lp.h, [810](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_FREE\_ECC\_IDX2
  - eHSM\_IntCfg\_lp.h, [811](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_KEY\_ADDR
  - eHSM\_IntCfg\_lp.h, [811](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_KEY\_SIZE
  - eHSM\_IntCfg\_lp.h, [811](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_LOG\_ADDR
  - eHSM\_IntCfg\_lp.h, [811](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_LOG\_SIZE
  - eHSM\_IntCfg\_lp.h, [811](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_NONE
  - eHSM\_IntCfg\_lp.h, [811](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_PAGE\_SIZE
  - eHSM\_IntCfg\_lp.h, [812](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_SIMULATE
  - eHSM\_IntCfg\_lp.h, [812](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_SIZE
  - eHSM\_IntCfg\_lp.h, [812](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_SYS\_ADDR
  - eHSM\_IntCfg\_lp.h, [812](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_SYS\_SIZE
  - eHSM\_IntCfg\_lp.h, [812](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_TYPE
  - eHSM\_IntCfg\_lp.h, [812](#)
- CONFIG\_EHSM\_HW\_V\_FLASH\_WRITE\_MIN\_BYTES
  - eHSM\_IntCfg\_lp.h, [813](#)
- CONFIG\_EHSM\_HW\_V\_NONE\_OTP
  - eHSM\_IntCfg\_lp.h, [813](#)
- CONFIG\_EHSM\_HW\_V\_OTP\_BASE\_ADDR
  - eHSM\_IntCfg\_lp.h, [813](#)
- CONFIG\_EHSM\_HW\_V\_OTP\_K\_ATTR\_LENGTH
  - eHSM\_IntCfg\_lp.h, [813](#)
- CONFIG\_EHSM\_HW\_V\_OTP\_KEY\_NUM
  - eHSM\_IntCfg\_lp.h, [813](#)
- CONFIG\_EHSM\_HW\_V\_OTP\_PAGE\_SIZE
  - eHSM\_IntCfg\_lp.h, [813](#)
- CONFIG\_EHSM\_HW\_V\_OTP\_SIZE
  - eHSM\_IntCfg\_lp.h, [814](#)
- CONFIG\_EHSM\_HW\_V\_OTP\_TYPE

eHSM\_IntCfg\_Ip.h, [814](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_VERSION\_LENGTH  
eHSM\_IntCfg\_Ip.h, [814](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_WRITE\_MIN\_BYTES  
eHSM\_IntCfg\_Ip.h, [814](#)  
CONFIG\_EHSM\_HW\_V\_SIMULATE\_OTP  
eHSM\_IntCfg\_Ip.h, [814](#)  
CONFIG\_EHSM\_HW\_V\_WORK\_FREQ  
eHSM\_IntCfg\_Ip.h, [814](#)  
CONFIG\_EHSM\_JTAG\_DEBUG\_AUTH  
eHSM\_IntCfg\_Ip.h, [815](#)  
CONFIG\_EHSM\_KMGR\_CHECK\_OTP\_K\_ATTR  
eHSM\_IntCfg\_Ip.h, [815](#)  
CONFIG\_EHSM\_KMGR\_PLAIN\_K\_IMPORT  
eHSM\_IntCfg\_Ip.h, [815](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [449](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_PERSIST  
eHSM\_Config\_Ip.h, [449](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [449](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [449](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [449](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PERSIST  
eHSM\_Config\_Ip.h, [450](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [450](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [450](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [450](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PERSIST  
eHSM\_Config\_Ip.h, [450](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [450](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [451](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_EDD\_DATA\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [451](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_EDD\_DATA\_PERSIST  
eHSM\_Config\_Ip.h, [451](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_EDD\_DATA\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [451](#)

```

CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SIGN↔
 EDDATA_WRITE_ACCESS
 eHSM_Config_Ip.h, 451
CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJ↔
 ECT_PUBLIC_K_PARTIAL_ACCESS
 eHSM_Config_Ip.h, 451
CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJ↔
 ECT_PUBLIC_K_PERSIST
 eHSM_Config_Ip.h, 452
CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJ↔
 ECT_PUBLIC_K_READ_ACCESS
 eHSM_Config_Ip.h, 452
CONFIG_EHSM_KMGR_V_ASR_CERTIFICATE_SUBJ↔
 ECT_PUBLIC_K_WRITE_ACCESS
 eHSM_Config_Ip.h, 452
CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_P↔
 ARTIAL_ACCESS
 eHSM_Config_Ip.h, 452
CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_P↔
 ERSIST
 eHSM_Config_Ip.h, 452
CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_R↔
 EAD_ACCESS
 eHSM_Config_Ip.h, 452
CONFIG_EHSM_KMGR_V_ASR_CIPHER_2NDKEY_W↔
 RITE_ACCESS
 eHSM_Config_Ip.h, 453
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_AL↔
 G_PARTIAL_ACCESS
 eHSM_Config_Ip.h, 453
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_AL↔
 G_PERSIST
 eHSM_Config_Ip.h, 453
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_AL↔
 G_READ_ACCESS
 eHSM_Config_Ip.h, 453
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CIPHER_AL↔
 G_WRITE_ACCESS
 eHSM_Config_Ip.h, 453
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID↔
 PARTIAL_ACCESS
 eHSM_Config_Ip.h, 453
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID↔
 PERSIST
 eHSM_Config_Ip.h, 454
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID↔
 READ_ACCESS
 eHSM_Config_Ip.h, 454
CONFIG_EHSM_KMGR_V_ASR_CIPHER_CURVE_ID↔
 WRITE_ACCESS
 eHSM_Config_Ip.h, 454
CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PARTIAL↔
 _ACCESS
 eHSM_Config_Ip.h, 454
CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_PERSIST
 eHSM_Config_Ip.h, 454
CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_READ_A↔
 CCESS
 eHSM_Config_Ip.h, 454
CONFIG_EHSM_KMGR_V_ASR_CIPHER_IV_WRITE↔
 ACCESS

```

eHSM\_Config\_Ip.h, [455](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [455](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_PERSIST  
 eHSM\_Config\_Ip.h, [455](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [455](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [455](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [455](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_PERSIST  
 eHSM\_Config\_Ip.h, [456](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [456](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_ALG\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [456](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [456](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_PERSIST  
 eHSM\_Config\_Ip.h, [456](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [456](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_ALG\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [457](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [457](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_PERSIST  
 eHSM\_Config\_Ip.h, [457](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [457](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_SIZE\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [457](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [457](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_PERSIST  
 eHSM\_Config\_Ip.h, [458](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [458](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROOF\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [458](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [458](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_PERSIST  
 eHSM\_Config\_Ip.h, [458](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [458](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARENT\_K\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [459](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [459](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_PERSIST  
 eHSM\_Config\_Ip.h, [459](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [459](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARGET\_K\_HANDLE\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [459](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [459](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_PERSIST  
 eHSM\_Config\_Ip.h, [460](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [460](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BLOB\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [460](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [460](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PERSIST  
 eHSM\_Config\_Ip.h, [460](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [460](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [461](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [461](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PERSIST  
 eHSM\_Config\_Ip.h, [461](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_READ\_ACCESS  
 eHSM\_Config\_Ip.h, [461](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_WRITE\_ACCESS  
 eHSM\_Config\_Ip.h, [461](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_HANDLE\_PARTIAL\_ACCESS  
 eHSM\_Config\_Ip.h, [461](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHAN↔  
     DLE\_PERSIST  
     eHSM\_Config\_lp.h, [462](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHAN↔  
     DLE\_READ\_ACCESS  
     eHSM\_Config\_lp.h, [462](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHAN↔  
     DLE\_WRITE\_ACCESS  
     eHSM\_Config\_lp.h, [462](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_P↔  
     ARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [462](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_P↔  
     ERSIST  
     eHSM\_Config\_lp.h, [462](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_R↔  
     EAD\_ACCESS  
     eHSM\_Config\_lp.h, [462](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_W↔  
     RITE\_ACCESS  
     eHSM\_Config\_lp.h, [463](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_PARTI↔  
     AL\_ACCESS  
     eHSM\_Config\_lp.h, [463](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_PERSI↔  
     ST  
     eHSM\_Config\_lp.h, [463](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_READ↔  
     \_ACCESS  
     eHSM\_Config\_lp.h, [463](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_WRITE↔  
     \_ACCESS  
     eHSM\_Config\_lp.h, [463](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PARTIAL↔  
     \_ACCESS  
     eHSM\_Config\_lp.h, [463](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PERSIST  
     eHSM\_Config\_lp.h, [464](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_READ\_A↔  
     CESS  
     eHSM\_Config\_lp.h, [464](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_WRITE↔  
     ACCESS  
     eHSM\_Config\_lp.h, [464](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALG↔  
     ORITHM\_PARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [464](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALG↔  
     ORITHM\_PERSIST  
     eHSM\_Config\_lp.h, [464](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALG↔  
     ORITHM\_READ\_ACCESS  
     eHSM\_Config\_lp.h, [464](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALG↔  
     ORITHM\_WRITE\_ACCESS  
     eHSM\_Config\_lp.h, [465](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITER↔  
     ATIONS\_PARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [465](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITER↔  
     ATIONS\_PERSIST  
     eHSM\_Config\_lp.h, [465](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITER↔  
     ATIONS\_READ\_ACCESS  
     eHSM\_Config\_lp.h, [465](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITER↔  
     ATIONS\_WRITE\_ACCESS  
     eHSM\_Config\_lp.h, [465](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PA↔  
     RTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [465](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PE↔  
     RSIST  
     eHSM\_Config\_lp.h, [466](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_RE↔  
     AD\_ACCESS  
     eHSM\_Config\_lp.h, [466](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_W↔  
     RITE\_ACCESS  
     eHSM\_Config\_lp.h, [466](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHA↔  
     NDLE\_PARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [466](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHA↔  
     NDLE\_PERSIST  
     eHSM\_Config\_lp.h, [466](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHA↔  
     NDLE\_READ\_ACCESS  
     eHSM\_Config\_lp.h, [466](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHA↔  
     NDLE\_WRITE\_ACCESS  
     eHSM\_Config\_lp.h, [467](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASS↔  
     WD\_PARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [467](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASS↔  
     WD\_PERSIST  
     eHSM\_Config\_lp.h, [467](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASS↔  
     WD\_READ\_ACCESS  
     eHSM\_Config\_lp.h, [467](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASS↔  
     WD\_WRITE\_ACCESS  
     eHSM\_Config\_lp.h, [467](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT↔  
     \_PARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [467](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT↔  
     \_PERSIST  
     eHSM\_Config\_lp.h, [468](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT↔  
     \_READ\_ACCESS  
     eHSM\_Config\_lp.h, [468](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT↔  
     \_WRITE\_ACCESS  
     eHSM\_Config\_lp.h, [468](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE↔  
     \_PARTIAL\_ACCESS  
     eHSM\_Config\_lp.h, [468](#)  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE↔  
     \_PERSIST  
     eHSM\_Config\_lp.h, [468](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE↔  
\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [468](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE↔  
\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [469](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGO↔  
RITHM\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [469](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGO↔  
RITHM\_PERSIST  
eHSM\_Config\_Ip.h, [469](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGO↔  
RITHM\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [469](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGO↔  
RITHM\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [469](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE↔  
\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [469](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE↔  
\_PERSIST  
eHSM\_Config\_Ip.h, [470](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE↔  
\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [470](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_BASE↔  
\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [470](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYIN↔  
FO\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [470](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYIN↔  
FO\_PERSIST  
eHSM\_Config\_Ip.h, [470](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYIN↔  
FO\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [470](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEYIN↔  
FO\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNP↔  
UBKEY\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNP↔  
UBKEY\_PERSIST  
eHSM\_Config\_Ip.h, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNP↔  
UBKEY\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_OWNP↔  
UBKEY\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEER↔  
PUBKEY\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEER↔  
PUBKEY\_PERSIST  
eHSM\_Config\_Ip.h, [472](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEER↔  
PUBKEY\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [472](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PEER↔  
PUBKEY\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [472](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVK↔  
EY\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [472](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVK↔  
EY\_PERSIST  
eHSM\_Config\_Ip.h, [472](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVK↔  
EY\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [472](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PRIVK↔  
EY\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [473](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBK↔  
TYPE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [473](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBK↔  
TYPE\_PERSIST  
eHSM\_Config\_Ip.h, [473](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBK↔  
TYPE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [473](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_PUBK↔  
TYPE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [473](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAR↔  
EDVALUE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [473](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAR↔  
EDVALUE\_PERSIST  
eHSM\_Config\_Ip.h, [474](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAR↔  
EDVALUE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [474](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SHAR↔  
EDVALUE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [474](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_L↔  
OCALTMPKINFO\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [474](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_L↔  
OCALTMPKINFO\_PERSIST  
eHSM\_Config\_Ip.h, [474](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_L↔  
OCALTMPKINFO\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [474](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_L↔  
OCALTMPKINFO\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [475](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2↔  
PEERTMPPUBK\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [475](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2↔  
PEERTMPPUBK\_PERSIST  
eHSM\_Config\_Ip.h, [475](#)



- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
PEERTMPPUBK\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [475](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
PEERTMPPUBK\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [475](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
ROLE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [475](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
ROLE\_PERSIST  
eHSM\_Config\_Ip.h, [476](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
ROLE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [476](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
ROLE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [476](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
S1\_S2\_VALUE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [476](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
S1\_S2\_VALUE\_PERSIST  
eHSM\_Config\_Ip.h, [476](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
S1\_S2\_VALUE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [476](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
S1\_S2\_VALUE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [477](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
SA\_SB\_VALUE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [477](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
SA\_SB\_VALUE\_PERSIST  
eHSM\_Config\_Ip.h, [477](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
SA\_SB\_VALUE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [477](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_SM2\_↵  
SA\_SB\_VALUE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [477](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGO\_↵  
RITHM\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [477](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGO\_↵  
RITHM\_PERSIST  
eHSM\_Config\_Ip.h, [478](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGO\_↵  
RITHM\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [478](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_ALGO\_↵  
RITHM\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [478](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_↵  
INFO\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [478](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_↵  
INFO\_PERSIST  
eHSM\_Config\_Ip.h, [478](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_↵  
INFO\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [478](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_DH\_K\_↵  
INFO\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [479](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_PAR\_↵  
TIAL\_ACCESS  
eHSM\_Config\_Ip.h, [479](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_PER\_↵  
SIST  
eHSM\_Config\_Ip.h, [479](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_REA\_↵  
D\_ACCESS  
eHSM\_Config\_Ip.h, [479](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K\_WRI\_↵  
TE\_ACCESS  
eHSM\_Config\_Ip.h, [479](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_↵  
\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [479](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_↵  
\_PERSIST  
eHSM\_Config\_Ip.h, [480](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_↵  
\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [480](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KINFO\_↵  
\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [480](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_↵  
PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [480](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_↵  
PERSIST  
eHSM\_Config\_Ip.h, [480](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_↵  
READ\_ACCESS  
eHSM\_Config\_Ip.h, [480](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_SEED\_↵  
WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_PARTIAL\_A\_↵  
CCESS  
eHSM\_Config\_Ip.h, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_PERSIST  
eHSM\_Config\_Ip.h, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_READ\_ACC\_↵  
ESS  
eHSM\_Config\_Ip.h, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_WRITE\_ACC\_↵  
ESS  
eHSM\_Config\_Ip.h, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_PARTI\_↵  
AL\_ACCESSRTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_PERSI\_↵  
ST  
eHSM\_Config\_Ip.h, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_READ\_↵  
\_ACCESS

- eHSM\_Config\_Ip.h, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_WRIT↔  
E\_ACCESS  
eHSM\_Config\_Ip.h, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PARTIA↔  
L\_ACCESS  
eHSM\_Config\_Ip.h, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PERSIST  
eHSM\_Config\_Ip.h, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_READ↔  
ACCESS  
eHSM\_Config\_Ip.h, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_WRITE↔  
ACCESS  
eHSM\_Config\_Ip.h, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_PAR↔  
TIAL\_ACCESS  
eHSM\_Config\_Ip.h, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_PER↔  
SIST  
eHSM\_Config\_Ip.h, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_REA↔  
D\_ACCESS  
eHSM\_Config\_Ip.h, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_WRI↔  
TE\_ACCESS  
eHSM\_Config\_Ip.h, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_C↔  
RT\_MODE\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_C↔  
RT\_MODE\_PERSIST  
eHSM\_Config\_Ip.h, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_C↔  
RT\_MODE\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RSA\_C↔  
RT\_MODE\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMES↔  
TAMPED\_PARTIAL\_ACCESS  
eHSM\_Config\_Ip.h, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMES↔  
TAMPED\_PERSIST  
eHSM\_Config\_Ip.h, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMES↔  
TAMPED\_READ\_ACCESS  
eHSM\_Config\_Ip.h, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TIMES↔  
TAMPED\_WRITE\_ACCESS  
eHSM\_Config\_Ip.h, [485](#)
- CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_AREA\_SIZE  
eHSM\_IntCfg\_Ip.h, [815](#)
- CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_END\_ADDR  
eHSM\_IntCfg\_Ip.h, [815](#)
- CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [816](#)
- CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_SLOT\_SIZE  
eHSM\_IntCfg\_Ip.h, [816](#)
- CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_START\_ADDR  
eHSM\_IntCfg\_Ip.h, [816](#)
- CONFIG\_EHSM\_KMGR\_V\_ECC\_RAM\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [816](#)
- CONFIG\_EHSM\_KMGR\_V\_FLASH\_K\_START\_OFFSET  
eHSM\_IntCfg\_Ip.h, [816](#)
- CONFIG\_EHSM\_KMGR\_V\_FLASH\_OTP\_K\_START\_O↔  
FFSET  
eHSM\_IntCfg\_Ip.h, [816](#)
- CONFIG\_EHSM\_KMGR\_V\_MAX\_AUTH\_CODE\_SIZE  
eHSM\_IntCfg\_Ip.h, [817](#)
- CONFIG\_EHSM\_KMGR\_V\_OTP\_EXT\_K\_OFF  
eHSM\_IntCfg\_Ip.h, [817](#)
- CONFIG\_EHSM\_KMGR\_V\_OTP\_EXT\_K\_SIZE  
eHSM\_IntCfg\_Ip.h, [817](#)
- CONFIG\_EHSM\_KMGR\_V\_RAM\_K\_MEM\_SIZE  
eHSM\_IntCfg\_Ip.h, [817](#)
- CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_AREA\_SIZE  
eHSM\_IntCfg\_Ip.h, [817](#)
- CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_END\_ADDR  
eHSM\_IntCfg\_Ip.h, [817](#)
- CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [818](#)
- CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_SLOT\_SIZE  
eHSM\_IntCfg\_Ip.h, [818](#)
- CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_START\_ADDR  
eHSM\_IntCfg\_Ip.h, [818](#)
- CONFIG\_EHSM\_KMGR\_V\_RSA\_RAM\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [818](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_BOOT\_MAC\_K  
eHSM\_If\_She\_Ip.c, [758](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_AREA\_SIZE  
eHSM\_IntCfg\_Ip.h, [818](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_END\_ADDR  
eHSM\_IntCfg\_Ip.h, [818](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_MIN  
eHSM\_If\_She\_Ip.c, [758](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [819](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_SLOT\_SIZE  
eHSM\_IntCfg\_Ip.h, [819](#)
- CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_START\_ADDR  
eHSM\_IntCfg\_Ip.h, [819](#)
- CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_AREA\_SIZE  
eHSM\_IntCfg\_Ip.h, [819](#)
- CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_END\_ADDR  
eHSM\_IntCfg\_Ip.h, [819](#)
- CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [820](#)
- CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_SLOT\_SIZE  
eHSM\_IntCfg\_Ip.h, [820](#)
- CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_START\_ADDR  
eHSM\_IntCfg\_Ip.h, [820](#)
- CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_AREA\_SIZE  
eHSM\_IntCfg\_Ip.h, [820](#)
- CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_END\_ADDR  
eHSM\_IntCfg\_Ip.h, [820](#)
- CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_NUM  
eHSM\_IntCfg\_Ip.h, [820](#)
- CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_SLOT\_SIZE  
eHSM\_IntCfg\_Ip.h, [821](#)
- CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_START\_ADDR  
eHSM\_IntCfg\_Ip.h, [821](#)



CONFIG\_EHSM\_KMGR\_V\_SYM\_RAM\_K\_NUM  
  eHSM\_IntCfg\_Ip.h, [821](#)

CONFIG\_EHSM\_LOG  
  eHSM\_IntCfg\_Ip.h, [821](#)

CONFIG\_EHSM\_PKE\_CORE\_NUM  
  eHSM\_IntCfg\_Ip.h, [821](#)

CONFIG\_EHSM\_SHE\_SOC\_BOOT  
  eHSM\_IntCfg\_Ip.h, [821](#)

CONFIG\_EHSM\_SHE  
  eHSM\_Config\_Ip.h, [485](#)

CONFIG\_EHSM\_SKE\_CORE\_NUM  
  eHSM\_IntCfg\_Ip.h, [822](#)

CONFIG\_EHSM\_SOC\_UPGRADE\_AND\_VERIFY  
  eHSM\_IntCfg\_Ip.h, [822](#)

CONFIG\_EHSM\_TRNG\_CORE\_NUM  
  eHSM\_IntCfg\_Ip.h, [822](#)

CONFIG\_EHSM\_UNIT\_TEST\_EVITA\_SYM  
  test\_int\_config.h, [994](#)

CONFIG\_EHSM\_UNIT\_TEST  
  eHSM\_IntCfg\_Ip.h, [822](#)

CONFIG\_EHSM\_USER\_AUTH\_KYE\_IN\_KMU  
  eHSM\_IntCfg\_Ip.h, [822](#)

CONFIG\_EHSM\_V\_LOG\_DEBUG  
  eHSM\_IntCfg\_Ip.h, [823](#)

CONFIG\_EHSM\_V\_LOG\_ERR  
  eHSM\_IntCfg\_Ip.h, [823](#)

CONFIG\_EHSM\_V\_LOG\_INFO  
  eHSM\_IntCfg\_Ip.h, [823](#)

CONFIG\_EHSM\_V\_LOG\_LEVEL  
  eHSM\_IntCfg\_Ip.h, [823](#)

CONFIG\_EHSM\_V\_LOG\_WARN  
  eHSM\_IntCfg\_Ip.h, [823](#)

CONFIG\_HOST\_AUTOSAR\_DEBUG\_ENABLE  
  eHSM\_Debug\_Ip.h, [488](#)

CONFIG\_HOST\_COMMON\_DEBUG\_ENABLE  
  eHSM\_Debug\_Ip.h, [488](#)

CONFIG\_HOST\_CUSTOM\_DEBUG\_ENABLE  
  eHSM\_Debug\_Ip.h, [488](#)

CONFIG\_HOST\_EVITA\_DEBUG\_ENABLE  
  eHSM\_Debug\_Ip.h, [489](#)

CONFIG\_HOST\_PERFORMANCE\_DEBUG\_ENABLE  
  eHSM\_Debug\_Ip.h, [489](#)

CONFIG\_HOST\_SHE\_DEBUG\_ENABLE  
  eHSM\_Debug\_Ip.h, [489](#)

CONFIG\_HOST\_V\_LOG\_DEBUG  
  eHSM\_Debug\_Ip.h, [489](#)

CONFIG\_HOST\_V\_LOG\_ERR  
  eHSM\_Debug\_Ip.h, [489](#)

CONFIG\_HOST\_V\_LOG\_INFO  
  eHSM\_Debug\_Ip.h, [489](#)

CONFIG\_HOST\_V\_LOG\_LEVEL  
  eHSM\_Debug\_Ip.h, [490](#)

CONFIG\_HOST\_V\_LOG\_WARN  
  eHSM\_Debug\_Ip.h, [490](#)

CONFIG\_SM9\_ENC\_DEFAULT\_HID\_VALUE  
  eHSM\_If\_Ext\_Types\_Ip.h, [753](#)

CONFIG\_SM9\_EXCHG\_DEFAULT\_HID\_VALUE  
  eHSM\_If\_Ext\_Types\_Ip.h, [753](#)

CONFIG\_SM9\_SIGN\_DEFAULT\_HID\_VALUE  
  eHSM\_If\_Ext\_Types\_Ip.h, [754](#)

CRYPTO\_CERTIFICATE\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [582](#)

CRYPTO\_E\_BUSY  
  Asr\_Standard\_Types.h, [394](#)

CRYPTO\_E\_ENTROPY\_EXHAUSTION  
  Asr\_Standard\_Types.h, [394](#)

CRYPTO\_E\_JOB\_CANCELED  
  Asr\_Standard\_Types.h, [394](#)

CRYPTO\_E\_KEY\_EMPTY  
  Asr\_Standard\_Types.h, [394](#)

CRYPTO\_E\_KEY\_NOT\_AVAILABLE  
  Asr\_Standard\_Types.h, [395](#)

CRYPTO\_E\_KEY\_NOT\_VALID  
  Asr\_Standard\_Types.h, [395](#)

CRYPTO\_E\_KEY\_READ\_FAIL  
  Asr\_Standard\_Types.h, [395](#)

CRYPTO\_E\_KEY\_SIZE\_MISMATCH  
  Asr\_Standard\_Types.h, [395](#)

CRYPTO\_E\_KEY\_WRITE\_FAIL  
  Asr\_Standard\_Types.h, [395](#)

CRYPTO\_E\_QUEUE\_FULL  
  Asr\_Standard\_Types.h, [396](#)

CRYPTO\_E\_SMALL\_BUFFER  
  Asr\_Standard\_Types.h, [396](#)

CRYPTO\_EVITA\_CERTIFICATE\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [582](#)

CRYPTO\_EVITA\_CIPHER\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [582](#)

CRYPTO\_EVITA\_CIPHER\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [583](#)

CRYPTO\_EVITA\_COPY\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [583](#)

CRYPTO\_EVITA\_DERIVE\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [583](#)

CRYPTO\_EVITA\_DERIVE\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [583](#)

CRYPTO\_EVITA\_EXCHANGE\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [584](#)

CRYPTO\_EVITA\_EXCHANGE\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [584](#)

CRYPTO\_EVITA\_EXPORT\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [584](#)

CRYPTO\_EVITA\_EXPORT\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [584](#)

CRYPTO\_EVITA\_GENERATE\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [584](#)

CRYPTO\_EVITA\_GENERATE\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [585](#)

CRYPTO\_EVITA\_IMPORT\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [585](#)

CRYPTO\_EVITA\_IMPORT\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [585](#)

CRYPTO\_EVITA\_KEY\_COPY\_KEY\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [585](#)

CRYPTO\_EVITA\_KEY\_STATUS\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [585](#)

CRYPTO\_EVITA\_KEY\_STATUS\_NUM  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [586](#)

CRYPTO\_EVITA\_MAC\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [586](#)

CRYPTO\_EVITA\_REMOVE\_KEY\_ELEMENT\_SIZE  
  eHSM\_If\_Asr\_KeyCfg\_Ip.h, [586](#)

CRYPTO\_EVITA\_REMOVE\_KEY\_NUM  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [586](#)

CRYPTO\_EVITA\_SIGNATURE\_KEY\_ELEMENT\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [586](#)

CRYPTO\_EVITA\_SIGNATURE\_KEY\_NUM  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [586](#)

CRYPTO\_INDEX\_M1\_U32  
    eHSM\_If\_Asr\_Types\_Ip.h, [605](#)

CRYPTO\_INDEX\_M2\_U32  
    eHSM\_If\_Asr\_Types\_Ip.h, [605](#)

CRYPTO\_INDEX\_M3\_U32  
    eHSM\_If\_Asr\_Types\_Ip.h, [605](#)

CRYPTO\_INDEX\_M4\_U32  
    eHSM\_If\_Asr\_Types\_Ip.h, [605](#)

CRYPTO\_INDEX\_M5\_U32  
    eHSM\_If\_Asr\_Types\_Ip.h, [606](#)

CRYPTO\_INVALID\_KEY\_ID  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [587](#)

CRYPTO\_KE\_AEAD\_TAG\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [587](#)

CRYPTO\_KE\_CERTIFICATE\_CURRENT\_TIME  
    Asr\_Standard\_Types.h, [396](#)

CRYPTO\_KE\_CERTIFICATE\_DATA  
    Asr\_Standard\_Types.h, [396](#)

CRYPTO\_KE\_CERTIFICATE\_EXTENSIONS  
    Asr\_Standard\_Types.h, [396](#)

CRYPTO\_KE\_CERTIFICATE\_ISSUER  
    Asr\_Standard\_Types.h, [396](#)

CRYPTO\_KE\_CERTIFICATE\_PARSING\_FORMAT  
    Asr\_Standard\_Types.h, [397](#)

CRYPTO\_KE\_CERTIFICATE\_SERIALNUMBER  
    Asr\_Standard\_Types.h, [397](#)

CRYPTO\_KE\_CERTIFICATE\_SIGNATURE\_ALGORITHM  
    Asr\_Standard\_Types.h, [397](#)

CRYPTO\_KE\_CERTIFICATE\_SIGNATURE  
    Asr\_Standard\_Types.h, [397](#)

CRYPTO\_KE\_CERTIFICATE\_SIGNEDDATA  
    eHSM\_If\_Asr\_Types\_Ip.h, [606](#)

CRYPTO\_KE\_CERTIFICATE\_SUBJECT\_PUBLIC\_KEY  
    Asr\_Standard\_Types.h, [397](#)

CRYPTO\_KE\_CERTIFICATE\_SUBJECT  
    Asr\_Standard\_Types.h, [397](#)

CRYPTO\_KE\_CERTIFICATE\_VALIDITY\_NOT\_AFTER  
    Asr\_Standard\_Types.h, [398](#)

CRYPTO\_KE\_CERTIFICATE\_VALIDITY\_NOT\_BEFORE  
    Asr\_Standard\_Types.h, [398](#)

CRYPTO\_KE\_CERTIFICATE\_VERSION  
    Asr\_Standard\_Types.h, [398](#)

CRYPTO\_KE\_CIPHER\_2NDKEY\_SIZE\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [587](#)

CRYPTO\_KE\_CIPHER\_2NDKEY  
    Asr\_Standard\_Types.h, [398](#)

CRYPTO\_KE\_CIPHER\_CIPHER\_ALG\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [587](#)

CRYPTO\_KE\_CIPHER\_CIPHER\_ALG  
    eHSM\_If\_Asr\_Types\_Ip.h, [606](#)

CRYPTO\_KE\_CIPHER\_CURVE\_ID\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [587](#)

CRYPTO\_KE\_CIPHER\_CURVE\_ID  
    eHSM\_If\_Asr\_Types\_Ip.h, [606](#)

CRYPTO\_KE\_CIPHER\_IV\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [587](#)

CRYPTO\_KE\_CIPHER\_IV  
    Asr\_Standard\_Types.h, [398](#)

CRYPTO\_KE\_CIPHER\_KDF\_ALG\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [588](#)

CRYPTO\_KE\_CIPHER\_KDF\_ALG  
    eHSM\_If\_Asr\_Types\_Ip.h, [606](#)

CRYPTO\_KE\_CIPHER\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [588](#)

CRYPTO\_KE\_CIPHER\_KEY  
    Asr\_Standard\_Types.h, [398](#)

CRYPTO\_KE\_CIPHER\_MAC\_ALG\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [588](#)

CRYPTO\_KE\_CIPHER\_MAC\_ALG  
    eHSM\_If\_Asr\_Types\_Ip.h, [606](#)

CRYPTO\_KE\_CIPHER\_MAC\_SIZE\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [588](#)

CRYPTO\_KE\_CIPHER\_MAC\_SIZE  
    eHSM\_If\_Asr\_Types\_Ip.h, [607](#)

CRYPTO\_KE\_CIPHER\_PROOF\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [588](#)

CRYPTO\_KE\_CIPHER\_PROOF  
    Asr\_Standard\_Types.h, [399](#)

CRYPTO\_KE\_CIPHER\_TAG\_SIZE  
    eHSM\_If\_Asr\_Types\_Ip.h, [607](#)

CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [588](#)

CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY  
    eHSM\_If\_Asr\_Types\_Ip.h, [607](#)

CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [589](#)

CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE  
    eHSM\_If\_Asr\_Types\_Ip.h, [607](#)

CRYPTO\_KE\_EXPORT\_KEY\_BLOB\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [589](#)

CRYPTO\_KE\_EXPORT\_KEY\_BLOB  
    eHSM\_If\_Asr\_Types\_Ip.h, [607](#)

CRYPTO\_KE\_EXPORT\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [589](#)

CRYPTO\_KE\_EXPORT\_KEY  
    eHSM\_If\_Asr\_Types\_Ip.h, [607](#)

CRYPTO\_KE\_EXT\_SHE\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [589](#)

CRYPTO\_KE\_EXT\_SHE\_KEY  
    eHSM\_If\_Asr\_Types\_Ip.h, [608](#)

CRYPTO\_KE\_IMPORT\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [589](#)

CRYPTO\_KE\_IMPORT\_KEY  
    eHSM\_If\_Asr\_Types\_Ip.h, [608](#)

CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [589](#)

CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE  
    eHSM\_If\_Asr\_Types\_Ip.h, [608](#)

CRYPTO\_KE\_KEY\_HANDLE  
    eHSM\_If\_Asr\_Types\_Ip.h, [608](#)

CRYPTO\_KE\_KEY\_MATERIAL\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_Ip.h, [590](#)

CRYPTO\_KE\_KEY\_MATERIAL  
    eHSM\_If\_Asr\_Types\_Ip.h, [608](#)

CRYPTO\_KE\_KEY\_STATUS\_BLOB\_SIZE

eHSM\_If\_Asr\_KeyCfg\_lp.h, [590](#)  
CRYPTO\_KE\_KEY\_STATUS\_BLOB  
eHSM\_If\_Asr\_Types\_lp.h, [609](#)  
CRYPTO\_KE\_KEY\_STATUS\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [590](#)  
CRYPTO\_KE\_KEY\_STATUS  
eHSM\_If\_Asr\_Types\_lp.h, [608](#)  
CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [590](#)  
CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM  
Asr\_Standard\_Types.h, [399](#)  
CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [590](#)  
CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS  
Asr\_Standard\_Types.h, [399](#)  
CRYPTO\_KE\_KEYDERIVATION\_KEY\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [590](#)  
CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [591](#)  
CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE  
eHSM\_If\_Asr\_Types\_lp.h, [609](#)  
CRYPTO\_KE\_KEYDERIVATION\_KEY  
eHSM\_If\_Asr\_Types\_lp.h, [609](#)  
CRYPTO\_KE\_KEYDERIVATION\_PASSWD\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [591](#)  
CRYPTO\_KE\_KEYDERIVATION\_PASSWD  
Asr\_Standard\_Types.h, [399](#)  
CRYPTO\_KE\_KEYDERIVATION\_SALT\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [591](#)  
CRYPTO\_KE\_KEYDERIVATION\_SALT  
Asr\_Standard\_Types.h, [399](#)  
CRYPTO\_KE\_KEYDERIVATION\_TYPE\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [591](#)  
CRYPTO\_KE\_KEYDERIVATION\_TYPE  
eHSM\_If\_Asr\_Types\_lp.h, [609](#)  
CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [591](#)  
CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM  
Asr\_Standard\_Types.h, [399](#)  
CRYPTO\_KE\_KEYEXCHANGE\_BASE\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [591](#)  
CRYPTO\_KE\_KEYEXCHANGE\_BASE  
Asr\_Standard\_Types.h, [400](#)  
CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [592](#)  
CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO  
eHSM\_If\_Asr\_Types\_lp.h, [609](#)  
CRYPTO\_KE\_KEYEXCHANGE\_OWNPUBLICKEY\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [592](#)  
CRYPTO\_KE\_KEYEXCHANGE\_OWNPUBLICKEY  
Asr\_Standard\_Types.h, [400](#)  
CRYPTO\_KE\_KEYEXCHANGE\_PEERPUBLICKEY\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [592](#)  
CRYPTO\_KE\_KEYEXCHANGE\_PEERPUBLICKEY  
eHSM\_If\_Asr\_Types\_lp.h, [610](#)  
CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [592](#)  
CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY  
Asr\_Standard\_Types.h, [400](#)  
CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [592](#)

CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE  
eHSM\_If\_Asr\_Types\_lp.h, [610](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINF↔  
O\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [592](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCALTMPKINFO  
eHSM\_If\_Asr\_Types\_lp.h, [610](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK↔  
\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [593](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEERTMPPUBK  
eHSM\_If\_Asr\_Types\_lp.h, [610](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [593](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE  
eHSM\_If\_Asr\_Types\_lp.h, [610](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE↔  
SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [593](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE  
eHSM\_If\_Asr\_Types\_lp.h, [610](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE↔  
SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [593](#)  
CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE  
eHSM\_If\_Asr\_Types\_lp.h, [611](#)  
CRYPTO\_KE\_KEYGENERATE\_ALGORITHM\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [593](#)  
CRYPTO\_KE\_KEYGENERATE\_ALGORITHM  
Asr\_Standard\_Types.h, [400](#)  
CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [593](#)  
CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO  
eHSM\_If\_Asr\_Types\_lp.h, [611](#)  
CRYPTO\_KE\_KEYGENERATE\_KEY\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [594](#)  
CRYPTO\_KE\_KEYGENERATE\_KEYINFO\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [594](#)  
CRYPTO\_KE\_KEYGENERATE\_KEYINFO  
eHSM\_If\_Asr\_Types\_lp.h, [611](#)  
CRYPTO\_KE\_KEYGENERATE\_KEY  
Asr\_Standard\_Types.h, [400](#)  
CRYPTO\_KE\_KEYGENERATE\_SEED\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [594](#)  
CRYPTO\_KE\_KEYGENERATE\_SEED  
Asr\_Standard\_Types.h, [400](#)  
CRYPTO\_KE\_MAC\_KEY\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [594](#)  
CRYPTO\_KE\_MAC\_KEY  
Asr\_Standard\_Types.h, [401](#)  
CRYPTO\_KE\_MAC\_PROOF\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [594](#)  
CRYPTO\_KE\_MAC\_PROOF  
Asr\_Standard\_Types.h, [401](#)  
CRYPTO\_KE\_RANDOM\_ALGORITHM\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [594](#)  
CRYPTO\_KE\_RANDOM\_ALGORITHM  
Asr\_Standard\_Types.h, [401](#)  
CRYPTO\_KE\_RANDOM\_SEED\_STATE\_SIZE  
eHSM\_If\_Asr\_KeyCfg\_lp.h, [595](#)  
CRYPTO\_KE\_RANDOM\_SEED\_STATE

Asr\_Standard\_Types.h, [401](#)  
CRYPTO\_KE\_REMOVE\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [595](#)  
CRYPTO\_KE\_REMOVE\_KEY  
    eHSM\_If\_Asr\_Types\_lp.h, [611](#)  
CRYPTO\_KE\_SHE\_PLAIN\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [595](#)  
CRYPTO\_KE\_SIGNATURE\_KEY\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [595](#)  
CRYPTO\_KE\_SIGNATURE\_KEY  
    Asr\_Standard\_Types.h, [401](#)  
CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [595](#)  
CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE  
    eHSM\_If\_Asr\_Types\_lp.h, [611](#)  
CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [595](#)  
CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED  
    eHSM\_If\_Asr\_Types\_lp.h, [611](#)  
CRYPTO\_KEY10\_EVITA\_IMPORT\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [596](#)  
CRYPTO\_KEY11\_EVITA\_EXPORT\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [596](#)  
CRYPTO\_KEY12\_EVITA\_REMOVE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [596](#)  
CRYPTO\_KEY13\_EVITA\_KEY\_STATUS  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [596](#)  
CRYPTO\_KEY14\_EVITA\_KEY\_COPY\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [596](#)  
CRYPTO\_KEY15\_EVITA\_KEY\_COPY\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [596](#)  
CRYPTO\_KEY16\_CERTIFICATE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [597](#)  
CRYPTO\_KEY17\_CERTIFICATE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [597](#)  
CRYPTO\_KEY18\_MAC\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [597](#)  
CRYPTO\_KEY1\_SHE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [597](#)  
CRYPTO\_KEY2\_SHE\_PLAIN\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [597](#)  
CRYPTO\_KEY3\_EVITA\_SIGNATURE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [597](#)  
CRYPTO\_KEY4\_EVITA\_CIPHER\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [598](#)  
CRYPTO\_KEY5\_EVITA\_EXCHANGE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [598](#)  
CRYPTO\_KEY6\_EVITA\_EXCHANGE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [598](#)  
CRYPTO\_KEY7\_EVITA\_DERIVE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [598](#)  
CRYPTO\_KEY8\_EVITA\_DERIVE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [598](#)  
CRYPTO\_KEY9\_EVITA\_GENERATE\_KEY  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [598](#)  
CRYPTO\_KEY\_ATTR\_ALLOW\_PARTIAL\_ACCESS  
    eHSM\_If\_Asr\_Types\_lp.h, [612](#)  
CRYPTO\_KEY\_ATTR\_READ\_ACCESS  
    eHSM\_If\_Asr\_Types\_lp.h, [612](#)  
CRYPTO\_KEY\_ATTR\_WRITE\_ACCESS  
    eHSM\_If\_Asr\_Types\_lp.h, [612](#)  
CRYPTO\_M1\_SIZE\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [612](#)  
CRYPTO\_M2\_SIZE\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [612](#)  
CRYPTO\_M3\_SIZE\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [612](#)  
CRYPTO\_M4\_SIZE\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [613](#)  
CRYPTO\_M5\_SIZE\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [613](#)  
CRYPTO\_MAC\_KEY\_NUM  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [599](#)  
CRYPTO\_MAX\_AUTH\_VALUE\_SIZE  
    eHSM\_If\_Asr\_Types\_lp.h, [613](#)  
CRYPTO\_MAX\_KEY\_ELEMENTS\_OF\_KEY\_TYPE  
    eHSM\_If\_Asr\_Types\_lp.h, [613](#)  
CRYPTO\_MAX\_KEY\_FLAG\_ELEMENT  
    eHSM\_If\_Asr\_Types\_lp.h, [613](#)  
CRYPTO\_MAX\_KEY\_NUM  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [599](#)  
CRYPTO\_SHE\_KEY\_ELEMENT\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [599](#)  
CRYPTO\_SHE\_KEY\_NUM  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [599](#)  
CRYPTO\_SHE\_MAC\_KEY\_ID  
    eHSM\_If\_Asr\_Types\_lp.h, [613](#), [614](#)  
CRYPTO\_SHE\_MAC\_KEY\_NUM  
    eHSM\_If\_Asr\_Types\_lp.h, [614](#)  
CRYPTO\_SHE\_PLAIN\_KEY\_ELEMENT\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [599](#)  
CRYPTO\_SHE\_PLAIN\_KEY\_NUM  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [600](#)  
CRYPTO\_SHE\_SIZE\_IN\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [614](#)  
CRYPTO\_SHE\_SIZE\_OUT\_U32  
    eHSM\_If\_Asr\_Types\_lp.h, [614](#)  
CUSTOM\_LOG\_DEBUG  
    eHSM\_Debug\_lp.h, [490](#)  
CUSTOM\_LOG\_ERROR  
    eHSM\_Debug\_lp.h, [490](#)  
CUSTOM\_LOG\_INFO  
    eHSM\_Debug\_lp.h, [490](#)  
CUSTOM\_LOG\_WARN  
    eHSM\_Debug\_lp.h, [490](#)  
CUSTOM\_PURE\_LOG\_DEBUG  
    eHSM\_Debug\_lp.h, [491](#)  
CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE\_SIZE  
    eHSM\_If\_Asr\_KeyCfg\_lp.h, [600](#)  
CYRPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE  
    Asr\_Standard\_Types.h, [401](#)  
callbackId  
    Crypto\_JobPrimitiveInfoType, [34](#)  
callbackUpdateNotification  
    Crypto\_JobPrimitiveInfoType, [34](#)  
cancel\_type  
    ehsm\_mbox\_cancel\_channel\_req, [204](#)  
    ehsm\_mbox\_cancel\_channel\_rps, [205](#)  
cb  
    ehsm\_service\_info, [235](#)  
cert\_data  
    eHSM\_If\_Evita\_Types\_lp.h, [709](#)

- ehsm\_key\_status\_, 186
- cert\_key\_auth\_size
  - ehsm\_key\_status\_cmd, 188
- cert\_key\_auth\_value
  - ehsm\_key\_status\_cmd, 188
- cert\_key\_handle
  - ehsm\_key\_status\_cmd, 188
- cert\_size
  - eHSM\_If\_Evita\_Types\_lp.h, 709
  - ehsm\_key\_status\_, 186
- certificate\_info
  - ehsm\_certificate\_verify\_st, 65
- certificate\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, 573
- certification\_key\_auth\_size
  - crypto\_key\_status\_info, 47
  - ehsm\_key\_status\_param, 190
- certification\_key\_auth\_value
  - crypto\_key\_status\_info, 47
  - ehsm\_key\_status\_param, 190
- certification\_key\_handle
  - crypto\_key\_status\_info, 47
  - ehsm\_key\_status\_param, 190
- CertificationAuth
  - HSM\_KeyStatusType, 318
- CertificationAuthSize
  - HSM\_KeyStatusType, 318
- CertificationKeyId
  - HSM\_KeyStatusType, 318
- challenge
  - ehsm\_she\_get\_id\_param\_st, 236
- challenge\_addr
  - ehsm\_get\_she\_id\_cmd, 160
- challenge\_size
  - ehsm\_get\_she\_id\_cmd, 160
  - ehsm\_she\_get\_id\_param\_st, 236
- change\_control\_field\_cmd
  - ehsm\_mbox\_mgr\_channel\_req, 206
- change\_lifecycle\_cmd
  - ehsm\_mbox\_mgr\_channel\_req, 206
- channel
  - ehsm\_cmd\_req, 84
- Check\_Time\_Stamp
  - eHSM\_If\_Evita\_lp.h, 637
- check\_version\_flag
  - soc\_image\_upgrade\_info, 368
  - soc\_image\_upgrade\_input, 373
- CheckVersionFlag
  - HSM\_SecureUpgradeType\_, 335
- Cipher\_Finish
  - eHSM\_If\_Evita\_lp.h, 638
  - eHSM\_If\_Evita\_SymCper\_lp.c, 679
- Cipher\_Init
  - eHSM\_If\_Evita\_lp.h, 639
  - eHSM\_If\_Evita\_SymCper\_lp.c, 680
- Cipher\_Process
  - eHSM\_If\_Evita\_lp.h, 640
  - eHSM\_If\_Evita\_SymCper\_lp.c, 681
- cipher\_addr
  - ehsm\_sm9\_unwrap\_key\_cmd, 264
  - ehsm\_sm9\_unwrap\_key\_param, 266
- cipher\_alg
  - ehsm\_cmd\_hdr\_ecise\_st, 75
- cipher\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, 573
- cipher\_mode
  - cipher\_session\_st, 15
  - ehsm\_cmd\_hdr\_ske\_st, 81
- cipher\_mode\_e
  - eHSM\_If\_Evita\_Types\_lp.h, 705
- cipher\_part1
  - ecies\_testvec, 60
- cipher\_part2\_part3\_with\_s1
  - ecies\_testvec, 60
- cipher\_part2\_part3\_with\_s1\_s2
  - ecies\_testvec, 60
- cipher\_part2\_part3\_with\_s2
  - ecies\_testvec, 60
- cipher\_part2\_part3\_without\_s1\_s2
  - ecies\_testvec, 60
- cipher\_session\_st, 14
  - aead\_data, 15
  - algorithm, 15
  - cipher\_mode, 15
  - cmd\_id, 16
  - ctx, 16
  - ctx\_block\_mgr, 16
  - direction, 16
  - is\_hmac, 16
  - job\_id, 16
  - key\_auth\_addr, 17
  - key\_auth\_size, 17
  - key\_handle, 17
  - key\_type, 17
  - mac\_bytes, 17
  - padding, 17
  - rsa\_crt\_mode, 18
  - signature\_addr, 18
  - signature\_size, 18
  - status, 18
  - time\_stamp, 18
  - utc\_time, 18
- cipher\_size
  - ehsm\_sm9\_unwrap\_key\_cmd, 264
  - ehsm\_sm9\_unwrap\_key\_param, 266
- cipher\_testvec, 19
  - clen, 19
  - crypt\_error, 19
  - ctxt, 20
  - fips\_skip, 20
  - iv, 20
  - iv\_out, 20
  - key, 20
  - klen, 20
  - len, 21
  - ptext, 21
  - setkey\_error, 21
  - wk, 21
- CipherDir
  - HSM\_AsymCfgType, 286
  - HSM\_SymCfgType, 340
- clen



- aead\_testvec, 9
- cipher\_testvec, 19
- close\_debug
  - ehsm\_mailbox\_req, 197
- cmd\_data
  - ehsm\_cmd\_req, 84
- cmd\_id
  - cipher\_session\_st, 16
  - ehsm\_cmd\_cipher\_st, 71
  - ehsm\_cmd\_req, 85
  - ehsm\_mailbox\_req, 197
  - ehsm\_mbox\_mgr\_channel\_req, 206
- cmd\_inprogres
  - mailbox\_channel, 353
- cmd\_input
  - soc\_image\_upgrade\_info, 368
  - soc\_image\_verify\_info, 378
- cmd\_limit
  - crypto\_object, 48
- cmd\_list
  - crypto\_object, 48
- cmd\_release\_cb
  - eHSM\_Srv\_CmdReq\_Ip.h, 887
- cmd\_req
  - ehsm\_cmd\_req\_buffer, 87
- cmd\_req\_cb
  - eHSM\_Srv\_CmdReq\_Ip.h, 887
- cmd\_sent
  - crypto\_object, 49
- cmd\_sent\_num
  - crypto\_object, 49
- cmd\_size
  - ehsm\_cmd\_req, 85
- cmd\_state
  - ehsm\_cmd\_req, 85
- cmd\_tag
  - ehsm\_mbox\_cancel\_channel\_req, 204
  - ehsm\_mbox\_cancel\_channel\_rps, 205
- cmd\_type\_e
  - eHSM\_Mailbox\_CmdId\_Ip.h, 844
- container\_of
  - eHSM\_Compt\_List.h, 438
- context\_addr
  - ehsm\_cmd\_cipher\_st, 71
- context\_size
  - ehsm\_cmd\_cipher\_st, 71
- copy\_key
  - ehsm\_mailbox\_req, 197
- copy\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 573
- counter
  - ehsm\_she\_key\_st, 241
- counter\_value\_64\_t, 21
  - high\_word, 22
  - low\_word, 22
- Create\_Counter
  - eHSM\_If\_Evita\_Ip.h, 640
- Create\_Derived\_Key
  - eHSM\_If\_Evita\_Ip.h, 641
  - eHSM\_If\_Evita\_Key\_Ip.c, 665
- Create\_Dh\_Key
  - eHSM\_If\_Evita\_Ip.h, 642
  - eHSM\_If\_Evita\_Key\_Ip.c, 666
- Create\_Random\_Dh\_Key\_Pair
  - eHSM\_If\_Evita\_Ip.h, 643
  - eHSM\_If\_Evita\_Key\_Ip.c, 667
- Create\_Random\_Key
  - eHSM\_If\_Evita\_Ip.h, 644
  - eHSM\_If\_Evita\_Key\_Ip.c, 668
- Create\_Time\_Stamp
  - eHSM\_If\_Evita\_Ip.h, 645
- create\_dh\_key
  - ehsm\_mailbox\_req, 197
- createkey
  - HSM\_KeyActUseFlagsType, 312
  - key\_act\_use\_flags\_t, 344
- CryIf\_CallbackNotification
  - eHSM\_If\_Asr\_Types\_Ip.h, 619
- cryIfKeyId
  - Crypto\_JobPrimitiveInfoType, 34
  - Crypto\_JobPrimitiveInputOutputType, 35
- crypt\_error
  - aead\_testvec, 9
  - cipher\_testvec, 19
- Crypto\_AlgorithmFamilyType
  - Asr\_Standard\_Types.h, 402
- Crypto\_AlgorithmInfoType, 22
  - family, 23
  - keyLength, 23
  - mode, 23
  - secondaryFamily, 23
- Crypto\_AlgorithmModeType
  - Asr\_Standard\_Types.h, 403
- Crypto\_ConfigType
  - Asr\_Standard\_Types.h, 402
- Crypto\_GetJobKey
  - eHSM\_If\_Asr\_Job\_Ip.h, 568
- Crypto\_GetKeyId
  - eHSM\_If\_Asr\_Job\_Ip.h, 568
- Crypto\_InputOutputRedirectionConfigType
  - Asr\_Standard\_Types.h, 404
- Crypto\_JobInfoType, 32
  - jobId, 33
  - jobPriority, 33
- Crypto\_JobPrimitiveInfoType, 33
  - callbackId, 34
  - callbackUpdateNotification, 34
  - cryIfKeyId, 34
  - primitiveInfo, 34
  - processingType, 34
- Crypto\_JobPrimitiveInputOutputType, 34
  - cryIfKeyId, 35
  - input64, 35
  - inputLength, 35
  - inputPtr, 35
  - mode, 36
  - output64Ptr, 36
  - outputLengthPtr, 36
  - outputPtr, 36
  - secondaryInputLength, 36
  - secondaryInputPtr, 36
  - secondaryOutputLengthPtr, 37

- secondaryOutputPtr, 37
- targetCryIfKeyId, 37
- tertiaryInputLength, 37
- tertiaryInputPtr, 37
- verifyPtr, 37
- Crypto\_JobRedirectionInfoType, 38
  - inputKeyElementId, 38
  - inputKeyId, 38
  - outputKeyElementId, 39
  - outputKeyId, 39
  - redirectionConfig, 39
  - secondaryInputKeyElementId, 39
  - secondaryInputKeyId, 39
  - secondaryOutputKeyElementId, 39
  - secondaryOutputKeyId, 40
  - tertiaryInputKeyElementId, 40
  - tertiaryInputKeyId, 40
- Crypto\_JobStateType
  - Asr\_Standard\_Types.h, 404
- Crypto\_JobType, 40
  - cryptoKeyId, 41
  - jobId, 41
  - jobInfo, 41
  - jobPrimitiveInfo, 41
  - jobPrimitiveInputOutput, 41
  - jobRedirectionInfoRef, 41
  - jobState, 42
- Crypto\_KeyElementReadAccessType
  - Asr\_Standard\_Types.h, 405
- Crypto\_KeyElementWriteAccessType
  - Asr\_Standard\_Types.h, 405
- Crypto\_MainFunction
  - eHSM\_Dspt\_Ip.c, 503
  - eHSM\_Dspt\_Ip.h, 505
- Crypto\_OperationModeType
  - Asr\_Standard\_Types.h, 405
- Crypto\_PrimitiveInfoType, 50
  - algorithm, 50
  - resultLength, 50
  - service, 50
- Crypto\_ProcessingType
  - Asr\_Standard\_Types.h, 406
- Crypto\_ServiceInfoType
  - Asr\_Standard\_Types.h, 406
- Crypto\_VerifyResultType
  - Asr\_Standard\_Types.h, 406
- crypto\_copy\_key\_dh\_key\_info, 23
  - g, 24
  - g\_size, 24
  - p, 24
  - p\_size, 24
  - q, 24
  - q\_size, 24
- crypto\_copy\_key\_dh\_key\_info\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 616
- crypto\_copy\_key\_info, 25
  - auth\_size, 25
  - auth\_value, 25
  - element\_data, 25
  - element\_size, 26
  - key\_handle, 26
- crypto\_copy\_key\_info\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 616
- crypto\_create\_evita\_key\_info, 26
  - element\_data, 26
  - element\_size, 27
  - key\_size, 27
  - type, 27
  - valid\_until, 27
- crypto\_create\_evita\_key\_info\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 616
- crypto\_evita\_key\_info, 27
  - auth\_size, 28
  - auth\_value, 28
  - key\_handle, 28
- crypto\_evita\_key\_info\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 617
- crypto\_exported\_key, 28
  - encrypted\_key, 29
  - encrypted\_key\_buffer\_size, 29
  - encrypted\_key\_size, 29
  - key\_auth\_code, 29
  - key\_auth\_code\_buffer\_size, 29
  - key\_auth\_code\_size, 30
- crypto\_exported\_key\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 617
- crypto\_import\_evita\_key\_info, 30
  - authenticity\_key\_authorization, 30
  - authenticity\_key\_authorization\_size, 31
  - authenticity\_key\_handle, 31
  - encrypted\_key, 31
  - encrypted\_key\_size, 31
  - key\_authenticity\_code, 31
  - key\_authenticity\_code\_size, 31
  - transport\_key\_authorization, 32
  - transport\_key\_authorization\_size, 32
  - transport\_key\_handle, 32
  - type, 32
- crypto\_import\_evita\_key\_info\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 617
- crypto\_key
  - ehsm\_crypto\_key, 103
- crypto\_key\_derive\_info, 42
  - derive\_type, 42
  - itera\_times, 42
  - key\_info, 43
  - passwd, 43
  - passwd\_size, 43
- crypto\_key\_derive\_info\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, 617
- crypto\_key\_derive\_type\_e
  - eHSM\_Com\_Struct\_Ip.h, 425
- crypto\_key\_element\_type\_info\_st, 43
  - CryptoKeyElementAllowPartialAccess, 44
  - CryptoKeyElementId, 44
  - CryptoKeyElementMaxSize, 44
  - CryptoKeyElementPersist, 44
  - CryptoKeyElementReadAccess, 44
  - CryptoKeyElementWriteAccess, 44
  - CryptoKeyFormat, 45
- crypto\_key\_export\_info, 45
  - authenticity\_key\_authorization, 45

- authenticity\_key\_authorization\_size, [45](#)
  - authenticity\_key\_handle, [46](#)
  - key\_handle, [46](#)
  - transport\_key\_authorization, [46](#)
  - transport\_key\_authorization\_size, [46](#)
  - transport\_key\_handle, [46](#)
  - use\_flags, [46](#)
- crypto\_key\_export\_info\_st
  - eHSM\_If\_Asr\_Types\_lp.h, [617](#)
- crypto\_key\_status\_info, [47](#)
  - certification\_key\_auth\_size, [47](#)
  - certification\_key\_auth\_value, [47](#)
  - certification\_key\_handle, [47](#)
  - key\_handle, [48](#)
- crypto\_key\_status\_info\_st
  - eHSM\_If\_Asr\_Types\_lp.h, [617](#)
- crypto\_key\_type\_e
  - eHSM\_If\_Asr\_Types\_lp.h, [617](#)
- crypto\_keys
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, [573](#)
- crypto\_object, [48](#)
  - cmd\_limit, [48](#)
  - cmd\_list, [48](#)
  - cmd\_sent, [49](#)
  - cmd\_sent\_num, [49](#)
  - name, [49](#)
  - queue\_capacity, [49](#)
  - state, [49](#)
  - type, [49](#)
- crypto\_object\_st
  - eHSM\_Dspt\_CryObj\_lp.h, [499](#)
- crypto\_object\_state\_e
  - eHSM\_Dspt\_CryObj\_lp.h, [499](#)
- crypto\_object\_type\_e
  - eHSM\_If\_Asr\_KeyCfg\_lp.h, [601](#)
- crypto\_she\_key, [51](#)
  - m1, [51](#)
  - m2, [51](#)
  - m3, [51](#)
  - m4, [52](#)
  - m5, [52](#)
  - she\_ext\_flag, [52](#)
- crypto\_she\_key\_st
  - eHSM\_If\_Asr\_Types\_lp.h, [617](#)
- CryptoDriverObject, [52](#)
  - CryptoDriverObjectEcucPartitionRef, [53](#)
  - CryptoDriverObjectId, [53](#)
  - CryptoPrimitiveNum, [53](#)
  - CryptoPrimitiveRef, [53](#)
  - CryptoQueueSize, [53](#)
- CryptoDriverObjectEcucPartitionRef
  - CryptoDriverObject, [53](#)
- CryptoDriverObjectId
  - CryptoDriverObject, [53](#)
- CryptoDriverObjects
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, [574](#)
- CryptoDriverStateType
  - Asr\_Standard\_Types.h, [408](#)
- CryptoElementActualSize
  - CryptoKeyElementType, [55](#)
- CryptoElementArray
  - CryptoKeyElementType, [55](#)
- CryptoKey, [53](#)
  - KeyType, [54](#)
  - TypeId, [54](#)
- CryptoKeyElementAllowPartialAccess
  - crypto\_key\_element\_type\_info\_st, [44](#)
  - CryptoKeyElementType, [55](#)
- CryptoKeyElementId
  - crypto\_key\_element\_type\_info\_st, [44](#)
  - CryptoKeyElementType, [55](#)
- CryptoKeyElementMaxSize
  - crypto\_key\_element\_type\_info\_st, [44](#)
  - CryptoKeyElementType, [55](#)
- CryptoKeyElementPersist
  - crypto\_key\_element\_type\_info\_st, [44](#)
  - CryptoKeyElementType, [55](#)
- CryptoKeyElementReadAccess
  - crypto\_key\_element\_type\_info\_st, [44](#)
  - CryptoKeyElementType, [56](#)
- CryptoKeyElementType, [54](#)
  - CryptoElementActualSize, [55](#)
  - CryptoElementArray, [55](#)
  - CryptoKeyElementAllowPartialAccess, [55](#)
  - CryptoKeyElementId, [55](#)
  - CryptoKeyElementMaxSize, [55](#)
  - CryptoKeyElementPersist, [55](#)
  - CryptoKeyElementReadAccess, [56](#)
  - CryptoKeyElementWriteAccess, [56](#)
  - CryptoKeyFormat, [56](#)
- CryptoKeyElementWriteAccess
  - crypto\_key\_element\_type\_info\_st, [44](#)
  - CryptoKeyElementType, [56](#)
- CryptoKeyFormat
  - crypto\_key\_element\_type\_info\_st, [45](#)
  - CryptoKeyElementType, [56](#)
- CryptoKeyFormatType
  - Asr\_Standard\_Types.h, [408](#)
- cryptoKeyId
  - Crypto\_JobType, [41](#)
- CryptoKeyType, [56](#)
  - element\_info, [57](#)
  - keyelement\_arr, [57](#)
  - keyelement\_num, [57](#)
- CryptoPrimitive, [57](#)
  - CryptoPrimitiveAlgorithmFamily, [58](#)
  - CryptoPrimitiveAlgorithmMode, [58](#)
  - CryptoPrimitiveAlgorithmSecondaryFamily, [58](#)
  - CryptoPrimitiveService, [58](#)
- CryptoPrimitiveAlgorithmFamily
  - CryptoPrimitive, [58](#)
- CryptoPrimitiveAlgorithmMode
  - CryptoPrimitive, [58](#)
- CryptoPrimitiveAlgorithmSecondaryFamily
  - CryptoPrimitive, [58](#)
- CryptoPrimitiveNum
  - CryptoDriverObject, [53](#)
- CryptoPrimitiveRef
  - CryptoDriverObject, [53](#)
- CryptoPrimitiveService
  - CryptoPrimitive, [58](#)
- CryptoQueueSize



- CryptoDriverObject, [53](#)
- ctx
  - aead\_testvec, [9](#)
  - cipher\_testvec, [20](#)
- ctx
  - cipher\_session\_st, [16](#)
  - ehsm\_image, [162](#)
  - ehsm\_image\_verify\_st, [169](#)
- ctx\_addr
  - ehsm\_image\_upgrade\_cmd, [163](#)
  - ehsm\_image\_verfiy\_cmd, [166](#)
  - soc\_image\_upgrade\_input, [373](#)
- ctx\_block\_mgr
  - cipher\_session\_st, [16](#)
- ctx\_size
  - ehsm\_image, [162](#)
  - ehsm\_image\_upgrade\_cmd, [164](#)
  - ehsm\_image\_verfiy\_cmd, [166](#)
  - ehsm\_image\_verify\_st, [169](#)
  - soc\_image\_upgrade\_input, [373](#)
- CtxAddr
  - HSM\_SecureUpgradeType\_, [335](#)
- CtxSize
  - HSM\_SecureUpgradeType\_, [335](#)
- current\_ticks
  - ehsm\_tick\_value, [280](#)
- curve\_id
  - ecies\_testvec, [60](#)
  - ehsm\_cmd\_hdr\_eccp\_keygen\_st, [73](#)
  - ehsm\_cmd\_hdr\_ecise\_st, [75](#)
- d
  - rsacipher\_testvec, [357](#)
- d\_byte\_size
  - rsacipher\_testvec, [357](#)
- DEFAULT\_RSA\_E\_SIZE
  - ehsm\_Srv\_Key\_Ip.c, [899](#)
- DEFAULT\_RSAKEY\_E\_SIZE
  - ehsm\_Com\_Struct\_Ip.h, [413](#)
- DLIST\_HEAD\_INIT
  - ehsm\_Compt\_List.h, [439](#)
- DLIST\_HEAD
  - ehsm\_Compt\_List.h, [439](#)
- DLIST\_POISON1
  - ehsm\_Compt\_List.h, [439](#)
- DLIST\_POISON2
  - ehsm\_Compt\_List.h, [440](#)
- data
  - ehsm\_sensor\_init\_param\_st, [232](#)
- data\_addr
  - ehsm\_sensor\_resp\_init\_cmd, [232](#)
- data\_ptr
  - ehsm\_cmd\_aead\_ptr\_st, [70](#)
- data\_size
  - ehsm\_sensor\_resp\_init\_cmd, [233](#)
- Debug\_Printf
  - ehsm\_Debug\_Ip.h, [495](#)
- debug\_authentication
  - ehsm\_mailbox\_req, [197](#)
- decrypt
  - ehsm\_key\_usages\_st, [192](#)
- HSM\_KeyActUseFlagsType, [312](#)
- HSM\_KeyUsagesType, [320](#)
- key\_act\_use\_flags\_t, [344](#)
- Delete\_Counter
  - ehsm\_Ip\_Evita\_Ip.h, [646](#)
- derive\_key
  - ehsm\_mailbox\_req, [197](#)
- derive\_key\_elements
  - ehsm\_Ip\_Asr\_KeyCfg\_Ip.c, [574](#)
- derive\_type
  - crypto\_key\_derive\_info, [42](#)
  - ehsm\_derive\_key\_cmd, [111](#)
  - ehsm\_key\_derived\_param, [178](#)
- Dh
  - Hsm\_PriKeyDataType\_, [325](#)
  - Hsm\_PubKeyDataType\_, [327](#)
- dh
  - ehsm\_prikey\_data\_, [216](#)
  - ehsm\_pubkey\_data\_, [219](#)
- dh\_k
  - ehsm\_prikey\_data\_, [216](#)
- dh\_key\_param
  - ehsm\_gen\_key\_param\_st, [145](#)
- dh\_key\_size
  - key\_info\_st, [346](#)
- dh\_mode
  - ehsm\_create\_dh\_key\_cmd, [91](#)
  - ehsm\_create\_dh\_key\_param, [94](#)
- dh\_param
  - ehsm\_dh\_pubkey, [116](#)
  - key\_info\_st, [346](#)
- dh\_pubkey\_bytes\_size
  - ehsm\_Ip\_Evita\_Types\_Ip.h, [709](#)
  - ehsm\_export\_pub\_key\_, [132](#)
- DhKey
  - Hsm\_PriKeyDataType\_, [326](#)
- DhParam
  - Hsm\_DhPubKeyType\_, [298](#)
- dhkey
  - ehsm\_key\_usages\_st, [192](#)
  - HSM\_KeyUsagesType, [320](#)
- digest
  - hash\_testvec, [283](#)
- digest\_error
  - hash\_testvec, [283](#)
- direction
  - cipher\_session\_st, [16](#)
  - ehsm\_cmd\_hdr\_ecise\_st, [75](#)
  - ehsm\_cmd\_hdr\_pke\_st, [76](#)
  - ehsm\_cmd\_hdr\_ske\_st, [81](#)
  - ehsm\_cmd\_hdr\_sm9\_st, [82](#)
  - ehsm\_fast\_cmac\_st, [134](#)
- dlist\_entry
  - ehsm\_Compt\_List.h, [438](#)
- dlist\_for\_each
  - ehsm\_Compt\_List.h, [439](#)
- dlist\_for\_each\_safe
  - ehsm\_Compt\_List.h, [439](#)
- dlist\_head, [58](#)
- next, [59](#)
- prev, [59](#)

- dp
  - ehsm\_rsa crt\_param\_, [222](#)
  - Hsm\_RsaCrtType\_, [330](#)
  - rsacipher\_testvec, [357](#)
- dp\_byte\_size
  - rsacipher\_testvec, [357](#)
- dq
  - ehsm\_rsa crt\_param\_, [222](#)
  - Hsm\_RsaCrtType\_, [331](#)
  - rsacipher\_testvec, [357](#)
- dq\_byte\_size
  - rsacipher\_testvec, [358](#)
- e
  - ehsm\_rsa\_pubkey, [226](#)
  - HSM\_RsaPubKeyType, [332](#)
  - rsacipher\_testvec, [358](#)
- E\_NOT\_OK
  - Asr\_Standard\_Types.h, [402](#)
- E\_OK
  - Asr\_Standard\_Types.h, [402](#)
- e\_bit\_size
  - ehsm\_cmd\_hdr\_rsa\_keygen\_st, [79](#)
- e\_byte\_size
  - rsacipher\_testvec, [358](#)
- e\_size
  - ehsm\_gen\_key\_cmd, [142](#)
- EHSM\_AES\_128
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [826](#)
- EHSM\_AES\_192
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [826](#)
- EHSM\_AES\_256
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [826](#)
- EHSM\_AES\_CTRDRBG
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [826](#)
- EHSM\_ARRAY\_SIZE
  - eHSM\_Types\_Ip.h, [921](#)
- EHSM\_CANCEL\_CERT\_TYPE\_CMD
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [852](#)
- EHSM\_CANCEL\_SINGLE\_CMD
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [853](#)
- EHSM\_CBC\_MAC\_MODE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [827](#)
- EHSM\_CBC\_MODE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [827](#)
- EHSM\_CFB\_MODE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [827](#)
- EHSM\_CMAC\_MODE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [827](#)
- EHSM\_CMD\_AEAD\_CCM
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [827](#)
- EHSM\_CMD\_AEAD\_GCM
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [827](#)
- EHSM\_CMD\_CHANGE\_CONTROL\_FIELD
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [828](#)
- EHSM\_CMD\_CHANGE\_LIFECYCLE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [828](#)
- EHSM\_CMD\_CIPHER\_KEY\_TYPE\_EVITA
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [853](#)
- EHSM\_CMD\_CIPHER\_KEY\_TYPE\_SHE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [853](#)
- EHSM\_CMD\_CLOSE\_DEBUG
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [828](#)
- EHSM\_CMD\_COPY\_EVITA\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [828](#)
- EHSM\_CMD\_CREATE\_COUNTER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [828](#)
- EHSM\_CMD\_CREATE\_DH\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [828](#)
- EHSM\_CMD\_DEBUG\_AUTHENCATION
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [829](#)
- EHSM\_CMD\_DELETE\_COUNTER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [829](#)
- EHSM\_CMD\_DERIVE\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [829](#)
- EHSM\_CMD\_ECCP\_GEN\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [829](#)
- EHSM\_CMD\_ECDSA
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [829](#)
- EHSM\_CMD\_ECIES
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [829](#)
- EHSM\_CMD\_EXPORT\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [830](#)
- EHSM\_CMD\_FW\_ENCRYPT\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [830](#)
- EHSM\_CMD\_FW\_GET\_RANDOM\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [830](#)
- EHSM\_CMD\_GEN\_DH\_KEY\_PAIR
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [830](#)
- EHSM\_CMD\_GET\_CHALLENGE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [830](#)
- EHSM\_CMD\_GET\_PUB\_FROM\_PRIV
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [830](#)
- EHSM\_CMD\_GET\_SHE\_ID
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [831](#)
- EHSM\_CMD\_GET\_SHE\_STATUS
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [831](#)
- EHSM\_CMD\_HASH
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [831](#)
- EHSM\_CMD\_IMAGE\_UPGRADE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [831](#)
- EHSM\_CMD\_IMAGE\_VERIFY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [831](#)
- EHSM\_CMD\_IMPORT\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [831](#)
- EHSM\_CMD\_INCREASE\_COUNTER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [832](#)
- EHSM\_CMD\_KEY\_REMOVE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [832](#)
- EHSM\_CMD\_KEY\_STATUS
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [832](#)
- EHSM\_CMD\_LOW\_POWER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [832](#)
- EHSM\_CMD\_MAC
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [832](#)
- EHSM\_CMD\_MODULE\_STATUS
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [832](#)
- EHSM\_CMD\_PRIORITY\_DEFAULT
  - eHSM\_Srv\_CmdReq\_Ip.h, [886](#)
- EHSM\_CMD\_READ\_COUNTER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [833](#)
- EHSM\_CMD\_READ\_OTP\_DATA

- eHSM\_Mailbox\_Cmdld\_Ip.h, [833](#)
- EHSM\_CMD\_RESET\_FIRMWARE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [833](#)
- EHSM\_CMD\_RNG\_GENERATE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [833](#)
- EHSM\_CMD\_RSA\_CIPHER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [833](#)
- EHSM\_CMD\_RSA\_GEN\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [833](#)
- EHSM\_CMD\_RSA\_SIGN
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [834](#)
- EHSM\_CMD\_SELF\_TEST
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [834](#)
- EHSM\_CMD\_SENSOR\_RESP\_INIT
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [834](#)
- EHSM\_CMD\_SET\_BAUDRATE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [834](#)
- EHSM\_CMD\_SHE\_LOAD\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [834](#)
- EHSM\_CMD\_SHE\_LOAD\_PLAIN\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [834](#)
- EHSM\_CMD\_SHE\_RAM\_KEY\_EXPORT
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [835](#)
- EHSM\_CMD\_SM2\_CIPHER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [835](#)
- EHSM\_CMD\_SM2\_GEN\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [835](#)
- EHSM\_CMD\_SM2\_SIGN
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [835](#)
- EHSM\_CMD\_SOC\_BOOT\_STATUS
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [835](#)
- EHSM\_CMD\_SOC\_IMAGE\_VERIFY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [835](#)
- EHSM\_CMD\_SYM\_CIPHER
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [836](#)
- EHSM\_CMD\_SYM\_GEN\_KEY
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [836](#)
- EHSM\_CMD\_UART\_COMMAND
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [836](#)
- EHSM\_CMD\_WRITE\_OTP\_DATA
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [836](#)
- EHSM\_CODE\_VERIFY\_FALG
  - eHSM\_Com\_Struct\_Ip.h, [413](#)
- EHSM\_CONTEXT\_SIZE
  - eHSM\_If\_Evita\_Types\_Ip.h, [689](#)
- EHSM\_CRT\_MODE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [836](#)
- EHSM\_CRYPT\_V\_SM9\_MAX\_ID\_SIZE
  - eHSM\_If\_Ext\_Sm9\_Ip.c, [738](#)
- EHSM\_CTR\_MODE
  - eHSM\_Mailbox\_Cmdld\_Ip.h, [837](#)
- eHSM\_Com\_Struct\_Ip.h, [408](#)
  - CODE\_VALID\_FLAG, [412](#)
  - crypto\_key\_derive\_type\_e, [425](#)
  - DEFAULT\_RSAKEY\_E\_SIZE, [413](#)
  - EHSM\_CODE\_VERIFY\_FALG, [413](#)
  - EHSM\_EVITA\_AUTH\_VALUE\_MAX\_SIZE, [413](#)
  - EHSM\_FAST\_CMACE\_AES128, [413](#)
  - EHSM\_FAST\_CMACE\_EVITA\_KEY, [413](#)
  - EHSM\_FAST\_CMACE\_GEN, [413](#)
  - EHSM\_FAST\_CMACE\_SHE\_KEY, [414](#)
- EHSM\_FAST\_CMACE\_SM4, [414](#)
- EHSM\_FAST\_CMACE\_VERIFY, [414](#)
- EHSM\_GET\_STATUS\_ERRORS, [414](#)
- EHSM\_GET\_STATUS\_MEM, [414](#)
- EHSM\_GET\_STATUS\_SBB, [414](#)
- EHSM\_GET\_STATUS\_SHE, [415](#)
- EHSM\_SELF\_TEST\_ALL, [415](#)
- EHSM\_SELF\_TEST\_HASH\_MD5, [415](#)
- EHSM\_SELF\_TEST\_HASH\_SHA1, [415](#)
- EHSM\_SELF\_TEST\_HASH\_SHA2, [416](#)
- EHSM\_SELF\_TEST\_HASH\_SHA256, [416](#)
- EHSM\_SELF\_TEST\_HASH\_SHA3, [416](#)
- EHSM\_SELF\_TEST\_HASH\_SM3, [416](#)
- EHSM\_SELF\_TEST\_HASH, [415](#)
- EHSM\_SELF\_TEST\_PKE\_ECC, [416](#)
- EHSM\_SELF\_TEST\_PKE\_RSA, [417](#)
- EHSM\_SELF\_TEST\_PKE\_SM2, [417](#)
- EHSM\_SELF\_TEST\_PKE\_SM9, [417](#)
- EHSM\_SELF\_TEST\_PKE, [416](#)
- EHSM\_SELF\_TEST\_SKE\_AES, [417](#)
- EHSM\_SELF\_TEST\_SKE\_DES, [417](#)
- EHSM\_SELF\_TEST\_SKE\_SM4, [418](#)
- EHSM\_SELF\_TEST\_SKE\_TDES, [418](#)
- EHSM\_SELF\_TEST\_SKE, [417](#)
- EHSM\_SELF\_TEST\_TRNG, [418](#)
- EHSM\_SHE\_M1\_MAX\_SIZE, [418](#)
- EHSM\_SHE\_M2\_MAX\_SIZE, [418](#)
- EHSM\_SHE\_M3\_MAX\_SIZE, [418](#)
- EHSM\_SHE\_M4\_MAX\_SIZE, [419](#)
- EHSM\_SHE\_M5\_MAX\_SIZE, [419](#)
- ehsm\_SM9\_exchg\_key\_role\_e, [431](#)
- ehsm\_api\_type\_e, [426](#)
- ehsm\_challenge\_type\_e, [426](#)
- ehsm\_code\_upgrade\_alg\_e, [426](#)
- ehsm\_code\_verify\_alg\_e, [427](#)
- ehsm\_control\_field\_type\_e, [427](#)
- ehsm\_create\_dh\_key\_param\_st, [421](#)
- ehsm\_create\_evita\_key\_param\_st, [421](#)
- ehsm\_create\_random\_key\_param\_st, [421](#)
- ehsm\_crypto\_randomgenerate\_param\_st, [421](#)
- ehsm\_debug\_auth\_alg\_e, [427](#)
- ehsm\_dh\_mode\_e, [428](#)
- ehsm\_emu\_status\_st, [421](#)
- ehsm\_evita\_key\_export\_st, [422](#)
- ehsm\_evita\_memory\_info\_st, [422](#)
- ehsm\_exchange\_sm9\_key\_param\_st, [422](#)
- ehsm\_fw\_encrypt\_key\_slot\_e, [428](#)
- ehsm\_fw\_encrypt\_key\_st, [422](#)
- ehsm\_fw\_encrypt\_key\_type\_e, [428](#)
- ehsm\_fw\_random\_key\_slot\_e, [429](#)
- ehsm\_fw\_random\_key\_st, [422](#)
- ehsm\_fw\_random\_key\_type\_e, [429](#)
- ehsm\_gen\_sm9\_key\_param\_st, [422](#)
- ehsm\_gen\_sm9\_key\_type\_e, [429](#)
- ehsm\_gen\_sm9\_master\_key\_param\_st, [422](#)
- ehsm\_gen\_sm9\_userpriv\_key\_param\_st, [422](#)
- ehsm\_get\_pub\_from\_priv\_param\_st, [423](#)
- ehsm\_image\_process\_mode\_e, [429](#)
- ehsm\_image\_upgrade\_st, [423](#)
- ehsm\_key\_copy\_param\_st, [423](#)
- ehsm\_key\_derived\_param\_st, [423](#)

- ehsm\_key\_mem\_type\_e, [430](#)
- ehsm\_key\_remove\_param\_st, [423](#)
- ehsm\_key\_status\_param\_st, [423](#)
- ehsm\_keyexchange\_key\_info\_st, [423](#)
- ehsm\_lifecycle\_e, [430](#)
- ehsm\_rsa\_key\_type\_e, [430](#)
- ehsm\_she\_key\_host\_param\_st, [423](#)
- ehsm\_she\_key\_param\_st, [424](#)
- ehsm\_she\_plain\_key\_host\_param\_st, [424](#)
- ehsm\_she\_plain\_key\_param\_st, [424](#)
- ehsm\_sm9\_exckey\_gen\_tmpkey\_st, [424](#)
- ehsm\_sm9\_gen\_mast\_pubkey\_st, [424](#)
- ehsm\_sm9\_gen\_tmp\_pubkey\_st, [424](#)
- ehsm\_sm9\_inexport\_key\_param\_st, [424](#)
- ehsm\_sm9\_master\_key\_type\_e, [431](#)
- ehsm\_sm9\_unwrap\_key\_param\_st, [424](#)
- ehsm\_sm9\_user\_privkey\_type\_e, [431](#)
- ehsm\_sm9\_wrap\_key\_param\_st, [425](#)
- ehsm\_soc\_image\_upgrade\_info\_st, [425](#)
- ehsm\_soc\_image\_upgrade\_input\_st, [425](#)
- ehsm\_soc\_image\_verify\_info\_st, [425](#)
- ehsm\_soc\_image\_verify\_input\_st, [425](#)
- ehsm\_uart\_baudrate\_e, [432](#)
- IMAGE\_ANALYSIS\_CODE, [419](#)
- IMAGE\_DECRYPT\_CODE, [419](#)
- IMAGE\_ENCRYPT\_CODE, [419](#)
- IMAGE\_PUBLIC\_KEY\_MAX\_LENGTH, [419](#)
- IMAGE\_SIGNATURE\_MAX\_LENGTH, [420](#)
- OTP\_CONTROL\_FILED\_BYTE\_SIZE, [420](#)
- SECURE\_BOOT\_TYPE\_IMAGE\_VERIFY, [420](#)
- SECURE\_BOOT\_TYPE\_SECURE\_BOOT, [420](#)
- SOC\_BOOT\_TYPE\_PARALLEL, [420](#)
- SOC\_BOOT\_TYPE\_SEQUENTIAL, [420](#)
- SOC\_CODE\_VERIFY\_FALG, [421](#)
- sm2\_ext\_param\_st, [425](#)
- sm2\_key\_exchange\_role\_e, [432](#)
- UPGRADE\_VALID\_FLAG, [421](#)
- eHSM\_Compt\_Bitmap.c, [432](#)
  - ehsm\_bitmap\_clr, [433](#)
  - ehsm\_bitmap\_count, [433](#)
  - ehsm\_bitmap\_first, [433](#)
  - ehsm\_bitmap\_init, [433](#)
  - ehsm\_bitmap\_last, [434](#)
  - ehsm\_bitmap\_reset, [434](#)
  - ehsm\_bitmap\_set, [434](#)
- eHSM\_Compt\_Bitmap.h, [435](#)
  - bitmap\_st, [435](#)
  - ehsm\_bitmap\_clr, [435](#)
  - ehsm\_bitmap\_count, [436](#)
  - ehsm\_bitmap\_create, [436](#)
  - ehsm\_bitmap\_destroy, [436](#)
  - ehsm\_bitmap\_first, [436](#)
  - ehsm\_bitmap\_init, [437](#)
  - ehsm\_bitmap\_last, [437](#)
  - ehsm\_bitmap\_reset, [437](#)
  - ehsm\_bitmap\_set, [437](#)
- eHSM\_Compt\_List.h, [438](#)
  - container\_of, [438](#)
  - DLIST\_HEAD\_INIT, [439](#)
  - DLIST\_HEAD, [439](#)
  - DLIST\_POISON1, [439](#)
  - DLIST\_POISON2, [440](#)
  - dlist\_entry, [438](#)
  - dlist\_for\_each, [439](#)
  - dlist\_for\_each\_safe, [439](#)
  - OFFSET, [440](#)
- eHSM\_Config\_Ip.h, [440](#)
  - CONFIG\_EHSM\_ARCH\_HOST\_MAILBOX\_POLLING, [446](#)
  - CONFIG\_EHSM\_ARCH\_V\_CMD\_QUEUE\_SIZE, [446](#)
  - CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_HASH\_QUEUE\_SIZE, [447](#)
  - CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_K\_QUEUE\_SIZE, [447](#)
  - CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_PKE\_QUEUE\_SIZE, [447](#)
  - CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_SKE\_QUEUE\_SIZE, [447](#)
  - CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_SYSGR\_QUEUE\_SIZE, [447](#)
  - CONFIG\_EHSM\_ARCH\_V\_CRYPTOBJ\_TRNG\_QUEUE\_SIZE, [447](#)
  - CONFIG\_EHSM\_ARCH\_V\_DEFAULT\_CMD\_TIMEOUT, [448](#)
  - CONFIG\_EHSM\_ARCH\_V\_JTAG\_TIMEOUT, [448](#)
  - CONFIG\_EHSM\_ARCH\_V\_MAILBOX\_TIMEOUT, [448](#)
  - CONFIG\_EHSM\_ARCH\_V\_RSA\_K\_CMD\_TIMEOUT, [448](#)
  - CONFIG\_EHSM\_AUTOSAR, [448](#)
  - CONFIG\_EHSM\_EVITA, [449](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_PARTIAL\_ACCESS, [449](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_PERSIST, [449](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_READ\_ACCESS, [449](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_AEAD\_TAG\_SIZE\_WRITE\_ACCESS, [449](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PARTIAL\_ACCESS, [449](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_PERSIST, [450](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_READ\_ACCESS, [450](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_DATA\_WRITE\_ACCESS, [450](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PARTIAL\_ACCESS, [450](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_PERSIST, [450](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_READ\_ACCESS, [450](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATURE\_WRITE\_ACCESS, [451](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATUREEDDATA\_PARTIAL\_ACCESS, [451](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATUREEDDATA\_PERSIST, [451](#)
  - CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_SIGNATUREEDDATA\_READ\_ACCESS, [451](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_S←  
IGNEDDATA\_WRITE\_ACCESS, 451

CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_S←  
UBJECT\_PUBLIC\_K\_PARTIAL\_ACCESS, 451

CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_S←  
UBJECT\_PUBLIC\_K\_PERSIST, 452

CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_S←  
UBJECT\_PUBLIC\_K\_READ\_ACCESS, 452

CONFIG\_EHSM\_KMGR\_V\_ASR\_CERTIFICATE\_S←  
UBJECT\_PUBLIC\_K\_WRITE\_ACCESS, 452

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKE←  
Y\_PARTIAL\_ACCESS, 452

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKE←  
Y\_PERSIST, 452

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKE←  
Y\_READ\_ACCESS, 452

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_2NDKE←  
Y\_WRITE\_ACCESS, 453

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHE←  
R\_ALG\_PARTIAL\_ACCESS, 453

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHE←  
R\_ALG\_PERSIST, 453

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHE←  
R\_ALG\_READ\_ACCESS, 453

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CIPHE←  
R\_ALG\_WRITE\_ACCESS, 453

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURV←  
E\_ID\_PARTIAL\_ACCESS, 453

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURV←  
E\_ID\_PERSIST, 454

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURV←  
E\_ID\_READ\_ACCESS, 454

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_CURV←  
E\_ID\_WRITE\_ACCESS, 454

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_PA←  
RTIAL\_ACCESS, 454

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_PE←  
RSIST, 454

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_RE←  
AD\_ACCESS, 454

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_IV\_WRI←  
TE\_ACCESS, 455

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_PAR←  
TIAL\_ACCESS, 455

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_PER←  
SIST, 455

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_REA←  
D\_ACCESS, 455

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_K\_WRI←  
TE\_ACCESS, 455

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_A←  
LG\_PARTIAL\_ACCESS, 455

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_A←  
LG\_PERSIST, 456

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_A←  
LG\_READ\_ACCESS, 456

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_KDF\_A←  
LG\_WRITE\_ACCESS, 456

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_A←  
LG\_PARTIAL\_ACCESS, 456

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_A←  
LG\_PERSIST, 456

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_A←  
LG\_READ\_ACCESS, 456

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_A←  
LG\_WRITE\_ACCESS, 457

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_S←  
IZE\_PARTIAL\_ACCESS, 457

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_S←  
IZE\_PERSIST, 457

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_S←  
IZE\_READ\_ACCESS, 457

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_MAC\_S←  
IZE\_WRITE\_ACCESS, 457

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROO←  
F\_PARTIAL\_ACCESS, 457

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROO←  
F\_PERSIST, 458

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROO←  
F\_READ\_ACCESS, 458

CONFIG\_EHSM\_KMGR\_V\_ASR\_CIPHER\_PROO←  
F\_WRITE\_ACCESS, 458

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARE←  
NT\_K\_PARTIAL\_ACCESS, 458

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARE←  
NT\_K\_PERSIST, 458

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARE←  
NT\_K\_READ\_ACCESS, 458

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_PARE←  
NT\_K\_WRITE\_ACCESS, 459

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARG←  
ET\_K\_HANDLE\_PARTIAL\_ACCESS, 459

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARG←  
ET\_K\_HANDLE\_PERSIST, 459

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARG←  
ET\_K\_HANDLE\_READ\_ACCESS, 459

CONFIG\_EHSM\_KMGR\_V\_ASR\_COPY\_K\_TARG←  
ET\_K\_HANDLE\_WRITE\_ACCESS, 459

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BL←  
OB\_PARTIAL\_ACCESS, 459

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BL←  
OB\_PERSIST, 460

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BL←  
OB\_READ\_ACCESS, 460

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_BL←  
OB\_WRITE\_ACCESS, 460

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PA←  
RTIAL\_ACCESS, 460

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_PE←  
RSIST, 460

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_RE←  
AD\_ACCESS, 460

CONFIG\_EHSM\_KMGR\_V\_ASR\_EXPORT\_K\_WR←  
ITE\_ACCESS, 461

CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PAR←  
TIAL\_ACCESS, 461

CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_PER←  
SIST, 461

CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_REA←  
D\_ACCESS, 461



CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORT\_K\_WRITE\_ACCESS, [461](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_PARTIAL\_ACCESS, [461](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_PERSIST, [462](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_READ\_ACCESS, [462](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_IMPORTED\_K\_KHANDLE\_WRITE\_ACCESS, [462](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_PARTIAL\_ACCESS, [462](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_PERSIST, [462](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_READ\_ACCESS, [462](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_EXT\_SHE\_KEY\_WRITE\_ACCESS, [463](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_PARTIAL\_ACCESS, [463](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_PERSIST, [463](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_READ\_ACCESS, [463](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_MATERIAL\_WRITE\_ACCESS, [463](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PARTIAL\_ACCESS, [463](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_PERSIST, [464](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_READ\_ACCESS, [464](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_K\_STATUS\_WRITE\_ACCESS, [464](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_PARTIAL\_ACCESS, [464](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_PERSIST, [464](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_READ\_ACCESS, [464](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ALGORITHM\_WRITE\_ACCESS, [465](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_PARTIAL\_ACCESS, [465](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_PERSIST, [465](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_READ\_ACCESS, [465](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_ITERATIONS\_WRITE\_ACCESS, [465](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PARTIAL\_ACCESS, [465](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_PERSIST, [466](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_READ\_ACCESS, [466](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_K\_WRITE\_ACCESS, [466](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_PARTIAL\_ACCESS, [466](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_PERSIST, [466](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_READ\_ACCESS, [466](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_KHANDLE\_WRITE\_ACCESS, [467](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_PARTIAL\_ACCESS, [467](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_PERSIST, [467](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_READ\_ACCESS, [467](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_PASSWD\_WRITE\_ACCESS, [467](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_PARTIAL\_ACCESS, [467](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_PERSIST, [468](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_READ\_ACCESS, [468](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_SALT\_WRITE\_ACCESS, [468](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_PARTIAL\_ACCESS, [468](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_PERSIST, [468](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_READ\_ACCESS, [468](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KDERIVATION\_TYPE\_WRITE\_ACCESS, [469](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_PARTIAL\_ACCESS, [469](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_PERSIST, [469](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_READ\_ACCESS, [469](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ALGORITHM\_WRITE\_ACCESS, [469](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ASE\_PARTIAL\_ACCESS, [469](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ASE\_PERSIST, [470](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ASE\_READ\_ACCESS, [470](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ASE\_WRITE\_ACCESS, [470](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEY\_EYINFO\_PARTIAL\_ACCESS, [470](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEY\_EYINFO\_PERSIST, [470](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEY\_EYINFO\_READ\_ACCESS, [470](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_KEY\_EYINFO\_WRITE\_ACCESS, [471](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ON\_WNPUBKEY\_PARTIAL\_ACCESS, [471](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ON\_WNPUBKEY\_PERSIST, [471](#)  
CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_ON\_WNPUBKEY\_READ\_ACCESS, [471](#)

CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_O↔  
 WNPUBKEY\_WRITE\_ACCESS, 471  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 EERPUBKEY\_PARTIAL\_ACCESS, 471  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 EERPUBKEY\_PERSIST, 472  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 EERPUBKEY\_READ\_ACCESS, 472  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 EERPUBKEY\_WRITE\_ACCESS, 472  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 RIVKEY\_PARTIAL\_ACCESS, 472  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 RIVKEY\_PERSIST, 472  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 RIVKEY\_READ\_ACCESS, 472  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 RIVKEY\_WRITE\_ACCESS, 473  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 UBKTYPE\_PARTIAL\_ACCESS, 473  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 UBKTYPE\_PERSIST, 473  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 UBKTYPE\_READ\_ACCESS, 473  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_P↔  
 UBKTYPE\_WRITE\_ACCESS, 473  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 HAREDVALUE\_PARTIAL\_ACCESS, 473  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 HAREDVALUE\_PERSIST, 474  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 HAREDVALUE\_READ\_ACCESS, 474  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 HAREDVALUE\_WRITE\_ACCESS, 474  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE↔  
 SM2\_LOCALTMPKINFO\_PARTIAL\_ACCESS,  
 474  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_LOCALTMPKINFO\_PERSIST, 474  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_LOCALTMPKINFO\_READ\_ACCESS, 474  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_LOCALTMPKINFO\_WRITE\_ACCESS, 475  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_PEERTMPPUBK\_PARTIAL\_ACCESS, 475  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_PEERTMPPUBK\_PERSIST, 475  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_PEERTMPPUBK\_READ\_ACCESS, 475  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_PEERTMPPUBK\_WRITE\_ACCESS, 475  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_ROLE\_PARTIAL\_ACCESS, 475  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_ROLE\_PERSIST, 476  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_ROLE\_READ\_ACCESS, 476  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_ROLE\_WRITE\_ACCESS, 476  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_S1\_S2\_VALUE\_PARTIAL\_ACCESS, 476  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_S1\_S2\_VALUE\_PERSIST, 476  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_S1\_S2\_VALUE\_READ\_ACCESS, 476  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_S1\_S2\_VALUE\_WRITE\_ACCESS, 477  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_SA\_SB\_VALUE\_PARTIAL\_ACCESS, 477  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_SA\_SB\_VALUE\_PERSIST, 477  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_SA\_SB\_VALUE\_READ\_ACCESS, 477  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KEXCHANGE\_S↔  
 M2\_SA\_SB\_VALUE\_WRITE\_ACCESS, 477  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_A↔  
 LGORITHM\_PARTIAL\_ACCESS, 477  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_A↔  
 LGORITHM\_PERSIST, 478  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_A↔  
 LGORITHM\_READ\_ACCESS, 478  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_A↔  
 LGORITHM\_WRITE\_ACCESS, 478  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_D↔  
 H\_K\_INFO\_PARTIAL\_ACCESS, 478  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_D↔  
 H\_K\_INFO\_PERSIST, 478  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_D↔  
 H\_K\_INFO\_READ\_ACCESS, 478  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_D↔  
 H\_K\_INFO\_WRITE\_ACCESS, 479  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K↔  
 PARTIAL\_ACCESS, 479  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K↔  
 PERSIST, 479  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K↔  
 READ\_ACCESS, 479  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_K↔  
 WRITE\_ACCESS, 479  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KI↔  
 NFO\_PARTIAL\_ACCESS, 479  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KI↔  
 NFO\_PERSIST, 480  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KI↔  
 NFO\_READ\_ACCESS, 480  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_KI↔  
 NFO\_WRITE\_ACCESS, 480  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_S↔  
 EED\_PARTIAL\_ACCESS, 480  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_S↔  
 EED\_PERSIST, 480  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_S↔  
 EED\_READ\_ACCESS, 480  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_KGENERATE\_S↔  
 EED\_WRITE\_ACCESS, 481  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_PARTIA↔  
 L\_ACCESS, 481  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_PERSIST,  
 481  
 CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_READ↔  
 ACCESS, 481

- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_K\_WRITE\_↔  
ACCESS, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_P\_↔  
ARTIAL\_ACCESS, [481](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_P\_↔  
PERSIST, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_R\_↔  
READ\_ACCESS, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_MAC\_PROOF\_↔  
WRITE\_ACCESS, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PA\_↔  
RTIAL\_ACCESS, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_PE\_↔  
PERSIST, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_RE\_↔  
AD\_ACCESS, [482](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_REMOVE\_K\_WR\_↔  
RITE\_ACCESS, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_↔  
PARTIAL\_ACCESS, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_↔  
PERSIST, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_↔  
READ\_ACCESS, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_K\_↔  
WRITE\_ACCESS, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RS\_↔  
A\_CRT\_MODE\_PARTIAL\_ACCESS, [483](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RS\_↔  
A\_CRT\_MODE\_PERSIST, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RS\_↔  
A\_CRT\_MODE\_READ\_ACCESS, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_RS\_↔  
A\_CRT\_MODE\_WRITE\_ACCESS, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TI\_↔  
MESTAMPED\_PARTIAL\_ACCESS, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TI\_↔  
MESTAMPED\_PERSIST, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TI\_↔  
MESTAMPED\_READ\_ACCESS, [484](#)
- CONFIG\_EHSM\_KMGR\_V\_ASR\_SIGNATURE\_TI\_↔  
MESTAMPED\_WRITE\_ACCESS, [485](#)
- CONFIG\_EHSM\_SHE, [485](#)
- EHSM\_DECRYPTION
  - eHSM\_Mailbox\_CmdId\_Ip.h, [837](#)
- EHSM\_DES
  - eHSM\_Mailbox\_CmdId\_Ip.h, [837](#)
- eHSM\_Debug\_Ip.h, [485](#)
  - AUTOSAR\_LOG\_DEBUG, [486](#)
  - AUTOSAR\_LOG\_ERROR, [486](#)
  - AUTOSAR\_LOG\_INFO, [487](#)
  - AUTOSAR\_LOG\_WARN, [487](#)
  - AUTOSAR\_PURE\_LOG\_DEBUG, [487](#)
  - COMMON\_LOG\_DEBUG, [487](#)
  - COMMON\_LOG\_ERROR, [487](#)
  - COMMON\_LOG\_INFO, [488](#)
  - COMMON\_LOG\_WARN, [488](#)
  - COMMON\_PURE\_LOG\_DEBUG, [488](#)
  - CONFIG\_HOST\_AUTOSAR\_DEBUG\_ENABLE, [488](#)
  - CONFIG\_HOST\_COMMON\_DEBUG\_ENABLE, [488](#)
  - CONFIG\_HOST\_CUSTOM\_DEBUG\_ENABLE, [488](#)
  - CONFIG\_HOST\_EVITA\_DEBUG\_ENABLE, [489](#)
  - CONFIG\_HOST\_PERFORMANCE\_DEBUG\_ENAB\_↔  
LE, [489](#)
  - CONFIG\_HOST\_SHE\_DEBUG\_ENABLE, [489](#)
  - CONFIG\_HOST\_V\_LOG\_DEBUG, [489](#)
  - CONFIG\_HOST\_V\_LOG\_ERR, [489](#)
  - CONFIG\_HOST\_V\_LOG\_INFO, [489](#)
  - CONFIG\_HOST\_V\_LOG\_LEVEL, [490](#)
  - CONFIG\_HOST\_V\_LOG\_WARN, [490](#)
  - CUSTOM\_LOG\_DEBUG, [490](#)
  - CUSTOM\_LOG\_ERROR, [490](#)
  - CUSTOM\_LOG\_INFO, [490](#)
  - CUSTOM\_LOG\_WARN, [490](#)
  - CUSTOM\_PURE\_LOG\_DEBUG, [491](#)
  - Debug\_Printf, [495](#)
  - EVITA\_LOG\_DEBUG, [491](#)
  - EVITA\_LOG\_ERROR, [491](#)
  - EVITA\_LOG\_INFO, [491](#)
  - EVITA\_LOG\_WARN, [491](#)
  - EVITA\_PURE\_LOG\_DEBUG, [492](#)
  - filename, [492](#)
  - HOST\_LOG\_DEBUG, [492](#)
  - HOST\_LOG\_ERROR, [492](#)
  - HOST\_LOG\_INFO, [493](#)
  - HOST\_LOG\_WARN, [493](#)
  - HOST\_LOG, [492](#)
  - HOST\_PURE\_LOG, [493](#)
  - PERFORMANCE\_LOG\_DEBUG, [493](#)
  - PERFORMANCE\_LOG\_ERROR, [493](#)
  - PERFORMANCE\_LOG\_INFO, [493](#)
  - PERFORMANCE\_LOG\_WARN, [494](#)
  - PERFORMANCE\_PURE\_LOG\_DEBUG, [494](#)
  - PURE\_LOG, [494](#)
  - SHE\_LOG\_DEBUG, [494](#)
  - SHE\_LOG\_ERROR, [494](#)
  - SHE\_LOG\_INFO, [495](#)
  - SHE\_LOG\_WARN, [495](#)
  - SHE\_PURE\_LOG\_DEBUG, [495](#)
- eHSM\_Dspt\_CryObj\_Ip.c, [496](#)
  - ehsm\_add\_cmd\_to\_priority\_queue, [496](#)
  - ehsm\_add\_cmd\_to\_sent\_queue, [496](#)
  - ehsm\_crypto\_object\_get\_cmd\_done, [496](#)
  - ehsm\_crypto\_object\_get\_job, [497](#)
  - ehsm\_crypto\_object\_init, [497](#)
  - ehsm\_crypto\_object\_is\_free, [497](#)
  - ehsm\_del\_cmd\_from\_priority\_queue, [497](#)
  - ehsm\_del\_cmd\_from\_sent\_queue, [498](#)
  - ehsm\_fetch\_cmd\_from\_crypto\_object, [498](#)
  - hw\_interrupt\_disable, [498](#)
  - hw\_interrupt\_enable, [498](#)
- eHSM\_Dspt\_CryObj\_Ip.h, [498](#)
  - crypto\_object\_st, [499](#)
  - crypto\_object\_state\_e, [499](#)
  - ehsm\_add\_cmd\_to\_priority\_queue, [500](#)
  - ehsm\_add\_cmd\_to\_sent\_queue, [500](#)
  - ehsm\_crypto\_object\_get\_cmd\_done, [500](#)
  - ehsm\_crypto\_object\_get\_job, [500](#)
  - ehsm\_crypto\_object\_init, [500](#)
  - ehsm\_crypto\_object\_is\_free, [501](#)
  - ehsm\_crypto\_object\_submit\_cmd, [501](#)
  - ehsm\_del\_cmd\_from\_priority\_queue, [501](#)



- ehsm\_del\_cmd\_from\_sent\_queue, [501](#)
- ehsm\_fetch\_cmd\_from\_crypto\_object, [502](#)
- eHSM\_Dspt\_lp.c, [502](#)
  - Crypto\_MainFunction, [503](#)
  - ehsm\_dispatcher\_init, [503](#)
  - ehsm\_mbox\_polling, [503](#)
  - ehsm\_remove\_cmd\_from\_queue, [503](#)
  - ehsm\_submit\_cmd\_req, [503](#)
  - hw\_interrupt\_disable, [504](#)
  - hw\_interrupt\_enable, [504](#)
  - pctest\_time\_counting\_end, [504](#)
  - pctest\_time\_counting\_get\_state, [504](#)
- eHSM\_Dspt\_lp.h, [504](#)
  - Crypto\_MainFunction, [505](#)
  - ehsm\_dispatcher\_init, [505](#)
  - ehsm\_remove\_cmd\_from\_queue, [505](#)
  - ehsm\_submit\_cmd\_req, [505](#)
- EHSM\_ECB\_MODE
  - eHSM\_Mailbox\_CmdId\_lp.h, [837](#)
- EHSM\_ECC\_KEY\_PAIR\_MAX\_SIZE
  - eHSM\_If\_Evita\_Types\_lp.h, [689](#)
- EHSM\_ENCRYPTION
  - eHSM\_Mailbox\_CmdId\_lp.h, [837](#)
- EHSM\_ERR\_ALGORITHM\_ERROR
  - eHSM\_Err\_Code\_lp.h, [511](#)
- EHSM\_ERR\_ALL\_COUNTER\_BUSY
  - eHSM\_Err\_Code\_lp.h, [511](#)
- EHSM\_ERR\_ALL\_KEY\_SPACE\_OCCUPIED
  - eHSM\_Err\_Code\_lp.h, [511](#)
- EHSM\_ERR\_AUTH\_FAILED
  - eHSM\_Err\_Code\_lp.h, [511](#)
- EHSM\_ERR\_CHALLENGE\_EXPIRED
  - eHSM\_Err\_Code\_lp.h, [512](#)
- EHSM\_ERR\_CHALLENGE\_FAILED
  - eHSM\_Err\_Code\_lp.h, [512](#)
- EHSM\_ERR\_CHECK\_TIME\_FAILED
  - eHSM\_Err\_Code\_lp.h, [512](#)
- EHSM\_ERR\_CHECK\_TIME\_STAMP\_ERROR
  - eHSM\_Err\_Code\_lp.h, [512](#)
- EHSM\_ERR\_CMD\_CANCELED
  - eHSM\_Err\_Code\_lp.h, [512](#)
- EHSM\_ERR\_CODE\_MOVE\_ERROR
  - eHSM\_Err\_Code\_lp.h, [513](#)
- EHSM\_ERR\_COUNTER\_AUTH\_FAILED
  - eHSM\_Err\_Code\_lp.h, [513](#)
- EHSM\_ERR\_COUNTER\_NOT\_INIT
  - eHSM\_Err\_Code\_lp.h, [513](#)
- EHSM\_ERR\_COUNTER\_WRONG\_ID
  - eHSM\_Err\_Code\_lp.h, [513](#)
- EHSM\_ERR\_CRYPTOCERT\_PARSE\_FAILED
  - eHSM\_Err\_Code\_lp.h, [513](#)
- EHSM\_ERR\_CRYPTOCERT\_VERIFY\_FAILED
  - eHSM\_Err\_Code\_lp.h, [514](#)
- EHSM\_ERR\_CRYPTOSM9\_WRONG\_ID\_SIZE
  - eHSM\_Err\_Code\_lp.h, [514](#)
- EHSM\_ERR\_CRYPTOSM9\_WRONG\_K2\_SIZE
  - eHSM\_Err\_Code\_lp.h, [514](#)
- EHSM\_ERR\_CTX\_MGR\_BUFFER\_DATA\_VALID
  - eHSM\_Mgr\_Ctx\_lp.h, [880](#)
- EHSM\_ERR\_CTX\_MGR\_DATA\_NOT\_READY
  - eHSM\_Mgr\_Ctx\_lp.h, [880](#)
- EHSM\_ERR\_DATA\_CHECK\_ERROR
  - eHSM\_Err\_Code\_lp.h, [514](#)
- EHSM\_ERR\_DATA\_EMPTY
  - eHSM\_Err\_Code\_lp.h, [514](#)
- EHSM\_ERR\_DEBUG\_AUTH\_FAILED
  - eHSM\_Err\_Code\_lp.h, [515](#)
- EHSM\_ERR\_DRBG\_BUFFER\_NULL
  - eHSM\_Err\_Code\_lp.h, [515](#)
- EHSM\_ERR\_DRBG\_DF\_OVERFLOW
  - eHSM\_Err\_Code\_lp.h, [515](#)
- EHSM\_ERR\_DRBG\_LENGTH\_INVALID
  - eHSM\_Err\_Code\_lp.h, [515](#)
- EHSM\_ERR\_DRBG\_LENGTH\_NOT\_MUL\_8
  - eHSM\_Err\_Code\_lp.h, [515](#)
- EHSM\_ERR\_DRBG\_RESEED\_FAILED
  - eHSM\_Err\_Code\_lp.h, [516](#)
- EHSM\_ERR\_EHSM\_BUSY
  - eHSM\_Err\_Code\_lp.h, [516](#)
- EHSM\_ERR\_EHSM\_LIFECYCLE\_LIMIT
  - eHSM\_Err\_Code\_lp.h, [516](#)
- EHSM\_ERR\_EQUAL\_VERSION\_COUNTER
  - eHSM\_Err\_Code\_lp.h, [516](#)
- EHSM\_ERR\_ERC\_NO\_SECURE\_BOOT
  - eHSM\_Err\_Code\_lp.h, [516](#)
- EHSM\_ERR\_GENERAL\_ERROR
  - eHSM\_Err\_Code\_lp.h, [517](#)
- EHSM\_ERR\_HASH\_BUFFER\_NULL
  - eHSM\_Err\_Code\_lp.h, [517](#)
- EHSM\_ERR\_HASH\_CONFIG\_INVALID
  - eHSM\_Err\_Code\_lp.h, [517](#)
- EHSM\_ERR\_HASH\_INPUT\_INVALID
  - eHSM\_Err\_Code\_lp.h, [517](#)
- EHSM\_ERR\_HASH\_LEN\_OVERFLOW
  - eHSM\_Err\_Code\_lp.h, [517](#)
- EHSM\_ERR\_HASH\_OUTPUT\_ZERO\_ALL
  - eHSM\_Err\_Code\_lp.h, [518](#)
- EHSM\_ERR\_HASH\_WORK\_ERROR
  - eHSM\_Err\_Code\_lp.h, [518](#)
- EHSM\_ERR\_HMAC\_VERIFY\_FAILED
  - eHSM\_Err\_Code\_lp.h, [518](#)
- EHSM\_ERR\_IMAGE\_VERIFY\_FAILED
  - eHSM\_Err\_Code\_lp.h, [518](#)
- EHSM\_ERR\_INVALID\_CODE\_FLAG
  - eHSM\_Err\_Code\_lp.h, [518](#)
- EHSM\_ERR\_INVALID\_COUNTER\_INCREMENTATION
  - eHSM\_Err\_Code\_lp.h, [519](#)
- EHSM\_ERR\_INVALID\_KEY\_FLAG
  - eHSM\_Err\_Code\_lp.h, [519](#)
- EHSM\_ERR\_IPCORE\_DH\_INTEGER\_TOO\_BIG
  - eHSM\_Err\_Code\_lp.h, [519](#)
- EHSM\_ERR\_IPCORE\_DH\_INVALID\_INPUT
  - eHSM\_Err\_Code\_lp.h, [519](#)
- EHSM\_ERR\_IPCORE\_DH\_POINTER\_NULL
  - eHSM\_Err\_Code\_lp.h, [519](#)
- EHSM\_ERR\_IPCORE\_DH\_VALUE\_ONE
  - eHSM\_Err\_Code\_lp.h, [519](#)
- EHSM\_ERR\_IPCORE\_DH\_ZERO\_ALL
  - eHSM\_Err\_Code\_lp.h, [520](#)
- EHSM\_ERR\_IPCORE\_ECDH\_INTEGER\_TOO\_BIG
  - eHSM\_Err\_Code\_lp.h, [520](#)
- EHSM\_ERR\_IPCORE\_ECDH\_INVALID\_INPUT

eHSM\_Err\_Code\_lp.h, [520](#)  
EHSM\_ERR\_IPCORE\_ECDH\_POINTOR\_NULL  
eHSM\_Err\_Code\_lp.h, [520](#)  
EHSM\_ERR\_IPCORE\_ECDH\_ZERO\_ALL  
eHSM\_Err\_Code\_lp.h, [520](#)  
EHSM\_ERR\_IPCORE\_ECDSA\_INTEGER\_TOO\_BIG  
eHSM\_Err\_Code\_lp.h, [520](#)  
EHSM\_ERR\_IPCORE\_ECDSA\_INVALID\_INPUT  
eHSM\_Err\_Code\_lp.h, [521](#)  
EHSM\_ERR\_IPCORE\_ECDSA\_POINTOR\_NULL  
eHSM\_Err\_Code\_lp.h, [521](#)  
EHSM\_ERR\_IPCORE\_ECDSA\_VERIFY\_FAILED  
eHSM\_Err\_Code\_lp.h, [521](#)  
EHSM\_ERR\_IPCORE\_ECDSA\_ZERO\_ALL  
eHSM\_Err\_Code\_lp.h, [521](#)  
EHSM\_ERR\_IPCORE\_ECIES\_ERROR  
eHSM\_Err\_Code\_lp.h, [521](#)  
EHSM\_ERR\_IPCORE\_ECIES\_INTEGER\_TOO\_BIG  
eHSM\_Err\_Code\_lp.h, [521](#)  
EHSM\_ERR\_IPCORE\_ECIES\_INVALID\_INPUT  
eHSM\_Err\_Code\_lp.h, [522](#)  
EHSM\_ERR\_IPCORE\_ECIES\_POINTOR\_NULL  
eHSM\_Err\_Code\_lp.h, [522](#)  
EHSM\_ERR\_IPCORE\_ECIES\_ZERO\_ALL  
eHSM\_Err\_Code\_lp.h, [522](#)  
EHSM\_ERR\_IPCORE\_HASH\_BUFFER\_NULL  
eHSM\_Err\_Code\_lp.h, [522](#)  
EHSM\_ERR\_IPCORE\_HASH\_CONFIG\_INVALID  
eHSM\_Err\_Code\_lp.h, [522](#)  
EHSM\_ERR\_IPCORE\_HASH\_ERROR  
eHSM\_Err\_Code\_lp.h, [522](#)  
EHSM\_ERR\_IPCORE\_HASH\_INPUT\_INVALID  
eHSM\_Err\_Code\_lp.h, [523](#)  
EHSM\_ERR\_IPCORE\_HASH\_LEN\_OVERFLOW  
eHSM\_Err\_Code\_lp.h, [523](#)  
EHSM\_ERR\_IPCORE\_HASH\_OUTPUT\_ZERO\_ALL  
eHSM\_Err\_Code\_lp.h, [523](#)  
EHSM\_ERR\_IPCORE\_PKE\_ERROR  
eHSM\_Err\_Code\_lp.h, [523](#)  
EHSM\_ERR\_IPCORE\_PKE\_FINISHED  
eHSM\_Err\_Code\_lp.h, [523](#)  
EHSM\_ERR\_IPCORE\_PKE\_INFINITY\_POINT  
eHSM\_Err\_Code\_lp.h, [523](#)  
EHSM\_ERR\_IPCORE\_PKE\_INTEGER\_TOO\_BIG  
eHSM\_Err\_Code\_lp.h, [524](#)  
EHSM\_ERR\_IPCORE\_PKE\_INVALID\_INPUT  
eHSM\_Err\_Code\_lp.h, [524](#)  
EHSM\_ERR\_IPCORE\_PKE\_NO\_MODINV  
eHSM\_Err\_Code\_lp.h, [524](#)  
EHSM\_ERR\_IPCORE\_PKE\_NOT\_ON\_CURVE  
eHSM\_Err\_Code\_lp.h, [524](#)  
EHSM\_ERR\_IPCORE\_PKE\_ZERO\_ALL  
eHSM\_Err\_Code\_lp.h, [524](#)  
EHSM\_ERR\_IPCORE\_SKE\_ATTACK\_ALARM  
eHSM\_Err\_Code\_lp.h, [524](#)  
EHSM\_ERR\_IPCORE\_SKE\_BUFFER\_NULL  
eHSM\_Err\_Code\_lp.h, [525](#)  
EHSM\_ERR\_IPCORE\_SKE\_CONFIG\_INVALID  
eHSM\_Err\_Code\_lp.h, [525](#)  
EHSM\_ERR\_IPCORE\_SKE\_ERROR  
eHSM\_Err\_Code\_lp.h, [525](#)  
EHSM\_ERR\_IPCORE\_SKE\_INPUT\_INVALID  
eHSM\_Err\_Code\_lp.h, [525](#)  
EHSM\_ERR\_IPCORE\_SKE\_PADDING\_ERROR  
eHSM\_Err\_Code\_lp.h, [525](#)  
EHSM\_ERR\_IPCORE\_SM2\_BUFFER\_NULL  
eHSM\_Err\_Code\_lp.h, [525](#)  
EHSM\_ERR\_IPCORE\_SM2\_DECRYPT\_VERIFY\_FAILED  
eHSM\_Err\_Code\_lp.h, [526](#)  
EHSM\_ERR\_IPCORE\_SM2\_EXCHANGE\_ROLE\_INVALID  
LID  
eHSM\_Err\_Code\_lp.h, [526](#)  
EHSM\_ERR\_IPCORE\_SM2\_INPUT\_INVALID  
eHSM\_Err\_Code\_lp.h, [526](#)  
EHSM\_ERR\_IPCORE\_SM2\_INTEGER\_TOO\_BIG  
eHSM\_Err\_Code\_lp.h, [526](#)  
EHSM\_ERR\_IPCORE\_SM2\_NOT\_ON\_CURVE  
eHSM\_Err\_Code\_lp.h, [526](#)  
EHSM\_ERR\_IPCORE\_SM2\_VERIFY\_FAILED  
eHSM\_Err\_Code\_lp.h, [526](#)  
EHSM\_ERR\_IPCORE\_SM2\_ZERO\_ALL  
eHSM\_Err\_Code\_lp.h, [527](#)  
EHSM\_ERR\_IPCORE\_TRNG\_BUFFER\_NULL  
eHSM\_Err\_Code\_lp.h, [527](#)  
EHSM\_ERR\_IPCORE\_TRNG\_ERROR  
eHSM\_Err\_Code\_lp.h, [527](#)  
EHSM\_ERR\_IPCORE\_TRNG\_HT\_ERROR  
eHSM\_Err\_Code\_lp.h, [527](#)  
EHSM\_ERR\_IPCORE\_TRNG\_INVALID\_CONFIG  
eHSM\_Err\_Code\_lp.h, [527](#)  
EHSM\_ERR\_IPCORE\_TRNG\_INVALID\_INPUT  
eHSM\_Err\_Code\_lp.h, [527](#)  
EHSM\_ERR\_IPCORE\_TRNG\_TIMEOUT\_ERROR  
eHSM\_Err\_Code\_lp.h, [528](#)  
EHSM\_ERR\_JOB\_CANCELED  
eHSM\_Err\_Code\_lp.h, [528](#)  
EHSM\_ERR\_KEY\_BUFF\_SMALLER  
eHSM\_Err\_Code\_lp.h, [528](#)  
EHSM\_ERR\_KEY\_EMPTY\_ERROR  
eHSM\_Err\_Code\_lp.h, [528](#)  
EHSM\_ERR\_KEY\_INVALID\_ERROR  
eHSM\_Err\_Code\_lp.h, [528](#)  
EHSM\_ERR\_KEY\_NOT\_AVAILABLE\_ERROR  
eHSM\_Err\_Code\_lp.h, [528](#)  
EHSM\_ERR\_KEY\_STORE\_FULL  
eHSM\_Err\_Code\_lp.h, [529](#)  
EHSM\_ERR\_KEY\_UPDATE\_ERROR  
eHSM\_Err\_Code\_lp.h, [529](#)  
EHSM\_ERR\_KMGR\_READ\_ERROR  
eHSM\_Err\_Code\_lp.h, [529](#)  
EHSM\_ERR\_MAILBOX\_SUCCESS  
eHSM\_Err\_Code\_lp.h, [529](#)  
EHSM\_ERR\_MEMORY\_FAILURE  
eHSM\_Err\_Code\_lp.h, [529](#)  
EHSM\_ERR\_MIDDLE\_SW  
eHSM\_Err\_Code\_lp.h, [530](#)  
EHSM\_ERR\_NO\_CHALLENGE\_AVAILABLE  
eHSM\_Err\_Code\_lp.h, [530](#)  
EHSM\_ERR\_NOT\_NIT  
eHSM\_Err\_Code\_lp.h, [530](#)  
EHSM\_ERR\_NOT\_SUPPORT  
eHSM\_Err\_Code\_lp.h, [530](#)

EHSM\_ERR\_OUT\_OF\_MEM  
eHSM\_Err\_Code\_lp.h, [530](#)

EHSM\_ERR\_PARAM\_ERROR  
eHSM\_Err\_Code\_lp.h, [531](#)

EHSM\_ERR\_PKE\_ED25519\_MSG\_FLOW  
eHSM\_Err\_Code\_lp.h, [531](#)

EHSM\_ERR\_PKE\_SIGN\_FAILED  
eHSM\_Err\_Code\_lp.h, [531](#)

EHSM\_ERR\_PKE\_VERIFY\_FAILED  
eHSM\_Err\_Code\_lp.h, [531](#)

EHSM\_ERR\_PKE\_WORK\_ERROR  
eHSM\_Err\_Code\_lp.h, [531](#)

EHSM\_ERR\_PKE\_WRONG\_CURVE\_ID  
eHSM\_Err\_Code\_lp.h, [532](#)

EHSM\_ERR\_PKE\_WRONG\_E\_SIZE  
eHSM\_Err\_Code\_lp.h, [532](#)

EHSM\_ERR\_PKE\_WRONG\_KDF\_ALG  
eHSM\_Err\_Code\_lp.h, [532](#)

EHSM\_ERR\_PKE\_WRONG\_N\_SIZE  
eHSM\_Err\_Code\_lp.h, [532](#)

EHSM\_ERR\_PKE\_WRONG\_RSA\_CRT\_MODE  
eHSM\_Err\_Code\_lp.h, [532](#)

EHSM\_ERR\_QUEUE\_FULL  
eHSM\_Err\_Code\_lp.h, [533](#)

EHSM\_ERR\_REG\_TWICE\_NOT\_MATCH  
eHSM\_Err\_Code\_lp.h, [533](#)

EHSM\_ERR\_REMOVE\_IMPOSSIBLE  
eHSM\_Err\_Code\_lp.h, [533](#)

EHSM\_ERR\_SKE\_IV\_SHOULD\_NOT\_NULL  
eHSM\_Err\_Code\_lp.h, [533](#)

EHSM\_ERR\_SKE\_MAC\_VRY\_FAILED  
eHSM\_Err\_Code\_lp.h, [533](#)

EHSM\_ERR\_SKE\_WORK\_ERROR  
eHSM\_Err\_Code\_lp.h, [534](#)

EHSM\_ERR\_SKE\_WRONG\_AAD\_SIZE  
eHSM\_Err\_Code\_lp.h, [534](#)

EHSM\_ERR\_SKE\_WRONG\_ALG  
eHSM\_Err\_Code\_lp.h, [534](#)

EHSM\_ERR\_SKE\_WRONG\_IV\_SIZE  
eHSM\_Err\_Code\_lp.h, [534](#)

EHSM\_ERR\_SKE\_WRONG\_K\_SIZE  
eHSM\_Err\_Code\_lp.h, [534](#)

EHSM\_ERR\_SKE\_WRONG\_L\_SIZE  
eHSM\_Err\_Code\_lp.h, [535](#)

EHSM\_ERR\_SKE\_WRONG\_MODE  
eHSM\_Err\_Code\_lp.h, [535](#)

EHSM\_ERR\_SKE\_WRONG\_PADIDNG  
eHSM\_Err\_Code\_lp.h, [535](#)

EHSM\_ERR\_SKE\_WRONG\_TAG\_SIZE  
eHSM\_Err\_Code\_lp.h, [535](#)

EHSM\_ERR\_SMALL\_BUFFER  
eHSM\_Err\_Code\_lp.h, [535](#)

EHSM\_ERR\_SW\_SUCCESS  
eHSM\_Err\_Code\_lp.h, [536](#)

EHSM\_ERR\_TIME\_CHALLENGE\_EXPIRED  
eHSM\_Err\_Code\_lp.h, [536](#)

EHSM\_ERR\_TIME\_STAMP\_EXPIRED  
eHSM\_Err\_Code\_lp.h, [536](#)

EHSM\_ERR\_TIME\_STAMP\_VERIFY\_FAILED  
eHSM\_Err\_Code\_lp.h, [536](#)

EHSM\_ERR\_TRANSPORT\_IMPOSSIBLE  
eHSM\_Err\_Code\_lp.h, [536](#)

EHSM\_ERR\_TRNG\_BUFFER\_NULL  
eHSM\_Err\_Code\_lp.h, [537](#)

EHSM\_ERR\_TRNG\_HT\_ERROR  
eHSM\_Err\_Code\_lp.h, [537](#)

EHSM\_ERR\_TRNG\_INVALID\_CONFIG  
eHSM\_Err\_Code\_lp.h, [537](#)

EHSM\_ERR\_TRNG\_INVALID\_INPUT  
eHSM\_Err\_Code\_lp.h, [537](#)

EHSM\_ERR\_TRNG\_TIMEOUT\_ERROR  
eHSM\_Err\_Code\_lp.h, [537](#)

EHSM\_ERR\_TRNG\_WORK\_ERROR  
eHSM\_Err\_Code\_lp.h, [538](#)

EHSM\_ERR\_UTC\_SYNCHRONIZATION\_FAILED  
eHSM\_Err\_Code\_lp.h, [538](#)

EHSM\_ERR\_UTC\_TIMER\_INVALID\_INDEX  
eHSM\_Err\_Code\_lp.h, [538](#)

EHSM\_ERR\_UTC\_TIMER\_NOT\_SYNC  
eHSM\_Err\_Code\_lp.h, [538](#)

EHSM\_ERR\_WRITE\_PROTECTED  
eHSM\_Err\_Code\_lp.h, [538](#)

EHSM\_ERR\_WRONG\_ALGORITHM  
eHSM\_Err\_Code\_lp.h, [539](#)

EHSM\_ERR\_WRONG\_AUTHORIZATION  
eHSM\_Err\_Code\_lp.h, [539](#)

EHSM\_ERR\_WRONG\_CERT\_KEY\_HANDLE  
eHSM\_Err\_Code\_lp.h, [539](#)

EHSM\_ERR\_WRONG\_CHALLENGE\_TYPE  
eHSM\_Err\_Code\_lp.h, [539](#)

EHSM\_ERR\_WRONG\_CONTEXT  
eHSM\_Err\_Code\_lp.h, [539](#)

EHSM\_ERR\_WRONG\_DATA\_LENGTH  
eHSM\_Err\_Code\_lp.h, [539](#)

EHSM\_ERR\_WRONG\_DIRECT  
eHSM\_Err\_Code\_lp.h, [540](#)

EHSM\_ERR\_WRONG\_EHSM\_ADDR  
eHSM\_Err\_Code\_lp.h, [540](#)

EHSM\_ERR\_WRONG\_JOB\_ID  
eHSM\_Err\_Code\_lp.h, [540](#)

EHSM\_ERR\_WRONG\_KEY\_COMBINATION  
eHSM\_Err\_Code\_lp.h, [540](#)

EHSM\_ERR\_WRONG\_KEY\_DERIVE\_FUNC  
eHSM\_Err\_Code\_lp.h, [540](#)

EHSM\_ERR\_WRONG\_KEY\_HANDLE  
eHSM\_Err\_Code\_lp.h, [541](#)

EHSM\_ERR\_WRONG\_KEY\_LEVEL  
eHSM\_Err\_Code\_lp.h, [541](#)

EHSM\_ERR\_WRONG\_KEY\_LIFE\_LIMIT  
eHSM\_Err\_Code\_lp.h, [541](#)

EHSM\_ERR\_WRONG\_KEY\_SIZE  
eHSM\_Err\_Code\_lp.h, [541](#)

EHSM\_ERR\_WRONG\_KEY\_TYPE  
eHSM\_Err\_Code\_lp.h, [541](#)

EHSM\_ERR\_WRONG\_KEY\_USAGE  
eHSM\_Err\_Code\_lp.h, [541](#)

EHSM\_ERR\_WRONG\_MODULE\_TYPE  
eHSM\_Err\_Code\_lp.h, [542](#)

EHSM\_ERR\_WRONG\_PADDING\_TYPE  
eHSM\_Err\_Code\_lp.h, [542](#)

EHSM\_ERR\_WRONG\_PROC\_MODE  
eHSM\_Err\_Code\_lp.h, [542](#)

- EHSM\_ERR\_WRONG\_PUB\_KEY
  - eHSM\_Err\_Code\_Ip.h, [542](#)
- EHSM\_ERR\_WRONG\_REMOTE\_KEY\_HANDLE
  - eHSM\_Err\_Code\_Ip.h, [542](#)
- EHSM\_ERR\_WRONG\_SALT\_SIZE
  - eHSM\_Err\_Code\_Ip.h, [543](#)
- EHSM\_ERR\_WRONG\_UTC\_TIME
  - eHSM\_Err\_Code\_Ip.h, [543](#)
- EHSM\_ERR\_WRONG\_VERSION\_COUNTER
  - eHSM\_Err\_Code\_Ip.h, [543](#)
- EHSM\_EVITA\_AUTH\_VALUE\_MAX\_SIZE
  - eHSM\_Com\_Struct\_Ip.h, [413](#)
- EHSM\_EVITA\_KEY\_NUM
  - eHSM\_If\_Evita\_Types\_Ip.h, [689](#)
- eHSM\_Err\_Code\_Ip.h, [506](#)
  - EHSM\_ERR\_ALGORITHM\_ERROR, [511](#)
  - EHSM\_ERR\_ALL\_COUNTER\_BUSY, [511](#)
  - EHSM\_ERR\_ALL\_KEY\_SPACE\_OCCUPIED, [511](#)
  - EHSM\_ERR\_AUTH\_FAILED, [511](#)
  - EHSM\_ERR\_CHALLENGE\_EXPIRED, [512](#)
  - EHSM\_ERR\_CHALLENGE\_FAILED, [512](#)
  - EHSM\_ERR\_CHECK\_TIME\_FAILED, [512](#)
  - EHSM\_ERR\_CHECK\_TIME\_STAMP\_ERROR, [512](#)
  - EHSM\_ERR\_CMD\_CANCELED, [512](#)
  - EHSM\_ERR\_CODE\_MOVE\_ERROR, [513](#)
  - EHSM\_ERR\_COUNTER\_AUTH\_FAILED, [513](#)
  - EHSM\_ERR\_COUNTER\_NOT\_INIT, [513](#)
  - EHSM\_ERR\_COUNTER\_WRONG\_ID, [513](#)
  - EHSM\_ERR\_CRYPTO\_CERT\_PARSE\_FAILED, [513](#)
  - EHSM\_ERR\_CRYPTO\_CERT\_VERIFY\_FAILED, [514](#)
  - EHSM\_ERR\_CRYPTO\_SM9\_WRONG\_ID\_SIZE, [514](#)
  - EHSM\_ERR\_CRYPTO\_SM9\_WRONG\_K2\_SIZE, [514](#)
  - EHSM\_ERR\_DATA\_CHECK\_ERROR, [514](#)
  - EHSM\_ERR\_DATA\_EMPTY, [514](#)
  - EHSM\_ERR\_DEBUG\_AUTH\_FAILED, [515](#)
  - EHSM\_ERR\_DRBG\_BUFFER\_NULL, [515](#)
  - EHSM\_ERR\_DRBG\_DF\_OVERFLOW, [515](#)
  - EHSM\_ERR\_DRBG\_LENGTH\_INVALID, [515](#)
  - EHSM\_ERR\_DRBG\_LENGTH\_NOT\_MUL\_8, [515](#)
  - EHSM\_ERR\_DRBG\_RESEED\_FAILED, [516](#)
  - EHSM\_ERR\_EHSM\_BUSY, [516](#)
  - EHSM\_ERR\_EHSM\_LIFECYCLE\_LIMIT, [516](#)
  - EHSM\_ERR\_EQUAL\_VERSION\_COUNTER, [516](#)
  - EHSM\_ERR\_ERC\_NO\_SECURE\_BOOT, [516](#)
  - EHSM\_ERR\_GENERAL\_ERROR, [517](#)
  - EHSM\_ERR\_HASH\_BUFFER\_NULL, [517](#)
  - EHSM\_ERR\_HASH\_CONFIG\_INVALID, [517](#)
  - EHSM\_ERR\_HASH\_INPUT\_INVALID, [517](#)
  - EHSM\_ERR\_HASH\_LEN\_OVERFLOW, [517](#)
  - EHSM\_ERR\_HASH\_OUTPUT\_ZERO\_ALL, [518](#)
  - EHSM\_ERR\_HASH\_WORK\_ERROR, [518](#)
  - EHSM\_ERR\_HMAC\_VERIFY\_FAILED, [518](#)
  - EHSM\_ERR\_IMAGE\_VERIFY\_FAILED, [518](#)
  - EHSM\_ERR\_INVALID\_CODE\_FLAG, [518](#)
  - EHSM\_ERR\_INVALID\_COUNTER\_INCREMENTATION, [519](#)
  - EHSM\_ERR\_INVALID\_KEY\_FLAG, [519](#)
  - EHSM\_ERR\_IPCORE\_DH\_INTEGER\_TOO\_BIG, [519](#)
  - EHSM\_ERR\_IPCORE\_DH\_INVALID\_INPUT, [519](#)
  - EHSM\_ERR\_IPCORE\_DH\_POINTER\_NULL, [519](#)
  - EHSM\_ERR\_IPCORE\_DH\_VALUE\_ONE, [519](#)
  - EHSM\_ERR\_IPCORE\_DH\_ZERO\_ALL, [520](#)
  - EHSM\_ERR\_IPCORE\_ECDH\_INTEGER\_TOO\_BIG, [520](#)
  - EHSM\_ERR\_IPCORE\_ECDH\_INVALID\_INPUT, [520](#)
  - EHSM\_ERR\_IPCORE\_ECDH\_POINTOR\_NULL, [520](#)
  - EHSM\_ERR\_IPCORE\_ECDH\_ZERO\_ALL, [520](#)
  - EHSM\_ERR\_IPCORE\_ECDSA\_INTEGER\_TOO\_BIG, [520](#)
  - EHSM\_ERR\_IPCORE\_ECDSA\_INVALID\_INPUT, [521](#)
  - EHSM\_ERR\_IPCORE\_ECDSA\_POINTOR\_NULL, [521](#)
  - EHSM\_ERR\_IPCORE\_ECDSA\_VERIFY\_FAILED, [521](#)
  - EHSM\_ERR\_IPCORE\_ECDSA\_ZERO\_ALL, [521](#)
  - EHSM\_ERR\_IPCORE\_ECIES\_ERROR, [521](#)
  - EHSM\_ERR\_IPCORE\_ECIES\_INTEGER\_TOO\_BIG, [521](#)
  - EHSM\_ERR\_IPCORE\_ECIES\_INVALID\_INPUT, [522](#)
  - EHSM\_ERR\_IPCORE\_ECIES\_POINTOR\_NULL, [522](#)
  - EHSM\_ERR\_IPCORE\_ECIES\_ZERO\_ALL, [522](#)
  - EHSM\_ERR\_IPCORE\_HASH\_BUFFER\_NULL, [522](#)
  - EHSM\_ERR\_IPCORE\_HASH\_CONFIG\_INVALID, [522](#)
  - EHSM\_ERR\_IPCORE\_HASH\_ERROR, [522](#)
  - EHSM\_ERR\_IPCORE\_HASH\_INPUT\_INVALID, [523](#)
  - EHSM\_ERR\_IPCORE\_HASH\_LEN\_OVERFLOW, [523](#)
  - EHSM\_ERR\_IPCORE\_HASH\_OUTPUT\_ZERO\_ALL, [523](#)
  - EHSM\_ERR\_IPCORE\_PKE\_ERROR, [523](#)
  - EHSM\_ERR\_IPCORE\_PKE\_FINISHED, [523](#)
  - EHSM\_ERR\_IPCORE\_PKE\_INFINITY\_POINT, [523](#)
  - EHSM\_ERR\_IPCORE\_PKE\_INTEGER\_TOO\_BIG, [524](#)
  - EHSM\_ERR\_IPCORE\_PKE\_INVALID\_INPUT, [524](#)
  - EHSM\_ERR\_IPCORE\_PKE\_NO\_MODINV, [524](#)
  - EHSM\_ERR\_IPCORE\_PKE\_NOT\_ON\_CURVE, [524](#)
  - EHSM\_ERR\_IPCORE\_PKE\_ZERO\_ALL, [524](#)
  - EHSM\_ERR\_IPCORE\_SKE\_ATTACK\_ALARM, [524](#)
  - EHSM\_ERR\_IPCORE\_SKE\_BUFFER\_NULL, [525](#)
  - EHSM\_ERR\_IPCORE\_SKE\_CONFIG\_INVALID, [525](#)
  - EHSM\_ERR\_IPCORE\_SKE\_ERROR, [525](#)
  - EHSM\_ERR\_IPCORE\_SKE\_INPUT\_INVALID, [525](#)
  - EHSM\_ERR\_IPCORE\_SKE\_PADDING\_ERROR, [525](#)
  - EHSM\_ERR\_IPCORE\_SM2\_BUFFER\_NULL, [525](#)
  - EHSM\_ERR\_IPCORE\_SM2\_DECRYPT\_VERIFY\_FAILED, [526](#)
  - EHSM\_ERR\_IPCORE\_SM2\_EXCHANGE\_ROLE\_INVALID, [526](#)
  - EHSM\_ERR\_IPCORE\_SM2\_INPUT\_INVALID, [526](#)
  - EHSM\_ERR\_IPCORE\_SM2\_INTEGER\_TOO\_BIG, [526](#)
  - EHSM\_ERR\_IPCORE\_SM2\_NOT\_ON\_CURVE, [526](#)
  - EHSM\_ERR\_IPCORE\_SM2\_VERIFY\_FAILED, [526](#)
  - EHSM\_ERR\_IPCORE\_SM2\_ZERO\_ALL, [527](#)
  - EHSM\_ERR\_IPCORE\_TRNG\_BUFFER\_NULL, [527](#)
  - EHSM\_ERR\_IPCORE\_TRNG\_ERROR, [527](#)
  - EHSM\_ERR\_IPCORE\_TRNG\_HT\_ERROR, [527](#)

- EHSM\_ERR\_IPCORE\_TRNG\_INVALID\_CONFIG, [527](#)
- EHSM\_ERR\_IPCORE\_TRNG\_INVALID\_INPUT, [527](#)
- EHSM\_ERR\_IPCORE\_TRNG\_TIMEOUT\_ERROR, [528](#)
- EHSM\_ERR\_JOB\_CANCELED, [528](#)
- EHSM\_ERR\_KEY\_BUFF\_SMALLER, [528](#)
- EHSM\_ERR\_KEY\_EMPTY\_ERROR, [528](#)
- EHSM\_ERR\_KEY\_INVALID\_ERROR, [528](#)
- EHSM\_ERR\_KEY\_NOT\_AVAILABLE\_ERROR, [528](#)
- EHSM\_ERR\_KEY\_STORE\_FULL, [529](#)
- EHSM\_ERR\_KEY\_UPDATE\_ERROR, [529](#)
- EHSM\_ERR\_KMGR\_READ\_ERROR, [529](#)
- EHSM\_ERR\_MAILBOX\_SUCCESS, [529](#)
- EHSM\_ERR\_MEMORY\_FAILURE, [529](#)
- EHSM\_ERR\_MIDDLE\_SW, [530](#)
- EHSM\_ERR\_NO\_CHALLENGE\_AVAILABLE, [530](#)
- EHSM\_ERR\_NOT\_NIT, [530](#)
- EHSM\_ERR\_NOT\_SUPPORT, [530](#)
- EHSM\_ERR\_OUT\_OF\_MEM, [530](#)
- EHSM\_ERR\_PARAM\_ERROR, [531](#)
- EHSM\_ERR\_PKE\_ED25519\_MSG\_FLOW, [531](#)
- EHSM\_ERR\_PKE\_SIGN\_FAILED, [531](#)
- EHSM\_ERR\_PKE\_VERIFY\_FAILED, [531](#)
- EHSM\_ERR\_PKE\_WORK\_ERROR, [531](#)
- EHSM\_ERR\_PKE\_WRONG\_CURVE\_ID, [532](#)
- EHSM\_ERR\_PKE\_WRONG\_E\_SIZE, [532](#)
- EHSM\_ERR\_PKE\_WRONG\_KDF\_ALG, [532](#)
- EHSM\_ERR\_PKE\_WRONG\_N\_SIZE, [532](#)
- EHSM\_ERR\_PKE\_WRONG\_RSA\_CRT\_MODE, [532](#)
- EHSM\_ERR\_QUEUE\_FULL, [533](#)
- EHSM\_ERR\_REG\_TWICE\_NOT\_MATCH, [533](#)
- EHSM\_ERR\_REMOVE\_IMPOSSIBLE, [533](#)
- EHSM\_ERR\_SKE\_IV\_SHOULD\_NOT\_NULL, [533](#)
- EHSM\_ERR\_SKE\_MAC\_VRY\_FAILED, [533](#)
- EHSM\_ERR\_SKE\_WORK\_ERROR, [534](#)
- EHSM\_ERR\_SKE\_WRONG\_AAD\_SIZE, [534](#)
- EHSM\_ERR\_SKE\_WRONG\_ALG, [534](#)
- EHSM\_ERR\_SKE\_WRONG\_IV\_SIZE, [534](#)
- EHSM\_ERR\_SKE\_WRONG\_K\_SIZE, [534](#)
- EHSM\_ERR\_SKE\_WRONG\_L\_SIZE, [535](#)
- EHSM\_ERR\_SKE\_WRONG\_MODE, [535](#)
- EHSM\_ERR\_SKE\_WRONG\_PADIDNG, [535](#)
- EHSM\_ERR\_SKE\_WRONG\_TAG\_SIZE, [535](#)
- EHSM\_ERR\_SMALL\_BUFFER, [535](#)
- EHSM\_ERR\_SW\_SUCCESS, [536](#)
- EHSM\_ERR\_TIME\_CHALLENGE\_EXPIRED, [536](#)
- EHSM\_ERR\_TIME\_STAMP\_EXPIRED, [536](#)
- EHSM\_ERR\_TIME\_STAMP\_VERIFY\_FAILED, [536](#)
- EHSM\_ERR\_TRANSPORT\_IMPOSSIBLE, [536](#)
- EHSM\_ERR\_TRNG\_BUFFER\_NULL, [537](#)
- EHSM\_ERR\_TRNG\_HT\_ERROR, [537](#)
- EHSM\_ERR\_TRNG\_INVALID\_CONFIG, [537](#)
- EHSM\_ERR\_TRNG\_INVALID\_INPUT, [537](#)
- EHSM\_ERR\_TRNG\_TIMEOUT\_ERROR, [537](#)
- EHSM\_ERR\_TRNG\_WORK\_ERROR, [538](#)
- EHSM\_ERR\_UTC\_SYNCHRONIZATION\_FAILED, [538](#)
- EHSM\_ERR\_UTC\_TIMER\_INVALID\_INDEX, [538](#)
- EHSM\_ERR\_UTC\_TIMER\_NOT\_SYNC, [538](#)
- EHSM\_ERR\_WRITE\_PROTECTED, [538](#)
- EHSM\_ERR\_WRONG\_ALGORITHM, [539](#)
- EHSM\_ERR\_WRONG\_AUTHORIZATION, [539](#)
- EHSM\_ERR\_WRONG\_CERT\_KEY\_HANDLE, [539](#)
- EHSM\_ERR\_WRONG\_CHALLENGE\_TYPE, [539](#)
- EHSM\_ERR\_WRONG\_CONTEXT, [539](#)
- EHSM\_ERR\_WRONG\_DATA\_LENGTH, [539](#)
- EHSM\_ERR\_WRONG\_DIRECT, [540](#)
- EHSM\_ERR\_WRONG\_EHSM\_ADDR, [540](#)
- EHSM\_ERR\_WRONG\_JOB\_ID, [540](#)
- EHSM\_ERR\_WRONG\_KEY\_COMBINATION, [540](#)
- EHSM\_ERR\_WRONG\_KEY\_DERIVE\_FUNC, [540](#)
- EHSM\_ERR\_WRONG\_KEY\_HANDLE, [541](#)
- EHSM\_ERR\_WRONG\_KEY\_LEVEL, [541](#)
- EHSM\_ERR\_WRONG\_KEY\_LIFE\_LIMIT, [541](#)
- EHSM\_ERR\_WRONG\_KEY\_SIZE, [541](#)
- EHSM\_ERR\_WRONG\_KEY\_TYPE, [541](#)
- EHSM\_ERR\_WRONG\_KEY\_USAGE, [541](#)
- EHSM\_ERR\_WRONG\_MODULE\_TYPE, [542](#)
- EHSM\_ERR\_WRONG\_PADDING\_TYPE, [542](#)
- EHSM\_ERR\_WRONG\_PROC\_MODE, [542](#)
- EHSM\_ERR\_WRONG\_PUB\_KEY, [542](#)
- EHSM\_ERR\_WRONG\_REMOTE\_KEY\_HANDLE, [542](#)
- EHSM\_ERR\_WRONG\_SALT\_SIZE, [543](#)
- EHSM\_ERR\_WRONG\_UTC\_TIME, [543](#)
- EHSM\_ERR\_WRONG\_VERSION\_COUNTER, [543](#)
- eHSM\_Exclusive\_Area.h, [543](#)
- Exclusive\_area\_enter, [543](#)
- Exclusive\_area\_exit, [544](#)
- EHSM\_FAST\_CMACE\_AES128
- eHSM\_Com\_Struct\_lp.h, [413](#)
- EHSM\_FAST\_CMACE\_EVITA\_KEY
- eHSM\_Com\_Struct\_lp.h, [413](#)
- EHSM\_FAST\_CMACE\_GEN
- eHSM\_Com\_Struct\_lp.h, [413](#)
- EHSM\_FAST\_CMACE\_SHE\_KEY
- eHSM\_Com\_Struct\_lp.h, [414](#)
- EHSM\_FAST\_CMACE\_SM4
- eHSM\_Com\_Struct\_lp.h, [414](#)
- EHSM\_FAST\_CMACE\_VERIFY
- eHSM\_Com\_Struct\_lp.h, [414](#)
- EHSM\_FINISH
- eHSM\_Mailbox\_Cmdld\_lp.h, [837](#)
- EHSM\_GET\_STATUS\_ERRORS
- eHSM\_Com\_Struct\_lp.h, [414](#)
- EHSM\_GET\_STATUS\_MEM
- eHSM\_Com\_Struct\_lp.h, [414](#)
- EHSM\_GET\_STATUS\_SBB
- eHSM\_Com\_Struct\_lp.h, [414](#)
- EHSM\_GET\_STATUS\_SHE
- eHSM\_Com\_Struct\_lp.h, [415](#)
- EHSM\_GMAC\_MODE
- eHSM\_Mailbox\_Cmdld\_lp.h, [838](#)
- EHSM\_IMAGE\_VERIFY\_TYPE\_FW\_UPGRADE
- eHSM\_If\_Ext\_SysMgr\_lp.c, [740](#)
- EHSM\_IMAGE\_VERIFY\_TYPE\_SOC\_BOOT
- eHSM\_If\_Ext\_SysMgr\_lp.c, [740](#)
- EHSM\_IMAGE\_VERIFY\_TYPE\_SOC\_UPGRADE
- eHSM\_If\_Ext\_SysMgr\_lp.c, [740](#)
- EHSM\_INVALID\_ALG
- eHSM\_Mailbox\_Cmdld\_lp.h, [838](#)



- EHSM\_INVALID\_DIR
  - eHSM\_Mailbox\_CmdId\_lp.h, [838](#)
- eHSM\_If\_Asr\_Cipher\_lp.c, [544](#)
  - ehsm\_aead\_request, [545](#)
  - ehsm\_encrypt\_request, [545](#)
  - ehsm\_hash\_hmac, [546](#)
  - ehsm\_mac\_request, [546](#)
  - ehsm\_random\_generate, [547](#)
  - ehsm\_signature\_gen\_vry, [547](#)
  - IV\_BUFF\_SIZE, [545](#)
- eHSM\_If\_Asr\_Cipher\_lp.h, [548](#)
  - ehsm\_aead\_request, [548](#)
  - ehsm\_encrypt\_request, [548](#)
  - ehsm\_hash\_hmac, [549](#)
  - ehsm\_mac\_request, [549](#)
  - ehsm\_random\_generate, [550](#)
  - ehsm\_signature\_gen\_vry, [550](#)
- eHSM\_If\_Asr\_ErrCode\_lp.c, [551](#)
  - ehsm\_autosar\_convert\_ret\_code, [551](#)
- eHSM\_If\_Asr\_ErrCode\_lp.h, [552](#)
  - ehsm\_autosar\_convert\_ret\_code, [552](#)
- eHSM\_If\_Asr\_lp.h, [552](#)
  - ehsm\_certificate\_parse, [553](#)
  - ehsm\_certificate\_verify, [554](#)
  - ehsm\_get\_version, [555](#)
  - ehsm\_init, [555](#)
  - ehsm\_job\_cancel, [556](#)
  - ehsm\_job\_submit, [556](#)
  - ehsm\_key\_copy, [557](#)
  - ehsm\_key\_derive, [558](#)
  - ehsm\_key\_element\_copy, [558](#)
  - ehsm\_key\_element\_copy\_partial, [559](#)
  - ehsm\_key\_element\_get, [560](#)
  - ehsm\_key\_element\_ids\_get, [561](#)
  - ehsm\_key\_element\_set, [562](#)
  - ehsm\_key\_exchange\_calcpubval, [562](#)
  - ehsm\_key\_exchange\_calcsecret, [563](#)
  - ehsm\_key\_generate, [564](#)
  - ehsm\_key\_set\_valid, [564](#)
  - ehsm\_remove\_key\_extend, [565](#)
- eHSM\_If\_Asr\_Job\_lp.c, [566](#)
  - ehsm\_init, [566](#)
  - ehsm\_job\_cancel, [566](#)
  - ehsm\_job\_submit, [567](#)
- eHSM\_If\_Asr\_Job\_lp.h, [568](#)
  - Crypto\_GetJobKey, [568](#)
  - Crypto\_GetKeyId, [568](#)
- eHSM\_If\_Asr\_Key\_lp.c, [569](#)
- eHSM\_If\_Asr\_Key\_lp.h, [569](#)
  - ehsm\_calcpubval\_with\_job, [569](#)
  - ehsm\_calcsecret\_with\_job, [569](#)
  - ehsm\_get\_ehsm\_key\_type, [570](#)
  - ehsm\_key\_element\_type\_get\_ex, [570](#)
  - ehsm\_key\_is\_valid, [570](#)
  - ehsm\_key\_mgr\_init, [570](#)
  - ehsm\_keyderi\_with\_job, [570](#)
  - ehsm\_keyexport\_with\_job, [570](#)
  - ehsm\_keygen\_with\_job, [571](#)
  - ehsm\_keyimport\_with\_job, [571](#)
  - ehsm\_keyremove\_with\_job, [571](#)
  - ehsm\_keysetvalid\_with\_job, [571](#)
- eHSM\_If\_Asr\_KeyCfg\_lp.c, [571](#)
  - certificate\_key\_elements, [573](#)
  - cipher\_key\_elements, [573](#)
  - copy\_key\_elements, [573](#)
  - crypto\_keys, [573](#)
  - CryptoDriverObjects, [574](#)
  - derive\_key\_elements, [574](#)
  - ehsm\_get\_crypto\_driver\_object, [572](#)
  - ehsm\_get\_crypto\_ke\_info, [572](#)
  - ehsm\_printf\_ke\_size, [572](#)
  - exchange\_key\_elements, [574](#)
  - export\_key\_elements, [574](#)
  - generate\_key\_elements, [574](#)
  - Hash\_CryptoPrimitives, [575](#)
  - import\_key\_elements, [575](#)
  - Key\_CryptoPrimitives, [575](#)
  - key\_remove\_elements, [575](#)
  - key\_status\_elements, [576](#)
  - mac\_key\_elements, [576](#)
  - Pke\_CryptoPrimitives, [577](#)
  - she\_key\_elements, [577](#)
  - she\_plain\_key\_elements, [577](#)
  - signature\_key\_elements, [578](#)
  - Ske\_CryptoPrimitives, [579](#)
  - Trng\_CryptoPrimitives, [579](#)
- eHSM\_If\_Asr\_KeyCfg\_lp.h, [580](#)
  - CRYPTO\_CERTIFICATE\_KEY\_NUM, [582](#)
  - CRYPTO\_EVITA\_CERTIFICATE\_KEY\_ELEMENT\_SIZE, [582](#)
  - CRYPTO\_EVITA\_CIPHER\_KEY\_ELEMENT\_SIZE, [582](#)
  - CRYPTO\_EVITA\_CIPHER\_KEY\_NUM, [583](#)
  - CRYPTO\_EVITA\_COPY\_KEY\_ELEMENT\_SIZE, [583](#)
  - CRYPTO\_EVITA\_DERIVE\_KEY\_ELEMENT\_SIZE, [583](#)
  - CRYPTO\_EVITA\_DERIVE\_KEY\_NUM, [583](#)
  - CRYPTO\_EVITA\_EXCHANGE\_KEY\_ELEMENT\_SIZE, [584](#)
  - CRYPTO\_EVITA\_EXCHANGE\_KEY\_NUM, [584](#)
  - CRYPTO\_EVITA\_EXPORT\_KEY\_ELEMENT\_SIZE, [584](#)
  - CRYPTO\_EVITA\_EXPORT\_KEY\_NUM, [584](#)
  - CRYPTO\_EVITA\_GENERATE\_KEY\_ELEMENT\_SIZE, [584](#)
  - CRYPTO\_EVITA\_GENERATE\_KEY\_NUM, [585](#)
  - CRYPTO\_EVITA\_IMPORT\_KEY\_ELEMENT\_SIZE, [585](#)
  - CRYPTO\_EVITA\_IMPORT\_KEY\_NUM, [585](#)
  - CRYPTO\_EVITA\_KEY\_COPY\_KEY\_NUM, [585](#)
  - CRYPTO\_EVITA\_KEY\_STATUS\_ELEMENT\_SIZE, [585](#)
  - CRYPTO\_EVITA\_KEY\_STATUS\_NUM, [586](#)
  - CRYPTO\_EVITA\_MAC\_KEY\_ELEMENT\_SIZE, [586](#)
  - CRYPTO\_EVITA\_REMOVE\_KEY\_ELEMENT\_SIZE, [586](#)
  - CRYPTO\_EVITA\_REMOVE\_KEY\_NUM, [586](#)
  - CRYPTO\_EVITA\_SIGNATURE\_KEY\_ELEMENT\_SIZE, [586](#)
  - CRYPTO\_EVITA\_SIGNATURE\_KEY\_NUM, [586](#)
  - CRYPTO\_INVALID\_KEY\_ID, [587](#)
  - CRYPTO\_KEY\_AEAD\_TAG\_SIZE, [587](#)

- CRYPTO\_KE\_CIPHER\_2NDKEY\_SIZE\_SIZE, 587
- CRYPTO\_KE\_CIPHER\_CIPHER\_ALG\_SIZE, 587
- CRYPTO\_KE\_CIPHER\_CURVE\_ID\_SIZE, 587
- CRYPTO\_KE\_CIPHER\_IV\_SIZE, 587
- CRYPTO\_KE\_CIPHER\_KDF\_ALG\_SIZE, 588
- CRYPTO\_KE\_CIPHER\_KEY\_SIZE, 588
- CRYPTO\_KE\_CIPHER\_MAC\_ALG\_SIZE, 588
- CRYPTO\_KE\_CIPHER\_MAC\_SIZE\_SIZE, 588
- CRYPTO\_KE\_CIPHER\_PROOF\_SIZE, 588
- CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY\_SIZE, 588
- CRYPTO\_KE\_COPY\_KEY\_TARGET\_KEY\_HANDLE\_SIZE, 589
- CRYPTO\_KE\_EXPORT\_KEY\_BLOB\_SIZE, 589
- CRYPTO\_KE\_EXPORT\_KEY\_SIZE, 589
- CRYPTO\_KE\_EXT\_SHE\_KEY\_SIZE, 589
- CRYPTO\_KE\_IMPORT\_KEY\_SIZE, 589
- CRYPTO\_KE\_IMPORTED\_KEY\_KEYHANDLE\_SIZE, 589
- CRYPTO\_KE\_KEY\_MATERIAL\_SIZE, 590
- CRYPTO\_KE\_KEY\_STATUS\_BLOB\_SIZE, 590
- CRYPTO\_KE\_KEY\_STATUS\_SIZE, 590
- CRYPTO\_KE\_KEYDERIVATION\_ALGORITHM\_SIZE, 590
- CRYPTO\_KE\_KEYDERIVATION\_ITERATIONS\_SIZE, 590
- CRYPTO\_KE\_KEYDERIVATION\_KEY\_SIZE, 590
- CRYPTO\_KE\_KEYDERIVATION\_KEYHANDLE\_SIZE, 591
- CRYPTO\_KE\_KEYDERIVATION\_PASSWD\_SIZE, 591
- CRYPTO\_KE\_KEYDERIVATION\_SALT\_SIZE, 591
- CRYPTO\_KE\_KEYDERIVATION\_TYPE\_SIZE, 591
- CRYPTO\_KE\_KEYEXCHANGE\_ALGORITHM\_SIZE, 591
- CRYPTO\_KE\_KEYEXCHANGE\_BASE\_SIZE, 591
- CRYPTO\_KE\_KEYEXCHANGE\_KEYINFO\_SIZE, 592
- CRYPTO\_KE\_KEYEXCHANGE\_OWN\_PUBKEY\_SIZE, 592
- CRYPTO\_KE\_KEYEXCHANGE\_PEER\_PUBKEY\_SIZE, 592
- CRYPTO\_KE\_KEYEXCHANGE\_PRIVKEY\_SIZE, 592
- CRYPTO\_KE\_KEYEXCHANGE\_PUBTYPE\_SIZE, 592
- CRYPTO\_KE\_KEYEXCHANGE\_SM2\_LOCAL\_TMP\_KEYINFO\_SIZE, 592
- CRYPTO\_KE\_KEYEXCHANGE\_SM2\_PEER\_TMP\_PUBKEY\_SIZE, 593
- CRYPTO\_KE\_KEYEXCHANGE\_SM2\_ROLE\_SIZE, 593
- CRYPTO\_KE\_KEYEXCHANGE\_SM2\_S1\_S2\_VALUE\_SIZE, 593
- CRYPTO\_KE\_KEYEXCHANGE\_SM2\_SA\_SB\_VALUE\_SIZE, 593
- CRYPTO\_KE\_KEYGENERATE\_ALGORITHM\_SIZE, 593
- CRYPTO\_KE\_KEYGENERATE\_DH\_KEY\_INFO\_SIZE, 593
- CRYPTO\_KE\_KEYGENERATE\_KEY\_SIZE, 594
- CRYPTO\_KE\_KEYGENERATE\_KEYINFO\_SIZE, 594
- CRYPTO\_KE\_KEYGENERATE\_SEED\_SIZE, 594
- CRYPTO\_KE\_MAC\_KEY\_SIZE, 594
- CRYPTO\_KE\_MAC\_PROOF\_SIZE, 594
- CRYPTO\_KE\_RANDOM\_ALGORITHM\_SIZE, 594
- CRYPTO\_KE\_RANDOM\_SEED\_STATE\_SIZE, 595
- CRYPTO\_KE\_REMOVE\_KEY\_SIZE, 595
- CRYPTO\_KE\_SHE\_PLAIN\_KEY\_SIZE, 595
- CRYPTO\_KE\_SIGNATURE\_KEY\_SIZE, 595
- CRYPTO\_KE\_SIGNATURE\_RSA\_CRT\_MODE\_SIZE, 595
- CRYPTO\_KE\_SIGNATURE\_TIMESTAMPED\_SIZE, 595
- CRYPTO\_KEY10\_EVITA\_IMPORT\_KEY, 596
- CRYPTO\_KEY11\_EVITA\_EXPORT\_KEY, 596
- CRYPTO\_KEY12\_EVITA\_REMOVE\_KEY, 596
- CRYPTO\_KEY13\_EVITA\_KEY\_STATUS, 596
- CRYPTO\_KEY14\_EVITA\_KEY\_COPY\_KEY, 596
- CRYPTO\_KEY15\_EVITA\_KEY\_COPY\_KEY, 596
- CRYPTO\_KEY16\_CERTIFICATE\_KEY, 597
- CRYPTO\_KEY17\_CERTIFICATE\_KEY, 597
- CRYPTO\_KEY18\_MAC\_KEY, 597
- CRYPTO\_KEY1\_SHE\_KEY, 597
- CRYPTO\_KEY2\_SHE\_PLAIN\_KEY, 597
- CRYPTO\_KEY3\_EVITA\_SIGNATURE\_KEY, 597
- CRYPTO\_KEY4\_EVITA\_CIPHER\_KEY, 598
- CRYPTO\_KEY5\_EVITA\_EXCHANGE\_KEY, 598
- CRYPTO\_KEY6\_EVITA\_EXCHANGE\_KEY, 598
- CRYPTO\_KEY7\_EVITA\_DERIVE\_KEY, 598
- CRYPTO\_KEY8\_EVITA\_DERIVE\_KEY, 598
- CRYPTO\_KEY9\_EVITA\_GENERATE\_KEY, 598
- CRYPTO\_MAC\_KEY\_NUM, 599
- CRYPTO\_MAX\_KEY\_NUM, 599
- CRYPTO\_SHE\_KEY\_ELEMENT\_SIZE, 599
- CRYPTO\_SHE\_KEY\_NUM, 599
- CRYPTO\_SHE\_PLAIN\_KEY\_ELEMENT\_SIZE, 599
- CRYPTO\_SHE\_PLAIN\_KEY\_NUM, 600
- CRYPTO\_KE\_KEYEXCHANGE\_SHAREDVALUE\_SIZE, 600
- crypto\_object\_type\_e, 601
- ehsm\_get\_crypto\_driver\_object, 602
- ehsm\_get\_crypto\_ke\_info, 602
- ehsm\_printf\_ke\_size, 602
- KEY\_ELEMENT\_NUM, 600
- KEY\_ELEMENT\_RESERVER, 600
- KEY\_ELEMENT\_VALUE\_BUFFER\_RESERVER, 601
- KEY\_ELEMENT\_VALUE\_BUFFER\_SIZE, 601
- eHSM\_If\_Asr\_Types\_lp.h, 602
- CRYPTO\_INDEX\_M1\_U32, 605
- CRYPTO\_INDEX\_M2\_U32, 605
- CRYPTO\_INDEX\_M3\_U32, 605
- CRYPTO\_INDEX\_M4\_U32, 605
- CRYPTO\_INDEX\_M5\_U32, 606
- CRYPTO\_KE\_CERTIFICATE\_SIGNEDDATA, 606
- CRYPTO\_KE\_CIPHER\_CIPHER\_ALG, 606
- CRYPTO\_KE\_CIPHER\_CURVE\_ID, 606
- CRYPTO\_KE\_CIPHER\_KDF\_ALG, 606
- CRYPTO\_KE\_CIPHER\_MAC\_ALG, 606
- CRYPTO\_KE\_CIPHER\_MAC\_SIZE, 607
- CRYPTO\_KE\_CIPHER\_TAG\_SIZE, 607
- CRYPTO\_KE\_COPY\_KEY\_PARENT\_KEY, 607

CRYPTO\_KEY\_COPY\_KEY\_TARGET\_KEY\_HANDLE, 607  
 CRYPTO\_KEY\_EXPORT\_KEY\_BLOB, 607  
 CRYPTO\_KEY\_EXPORT\_KEY, 607  
 CRYPTO\_KEY\_EXT\_SHE\_KEY, 608  
 CRYPTO\_KEY\_IMPORT\_KEY, 608  
 CRYPTO\_KEY\_IMPORTED\_KEY\_KEYHANDLE, 608  
 CRYPTO\_KEY\_KEY\_HANDLE, 608  
 CRYPTO\_KEY\_KEY\_MATERIAL, 608  
 CRYPTO\_KEY\_KEY\_STATUS\_BLOB, 609  
 CRYPTO\_KEY\_KEY\_STATUS, 608  
 CRYPTO\_KEY\_KEYDERIVATION\_KEYHANDLE, 609  
 CRYPTO\_KEY\_KEYDERIVATION\_KEY, 609  
 CRYPTO\_KEY\_KEYDERIVATION\_TYPE, 609  
 CRYPTO\_KEY\_KEYEXCHANGE\_KEYINFO, 609  
 CRYPTO\_KEY\_KEYEXCHANGE\_PEERKEY, 610  
 CRYPTO\_KEY\_KEYEXCHANGE\_PUBTYPE, 610  
 CRYPTO\_KEY\_KEYEXCHANGE\_SM2\_LOCALTMP↔KINFO, 610  
 CRYPTO\_KEY\_KEYEXCHANGE\_SM2\_PEERTMP↔UBK, 610  
 CRYPTO\_KEY\_KEYEXCHANGE\_SM2\_ROLE, 610  
 CRYPTO\_KEY\_KEYEXCHANGE\_SM2\_S1\_S2\_VAL↔UE, 610  
 CRYPTO\_KEY\_KEYEXCHANGE\_SM2\_SA\_SB\_VAL↔UE, 611  
 CRYPTO\_KEY\_KEYGENERATE\_DH\_KEY\_INFO, 611  
 CRYPTO\_KEY\_KEYGENERATE\_KEYINFO, 611  
 CRYPTO\_KEY\_REMOVE\_KEY, 611  
 CRYPTO\_KEY\_SIGNATURE\_RSA\_CERT\_MODE, 611  
 CRYPTO\_KEY\_SIGNATURE\_TIMESTAMPED, 611  
 CRYPTO\_KEY\_ATTR\_ALLOW\_PARTIAL\_ACCESS, 612  
 CRYPTO\_KEY\_ATTR\_READ\_ACCESS, 612  
 CRYPTO\_KEY\_ATTR\_WRITE\_ACCESS, 612  
 CRYPTO\_M1\_SIZE\_U32, 612  
 CRYPTO\_M2\_SIZE\_U32, 612  
 CRYPTO\_M3\_SIZE\_U32, 612  
 CRYPTO\_M4\_SIZE\_U32, 613  
 CRYPTO\_M5\_SIZE\_U32, 613  
 CRYPTO\_MAX\_AUTH\_VALUE\_SIZE, 613  
 CRYPTO\_MAX\_KEY\_ELEMENTS\_OF\_KEY\_TYPE, 613  
 CRYPTO\_MAX\_KEY\_FLAG\_ELEMENT, 613  
 CRYPTO\_SHE\_MAC\_KEY\_ID, 613, 614  
 CRYPTO\_SHE\_MAC\_KEY\_NUM, 614  
 CRYPTO\_SHE\_SIZE\_IN\_U32, 614  
 CRYPTO\_SHE\_SIZE\_OUT\_U32, 614  
 CryIf\_CallbackNotification, 619  
 crypto\_copy\_key\_dh\_key\_info\_st, 616  
 crypto\_copy\_key\_info\_st, 616  
 crypto\_create\_evita\_key\_info\_st, 616  
 crypto\_evita\_key\_info\_st, 617  
 crypto\_exported\_key\_st, 617  
 crypto\_import\_evita\_key\_info\_st, 617  
 crypto\_key\_derive\_info\_st, 617  
 crypto\_key\_export\_info\_st, 617  
 crypto\_key\_status\_info\_st, 617  
 crypto\_key\_type\_e, 617  
 crypto\_she\_key\_st, 617  
 EVITA\_KEY\_USE\_FLAG\_DECRYPT, 614  
 EVITA\_KEY\_USE\_FLAG\_DHKE, 614  
 EVITA\_KEY\_USE\_FLAG\_ENCRYPT, 615  
 EVITA\_KEY\_USE\_FLAG\_REMOVE, 615  
 EVITA\_KEY\_USE\_FLAG\_SECUREBOOT, 615  
 EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE, 615  
 EVITA\_KEY\_USE\_FLAG\_SIGN, 615  
 EVITA\_KEY\_USE\_FLAG\_TIMESTAMP, 615  
 EVITA\_KEY\_USE\_FLAG\_TRANSPORT, 616  
 EVITA\_KEY\_USE\_FLAG\_UTCSYNC, 616  
 EVITA\_KEY\_USE\_FLAG\_VERIFY, 616  
 ehsm\_crypto\_key\_st, 617  
 ehsm\_key\_type\_e, 618  
 ehsm\_key\_use\_state\_e, 618  
 key\_storage\_type\_e, 618  
 eHSM\_If\_Evita\_AsymCper\_Ip.c, 619  
 Sign\_Finish, 620  
 Sign\_Init, 620  
 Sign\_Update, 621  
 Verify\_Finish, 622  
 Verify\_Init, 622  
 Verify\_Update, 623  
 eHSM\_If\_Evita\_Counter\_Ip.c, 624  
 eHSM\_If\_Evita\_ErrCode\_Ip.c, 624  
 ehsm\_evita\_convert\_ret\_code, 624  
 Evita\_Check\_Authorization\_Code, 625  
 Evita\_Check\_Key\_Handle, 625  
 eHSM\_If\_Evita\_ErrCode\_Ip.h, 626  
 ehsm\_evita\_convert\_ret\_code, 626  
 Evita\_Check\_Authorization\_Code, 627  
 Evita\_Check\_Key\_Handle, 627  
 eHSM\_If\_Evita\_Hash\_Ip.c, 628  
 HASH\_SIZE, 629  
 Hash\_Finish, 630  
 Hash\_Init, 630  
 Hash\_Update, 631  
 X, 629  
 eHSM\_If\_Evita\_Ip.h, 632  
 Aead\_Finish, 635  
 Aead\_Init, 636  
 Aead\_Process, 637  
 Check\_Time\_Stamp, 637  
 Cipher\_Finish, 638  
 Cipher\_Init, 639  
 Cipher\_Process, 640  
 Create\_Counter, 640  
 Create\_Derived\_Key, 641  
 Create\_Dh\_Key, 642  
 Create\_Random\_Dh\_Key\_Pair, 643  
 Create\_Random\_Key, 644  
 Create\_Time\_Stamp, 645  
 Delete\_Counter, 646  
 ehsm\_get\_pub\_from\_priv, 646  
 ehsm\_self\_test, 647  
 Get\_Tick\_Count, 647  
 Get\_Time\_Sync\_Challenge, 648  
 Get\_UTC\_Time, 648  
 Hash\_Finish, 649  
 Hash\_Init, 649  
 Hash\_Update, 650  
 Increment\_Counter, 651  
 Key\_Export, 651



- Key\_Import, [652](#)
- Key\_Remove, [653](#)
- Key\_Status, [654](#)
- MAC\_Finish, [655](#)
- MAC\_Init, [655](#)
- MAC\_Update, [656](#)
- Module\_Status, [657](#)
- RNG\_Get\_Random, [658](#)
- Read\_Counter, [658](#)
- Set\_UTC\_Time, [659](#)
- Sign\_Finish, [660](#)
- Sign\_Init, [660](#)
- Sign\_Update, [661](#)
- Verify\_Finish, [662](#)
- Verify\_Init, [662](#)
- Verify\_Update, [663](#)
- eHSM\_If\_Evita\_Key\_Ip.c, [664](#)
  - \_translate\_bool\_array\_to\_use\_flags, [665](#)
  - Create\_Derived\_Key, [665](#)
  - Create\_Dh\_Key, [666](#)
  - Create\_Random\_Dh\_Key\_Pair, [667](#)
  - Create\_Random\_Key, [668](#)
  - ehsm\_get\_pub\_from\_priv, [669](#)
  - Key\_Export, [670](#)
  - Key\_Import, [671](#)
  - Key\_Remove, [672](#)
  - Key\_Status, [673](#)
  - Module\_Status, [674](#)
- eHSM\_If\_Evita\_Rng\_Ip.c, [674](#)
  - RNG\_Get\_Random, [675](#)
- eHSM\_If\_Evita\_SymCper\_Ip.c, [676](#)
  - Aead\_Finish, [677](#)
  - Aead\_Init, [677](#)
  - Aead\_Process, [678](#)
  - Cipher\_Finish, [679](#)
  - Cipher\_Init, [680](#)
  - Cipher\_Process, [681](#)
  - MAC\_Finish, [681](#)
  - MAC\_Init, [683](#)
  - MAC\_Update, [684](#)
- eHSM\_If\_Evita\_Timer\_Ip.c, [684](#)
- eHSM\_If\_Evita\_Types\_Ip.h, [685](#)
  - ALIGN\_BYTE, [688](#), [708](#)
  - activeUseFlag, [708](#)
  - algo\_id, [708](#)
  - attr, [708](#)
  - auth\_sign\_data, [708](#)
  - auth\_size, [709](#)
  - auth\_value, [709](#)
  - cert\_data, [709](#)
  - cert\_size, [709](#)
  - cipher\_mode\_e, [705](#)
  - dh\_pubkey\_bytes\_size, [709](#)
  - EHSM\_CONTEXT\_SIZE, [689](#)
  - EHSM\_ECC\_KEY\_PAIR\_MAX\_SIZE, [689](#)
  - EHSM\_EVITA\_KEY\_NUM, [689](#)
  - EHSM\_KEY\_AUTH\_VALUE\_MAX\_SIZE, [689](#)
  - EHSM\_KEY\_DATA\_MAX\_SIZE, [689](#), [690](#)
  - EHSM\_KEY\_HEAD\_SIZE, [690](#)
  - EHSM\_KEY\_SIZE\_INFO\_MAX\_LEN, [690](#)
  - EHSM\_RSA\_DH\_KEY\_PAIR\_MAX\_SIZE, [690](#)
  - EHSM\_RSA\_KEY\_PAIR\_MAX\_SIZE, [690](#)
  - EHSM\_SM2\_SM2\_KEY\_MAX\_SIZE, [690](#)
  - EHSM\_SYM\_KEY\_PAIR\_MAX\_SIZE, [691](#)
  - EVITA\_ALGORITHM\_ERROR, [691](#)
  - EVITA\_ALL\_COUNTERS\_OCCUPIED, [691](#)
  - EVITA\_ALL\_KEY\_SPACE\_OCCUPIED, [691](#)
  - EVITA\_ALL\_SESSIONS\_OCCUPIED, [691](#)
  - EVITA\_AUTH\_TYPE\_NONE, [691](#)
  - EVITA\_AUTH\_TYPE\_PASSWD, [692](#)
  - EVITA\_AUTHORIZATION\_FAILED, [692](#)
  - EVITA\_CLOCK\_NOT\_SYNCHRONIZED, [692](#)
  - EVITA\_GENERAL\_ERROR, [692](#)
  - EVITA\_HASH\_BUF\_SIZE, [692](#)
  - EVITA\_INVALID\_COUNTER\_INCREMENTATION, [692](#)
  - EVITA\_INVALID\_KEY\_FLAG, [693](#)
  - EVITA\_INVALID\_KEY\_SIZE, [693](#)
  - EVITA\_INVALID\_MSG\_SIZE, [693](#)
  - EVITA\_INVALID\_TIME\_STAMP, [693](#)
  - EVITA\_INVALID\_UTC\_TIME, [693](#)
  - EVITA\_KEY\_DERIVE\_KDFX963, [693](#)
  - EVITA\_KEY\_DERIVE\_PBKDF2, [694](#)
  - EVITA\_KEY\_MAX\_SIZE, [694](#)
  - EVITA\_KEY\_SIGNATRUE\_SIZE, [694](#)
  - EVITA\_KEY\_TRNSP\_EXT, [694](#)
  - EVITA\_KEY\_TRNSP\_INI, [694](#)
  - EVITA\_KEY\_TRNSP\_MIG, [694](#)
  - EVITA\_KEY\_TRNSP\_OEM, [695](#)
  - EVITA\_KEY\_USE\_FLAG\_DECRYPT, [695](#)
  - EVITA\_KEY\_USE\_FLAG\_ENCRYPT, [695](#)
  - EVITA\_KEY\_USE\_FLAG\_KEYCREATION, [695](#)
  - EVITA\_KEY\_USE\_FLAG\_REMOVE, [695](#)
  - EVITA\_KEY\_USE\_FLAG\_SECUREBOOT, [695](#)
  - EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE, [696](#)
  - EVITA\_KEY\_USE\_FLAG\_SIGN, [696](#)
  - EVITA\_KEY\_USE\_FLAG\_TIMESTAMP, [696](#)
  - EVITA\_KEY\_USE\_FLAG\_TRANSPORT, [696](#)
  - EVITA\_KEY\_USE\_FLAG\_UTCSYNC, [696](#)
  - EVITA\_KEY\_USE\_FLAG\_VERIFY, [696](#)
  - EVITA\_MAC\_BUF\_SIZE, [697](#)
  - EVITA\_MAC\_LENGTH\_OVERSIZE, [697](#)
  - EVITA\_MAX\_CHUNK\_SIZE, [697](#)
  - EVITA\_MAX\_OTP\_KEY\_SIZE, [697](#)
  - EVITA\_MAX\_RANDOM\_KEY\_SIZE, [697](#)
  - EVITA\_OTP\_ASYM\_KEY\_SIZE, [698](#)
  - EVITA\_OTP\_PRIVKEY\_SIZE, [698](#)
  - EVITA\_OTP\_PUBKEY\_SIZE, [698](#)
  - EVITA\_OTP\_SYM\_KEY\_SIZE, [698](#)
  - EVITA\_OK, [697](#)
  - EVITA\_PRNG\_REQUEST\_OVERSIZE, [698](#)
  - EVITA\_REMOVE\_IMPOSSIBLE, [698](#)
  - EVITA\_SALT\_VALUE\_MAX\_SIZE, [699](#)
  - EVITA\_SIGNATURE\_BUF\_SIZE, [699](#)
  - EVITA\_STATUS\_TYPE\_NOT\_AVAILABLE, [699](#)
  - EVITA\_TEST\_CASE\_FAILED, [699](#)
  - EVITA\_TEST\_CASE\_NOT\_AVAILABLE, [699](#)
  - EVITA\_TRANSPORT\_IMPOSSIBLE, [699](#)
  - EVITA\_TRNG\_SEED\_FAILURE, [700](#)
  - EVITA\_UNKNOWN\_COUNTER\_ID, [700](#)
  - EVITA\_UTC\_CHALLENGE\_EXPIRED, [700](#)
  - EVITA\_UTC\_SYNCHRONIZATION\_FAILED, [700](#)

- EVITA\_WRONG\_AUTHORIZATION, [700](#)
- EVITA\_WRONG\_CERT\_KEY\_HANDLE, [700](#)
- EVITA\_WRONG\_CHUNK\_SIZE, [701](#)
- EVITA\_WRONG\_ECR\_INDEX, [701](#)
- EVITA\_WRONG\_IV, [701](#)
- EVITA\_WRONG\_KEY\_COMBINATION, [701](#)
- EVITA\_WRONG\_KEY\_HANDLE, [701](#)
- EVITA\_WRONG\_REMOTE\_KEY\_HANDLE, [701](#)
- EVITA\_WRONG\_SESSION\_HANDLE, [702](#)
- ehsm\_counter\_value\_st, [702](#)
- ehsm\_dh\_param\_size\_info\_st, [702](#)
- ehsm\_dh\_param\_st, [702](#)
- ehsm\_dh\_prikey\_st, [703](#)
- ehsm\_dh\_pubkey\_st, [703](#)
- ehsm\_ecc\_key\_size\_st, [703](#)
- ehsm\_ecc\_pubkey\_st, [703](#)
- ehsm\_evita\_key\_handle\_e, [705](#)
- ehsm\_key\_attr\_data\_st, [703](#)
- ehsm\_key\_signatrue\_st, [703](#)
- ehsm\_prikey\_data\_st, [703](#)
- ehsm\_pubkey\_data\_st, [703](#)
- ehsm\_rsa\_ctr\_st, [704](#)
- ehsm\_rsa\_dh\_key\_size\_st, [704](#)
- ehsm\_rsa\_key\_size\_st, [704](#)
- ehsm\_rsa\_pubkey\_st, [704](#)
- ehsm\_sym\_key\_size\_st, [704](#)
- ehsm\_tick\_value\_st, [704](#)
- ehsm\_utc\_time\_t, [704](#)
- evita\_internal\_key, [709](#)
- hash\_hmac\_st, [704](#)
- hash\_mode\_e, [706](#)
- key, [710](#)
- key\_data, [710](#)
- key\_handle, [710](#)
- key\_pub\_data, [710](#)
- key\_sign\_data, [710](#)
- key\_signatrue, [710](#)
- key\_size\_info, [711](#)
- key\_usage, [711](#)
- keyId, [711](#)
- keyIdSize, [711](#)
- mac\_mode\_e, [706](#)
- mac\_st, [705](#)
- mem\_location, [711](#)
- operation\_mode\_e, [706](#)
- padding\_scheme\_e, [707](#)
- prikey, [711](#)
- prikey\_enc\_size, [712](#)
- pubkey, [712](#)
- RNG\_REQUEST\_MAX, [702](#)
- reserved, [712](#)
- rsa\_e\_bytes\_size, [712](#)
- SKE\_CTX\_BUF\_SIZE, [702](#)
- session\_status\_e, [707](#)
- signature\_st, [705](#)
- size\_info, [712](#)
- storage\_key\_type\_e, [707](#)
- time\_stamp\_st, [705](#)
- valid\_util, [712](#)
- eHSM\_If\_Ext\_Ip.h, [713](#)
  - ehsm\_change\_controlfield, [715](#)
  - ehsm\_change\_lifecycle, [716](#)
  - ehsm\_close\_debug, [716](#)
  - ehsm\_debug\_auth, [717](#)
  - ehsm\_encrypt\_key, [717](#)
  - ehsm\_get\_challenge, [718](#)
  - ehsm\_get\_emu\_status, [719](#)
  - ehsm\_get\_random\_key, [719](#)
  - ehsm\_hsm\_fw\_upgrade\_finish, [720](#)
  - ehsm\_hsm\_fw\_upgrade\_init, [721](#)
  - ehsm\_hsm\_fw\_upgrade\_update, [721](#)
  - ehsm\_hsm\_fw\_upgrade\_verify\_finish, [722](#)
  - ehsm\_hsm\_fw\_upgrade\_verify\_init, [723](#)
  - ehsm\_hsm\_fw\_upgrade\_verify\_update, [723](#)
  - ehsm\_low\_power, [724](#)
  - ehsm\_power\_mode\_e, [715](#)
  - ehsm\_read\_otp\_data, [725](#)
  - ehsm\_secure\_boot, [725](#)
  - ehsm\_set\_uart\_baudrate, [726](#)
  - ehsm\_sm9\_cipher, [726](#)
  - ehsm\_sm9\_exchg\_key, [727](#)
  - ehsm\_sm9\_exckey\_gen\_tmpkey, [728](#)
  - ehsm\_sm9\_export\_key, [729](#)
  - ehsm\_sm9\_gen\_mastpubkey\_from\_mastprivkey, [730](#)
  - ehsm\_sm9\_gen\_tmppubkey\_from\_tmpprivkey, [730](#)
  - ehsm\_sm9\_generate\_master\_key, [732](#)
  - ehsm\_sm9\_generate\_priv\_key, [732](#)
  - ehsm\_sm9\_import\_key, [733](#)
  - ehsm\_sm9\_remove\_key, [734](#)
  - ehsm\_sm9\_sign, [734](#)
  - ehsm\_sm9\_unwrap\_key, [735](#)
  - ehsm\_sm9\_wrap\_key, [736](#)
  - ehsm\_write\_otp\_data, [737](#)
- eHSM\_If\_Ext\_Sm9\_Ip.c, [737](#)
  - EHSM\_CRYPT0\_V\_SM9\_MAX\_ID\_SIZE, [738](#)
- eHSM\_If\_Ext\_SysMgr\_Ip.c, [738](#)
  - EHSM\_IMAGE\_VERIFY\_TYPE\_FW\_UPGRADE, [740](#)
  - EHSM\_IMAGE\_VERIFY\_TYPE\_SOC\_BOOT, [740](#)
  - EHSM\_IMAGE\_VERIFY\_TYPE\_SOC\_UPGRADE, [740](#)
  - ehsm\_change\_controlfield, [740](#)
  - ehsm\_change\_lifecycle, [741](#)
  - ehsm\_close\_debug, [741](#)
  - ehsm\_debug\_auth, [742](#)
  - ehsm\_encrypt\_key, [743](#)
  - ehsm\_get\_challenge, [743](#)
  - ehsm\_get\_emu\_status, [744](#)
  - ehsm\_get\_random\_key, [744](#)
  - ehsm\_hsm\_fw\_upgrade\_finish, [745](#)
  - ehsm\_hsm\_fw\_upgrade\_init, [746](#)
  - ehsm\_hsm\_fw\_upgrade\_update, [746](#)
  - ehsm\_hsm\_fw\_upgrade\_verify\_finish, [747](#)
  - ehsm\_hsm\_fw\_upgrade\_verify\_init, [748](#)
  - ehsm\_hsm\_fw\_upgrade\_verify\_update, [748](#)
  - ehsm\_low\_power, [749](#)
  - ehsm\_read\_otp\_data, [750](#)
  - ehsm\_secure\_boot, [750](#)
  - ehsm\_self\_test, [751](#)
  - ehsm\_set\_uart\_baudrate, [751](#)
  - ehsm\_write\_otp\_data, [752](#)
  - g\_ctx, [753](#)
- eHSM\_If\_Ext\_Types\_Ip.h, [753](#)

- CONFIG\_SM9\_ENC\_DEFAULT\_HID\_VALUE, [753](#)
- CONFIG\_SM9\_EXCHG\_DEFAULT\_HID\_VALUE, [753](#)
- CONFIG\_SM9\_SIGN\_DEFAULT\_HID\_VALUE, [754](#)
- ehsm\_gen\_key\_alg\_e, [754](#)
- SM2\_PUBLIC\_KEY\_SIZE, [754](#)
- SM2\_S1\_S2\_SIZE, [754](#)
- eHSM\_If\_She\_ErrCode\_lp.c, [755](#)
  - ehsm\_she\_convert\_ret\_code, [755](#)
- eHSM\_If\_She\_ErrCode\_lp.h, [756](#)
  - ehsm\_she\_convert\_ret\_code, [756](#)
- eHSM\_If\_She\_lp.c, [756](#)
  - CONFIG\_EHSM\_KMGR\_V\_SHE\_BOOT\_MAC\_↔  
K, [758](#)
  - CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_MIN, [758](#)
  - EHSM\_SOC\_BOOT\_STATUS\_FAIL, [758](#)
  - EHSM\_SOC\_BOOT\_STATUS\_OK, [758](#)
  - ehsm\_she\_cancel, [758](#)
  - she\_boot\_failure, [759](#)
  - she\_boot\_ok, [759](#)
  - she\_crypto\_cbc, [760](#)
  - she\_crypto\_ecb, [760](#)
  - she\_crypto\_ecb\_extend, [761](#)
  - she\_debug, [761](#)
  - she\_export\_ram\_key, [762](#)
  - she\_extend\_seed, [763](#)
  - she\_generate\_mac, [763](#)
  - she\_get\_id, [765](#)
  - she\_get\_status, [766](#)
  - she\_init\_rng, [766](#)
  - she\_load\_key, [767](#)
  - she\_load\_key\_extend, [768](#)
  - she\_load\_plain\_key, [768](#)
  - she\_rnd, [769](#)
  - she\_secure\_boot, [770](#)
  - she\_verify\_mac, [770](#)
- eHSM\_If\_She\_lp.h, [771](#)
  - AES\_CTRDRBG, [772](#)
  - ehsm\_she\_cancel, [772](#)
  - SM4\_CTRDRBG, [772](#)
  - she\_boot\_failure, [773](#)
  - she\_boot\_ok, [773](#)
  - she\_crypto\_cbc, [774](#)
  - she\_crypto\_ecb, [775](#)
  - she\_crypto\_ecb\_extend, [775](#)
  - she\_debug, [776](#)
  - she\_export\_ram\_key, [776](#)
  - she\_extend\_seed, [777](#)
  - she\_generate\_mac, [778](#)
  - she\_get\_id, [778](#)
  - she\_get\_status, [779](#)
  - she\_init\_rng, [779](#)
  - she\_load\_key, [780](#)
  - she\_load\_key\_extend, [781](#)
  - she\_load\_plain\_key, [782](#)
  - she\_rnd, [782](#)
  - she\_secure\_boot, [783](#)
  - she\_verify\_mac, [784](#)
- eHSM\_If\_She\_Types\_lp.h, [784](#)
  - EHSM\_SHE\_KEY\_MAC\_VERIFY\_ONLY, [786](#)
  - EHSM\_SHE\_KEY\_MAX\_SIZE, [786](#)
  - EHSM\_SHE\_KEY\_PROP\_BOOT\_PRT, [786](#)
  - EHSM\_SHE\_KEY\_PROP\_DEBUG\_PRT, [786](#)
  - EHSM\_SHE\_KEY\_PROP\_KEY\_USAGE, [786](#)
  - EHSM\_SHE\_KEY\_PROP\_WILDCARD, [786](#)
  - EHSM\_SHE\_KEY\_PROP\_WR\_PRT, [786](#)
  - EHSM\_SHE\_KEY\_SIZE, [787](#)
  - EHSM\_SHE\_M1\_STD\_SIZE, [787](#)
  - EHSM\_SHE\_M4\_STD\_SIZE, [787](#)
  - EHSM\_SHE\_NVM\_KEY\_NUM, [787](#)
  - EHSM\_SHE\_OTP\_KEY\_ATTR\_SIZE, [787](#)
  - EHSM\_SHE\_UID\_MAX\_SIZE, [787](#)
  - ERC\_BUSY, [788](#)
  - ERC\_GENERAL\_ERROR, [788](#)
  - ERC\_KEY\_EMPTY, [788](#)
  - ERC\_KEY\_INVALID, [788](#)
  - ERC\_KEY\_NOT\_AVAILABLE, [788](#)
  - ERC\_KEY\_UPDATE\_ERROR, [789](#)
  - ERC\_KEY\_WRITE\_PROTECTED, [789](#)
  - ERC\_MEMORY\_FAILURE, [789](#)
  - ERC\_NO\_DEBUGGING, [789](#)
  - ERC\_NO\_ERROR, [789](#)
  - ERC\_NO\_SECURE\_BOOT, [790](#)
  - ERC\_RNG\_SEED, [790](#)
  - ERC\_SEQUENCE\_ERROR, [790](#)
  - ehsm\_she\_key\_handle\_e, [790](#)
  - ehsm\_she\_status\_type\_, [791](#)
  - ehsm\_she\_status\_type\_e, [790](#)
- eHSM\_IntCfg\_lp.h, [791](#)
  - CONFIG\_EHSM\_ARCH\_MAIN\_HOOK, [795](#)
  - CONFIG\_EHSM\_ARCH\_MULTI\_CHANNEL, [795](#)
  - CONFIG\_EHSM\_ARCH\_OS\_NONE, [796](#)
  - CONFIG\_EHSM\_ARCH\_SHARE\_MEM, [796](#)
  - CONFIG\_EHSM\_ARCH\_V\_REQ\_HASH\_MAX\_SIZE,  
[796](#)
  - CONFIG\_EHSM\_ARCH\_V\_REQ\_SKE\_MAX\_SIZE,  
[796](#)
  - CONFIG\_EHSM\_COUNTER\_AUTO\_INCREASE, [796](#)
  - CONFIG\_EHSM\_CRYPT0\_AEAD, [797](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_AES, [797](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_CTRDRBG,  
[797](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_DH, [797](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_ECC, [797](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_ED25519,  
[797](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_MD5, [798](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_PBKDF2,  
[798](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_1024,  
[798](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_1024↔  
\_CRT, [798](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_2048,  
[798](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA\_2048↔  
\_CRT, [799](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_RSA, [798](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SECP256R1,  
[799](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SECP384R1,  
[799](#)
  - CONFIG\_EHSM\_CRYPT0\_ALGOFAM\_SHA1, [799](#)

CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA2, [799](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA224, [799](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA256, [800](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA384, [800](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA512, [800](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA512\_↵  
224, [800](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SHA512\_↵  
256, [800](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SM2, [800](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SM3, [801](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_SM4, [801](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOFAM\_X963, [801](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CBC\_MAC,  
[801](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CBC, [801](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CFB, [801](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CMAC, [802](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_CTR, [802](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_ECB, [802](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_GMAC, [802](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_OFB, [802](#)  
CONFIG\_EHSM\_CRYPTO\_ALGOMODE\_RSASSA\_↵  
PSS, [802](#)  
CONFIG\_EHSM\_CRYPTO\_RSA\_CRT\_MODE, [803](#)  
CONFIG\_EHSM\_CRYPTO\_V\_CERT\_MAX\_PUB\_↵  
K\_SIZE, [803](#)  
CONFIG\_EHSM\_CRYPTO\_V\_CERT\_MAX\_SIZE,  
[803](#)  
CONFIG\_EHSM\_CRYPTO\_V\_GCM\_MAX\_AAD\_SI\_↵  
ZE, [803](#)  
CONFIG\_EHSM\_CRYPTO\_V\_HMAC\_MAX\_KSIZE,  
[803](#)  
CONFIG\_EHSM\_FIRMWARE\_UPGRADE, [804](#)  
CONFIG\_EHSM\_HASH\_CORE\_NUM, [804](#)  
CONFIG\_EHSM\_HW\_AHB\_BYTE, [804](#)  
CONFIG\_EHSM\_HW\_BRANCH\_2\_0\_0, [804](#)  
CONFIG\_EHSM\_HW\_COUNTER, [804](#)  
CONFIG\_EHSM\_HW\_FLASH\_ECC, [805](#)  
CONFIG\_EHSM\_HW\_FLASH, [805](#)  
CONFIG\_EHSM\_HW\_GUOMI\_LEVEL1, [805](#)  
CONFIG\_EHSM\_HW\_HASH\_DMA, [805](#)  
CONFIG\_EHSM\_HW\_HASH\_LP, [806](#)  
CONFIG\_EHSM\_HW\_HASH, [805](#)  
CONFIG\_EHSM\_HW\_INSTALL\_K\_KEY, [806](#)  
CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DEBUG\_MODE,  
[806](#)  
CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DESTROY\_M\_↵  
ODE, [806](#)  
CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_DEVELOP\_M\_↵  
ODE, [806](#)  
CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_MANUFACTU\_↵  
RE\_MODE, [807](#)  
CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_TEST\_MODE,  
[807](#)  
CONFIG\_EHSM\_HW\_LIFE\_CYCLE\_USER\_MODE,  
[807](#)  
CONFIG\_EHSM\_HW\_LOW\_POWER, [807](#)  
CONFIG\_EHSM\_HW\_OTP\_MAP, [808](#)  
CONFIG\_EHSM\_HW\_OTP, [807](#)  
CONFIG\_EHSM\_HW\_PKE\_LP, [808](#)  
CONFIG\_EHSM\_HW\_PKE, [808](#)  
CONFIG\_EHSM\_HW\_SKE\_DMA, [808](#)  
CONFIG\_EHSM\_HW\_SKE\_LP, [808](#)  
CONFIG\_EHSM\_HW\_SKE\_SECURE\_PORT, [809](#)  
CONFIG\_EHSM\_HW\_TRNG, [809](#)  
CONFIG\_EHSM\_HW\_UTC\_TIME, [809](#)  
CONFIG\_EHSM\_HW\_V\_CODE\_MAX\_SIZE, [809](#)  
CONFIG\_EHSM\_HW\_V\_CUSTOMER\_OTP, [809](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_BASE\_ADDR, [810](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_CUSTOMER, [810](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_DATA\_ADDR, [810](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_DATA\_SIZE, [810](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_ERASE\_CELL\_V\_↵  
ALUE, [810](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_FREE\_ECC\_IDX1,  
[810](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_FREE\_ECC\_IDX2,  
[811](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_KEY\_ADDR, [811](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_KEY\_SIZE, [811](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_LOG\_ADDR, [811](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_LOG\_SIZE, [811](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_NONE, [811](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_PAGE\_SIZE, [812](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_SIMULATE, [812](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_SIZE, [812](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_SYS\_ADDR, [812](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_SYS\_SIZE, [812](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_TYPE, [812](#)  
CONFIG\_EHSM\_HW\_V\_FLASH\_WRITE\_MIN\_BY\_↵  
TES, [813](#)  
CONFIG\_EHSM\_HW\_V\_NONE\_OTP, [813](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_BASE\_ADDR, [813](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_K\_ATTR\_LENGTH,  
[813](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_KEY\_NUM, [813](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_PAGE\_SIZE, [813](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_SIZE, [814](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_TYPE, [814](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_VERSION\_LENGTH,  
[814](#)  
CONFIG\_EHSM\_HW\_V\_OTP\_WRITE\_MIN\_BYTES,  
[814](#)  
CONFIG\_EHSM\_HW\_V\_SIMULATE\_OTP, [814](#)  
CONFIG\_EHSM\_HW\_V\_WORK\_FREQ, [814](#)  
CONFIG\_EHSM\_JTAG\_DEBUG\_AUTH, [815](#)  
CONFIG\_EHSM\_KMGR\_CHECK\_OTP\_K\_ATTR,  
[815](#)  
CONFIG\_EHSM\_KMGR\_PLAIN\_K\_IMPORT, [815](#)  
CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_AREA\_SIZE,  
[815](#)  
CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_END\_ADDR,  
[815](#)  
CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_NUM, [816](#)  
CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_SLOT\_SIZE,  
[816](#)  
CONFIG\_EHSM\_KMGR\_V\_ECC\_K\_START\_ADDR,  
[816](#)  
CONFIG\_EHSM\_KMGR\_V\_ECC\_RAM\_K\_NUM, [816](#)  
CONFIG\_EHSM\_KMGR\_V\_FLASH\_K\_START\_OF\_↵  
FSET, [816](#)

CONFIG\_EHSM\_KMGR\_V\_FLASH\_OTP\_K\_START\_OFFSET, [816](#)  
 CONFIG\_EHSM\_KMGR\_V\_MAX\_AUTH\_CODE\_SIZE, [817](#)  
 CONFIG\_EHSM\_KMGR\_V\_OTP\_EXT\_K\_OFF, [817](#)  
 CONFIG\_EHSM\_KMGR\_V\_OTP\_EXT\_K\_SIZE, [817](#)  
 CONFIG\_EHSM\_KMGR\_V\_RAM\_K\_MEM\_SIZE, [817](#)  
 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_AREA\_SIZE, [817](#)  
 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_END\_ADDR, [817](#)  
 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_NUM, [818](#)  
 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_SLOT\_SIZE, [818](#)  
 CONFIG\_EHSM\_KMGR\_V\_RSA\_K\_START\_ADDR, [818](#)  
 CONFIG\_EHSM\_KMGR\_V\_RSA\_RAM\_K\_NUM, [818](#)  
 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_AREA\_SIZE, [818](#)  
 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_END\_ADDR, [818](#)  
 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_NUM, [819](#)  
 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_SLOT\_SIZE, [819](#)  
 CONFIG\_EHSM\_KMGR\_V\_SHE\_K\_START\_ADDR, [819](#)  
 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_AREA\_SIZE, [819](#)  
 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_END\_ADDR, [819](#)  
 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_NUM, [820](#)  
 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_SLOT\_SIZE, [820](#)  
 CONFIG\_EHSM\_KMGR\_V\_SM9\_K\_START\_ADDR, [820](#)  
 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_AREA\_SIZE, [820](#)  
 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_END\_ADDR, [820](#)  
 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_NUM, [820](#)  
 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_SLOT\_SIZE, [821](#)  
 CONFIG\_EHSM\_KMGR\_V\_SYM\_K\_START\_ADDR, [821](#)  
 CONFIG\_EHSM\_KMGR\_V\_SYM\_RAM\_K\_NUM, [821](#)  
 CONFIG\_EHSM\_LOG, [821](#)  
 CONFIG\_EHSM\_PKE\_CORE\_NUM, [821](#)  
 CONFIG\_EHSM\_SHE\_SOC\_BOOT, [821](#)  
 CONFIG\_EHSM\_SKE\_CORE\_NUM, [822](#)  
 CONFIG\_EHSM\_SOC\_UPGRADE\_AND\_VERIFY, [822](#)  
 CONFIG\_EHSM\_TRNG\_CORE\_NUM, [822](#)  
 CONFIG\_EHSM\_UNIT\_TEST, [822](#)  
 CONFIG\_EHSM\_USER\_AUTH\_KEY\_IN\_KMU, [822](#)  
 CONFIG\_EHSM\_V\_LOG\_DEBUG, [823](#)  
 CONFIG\_EHSM\_V\_LOG\_ERR, [823](#)  
 CONFIG\_EHSM\_V\_LOG\_INFO, [823](#)  
 CONFIG\_EHSM\_V\_LOG\_LEVEL, [823](#)  
 CONFIG\_EHSM\_V\_LOG\_WARN, [823](#)  
 EHSM\_KEY\_AUTH\_VALUE\_MAX\_SIZE  
     eHSM\_If\_Evita\_Types\_Ip.h, [689](#)  
 EHSM\_KEY\_DATA\_MAX\_SIZE  
     eHSM\_If\_Evita\_Types\_Ip.h, [689](#), [690](#)  
 EHSM\_KEY\_HEAD\_SIZE  
     eHSM\_If\_Evita\_Types\_Ip.h, [690](#)  
 EHSM\_KEY\_SIZE\_INFO\_MAX\_LEN  
     eHSM\_If\_Evita\_Types\_Ip.h, [690](#)  
 EHSM\_LOW\_POWER\_MODE  
     eHSM\_Mailbox\_Prtcl\_Ip.h, [853](#)  
 EHSM\_MAC\_GENERATION  
     eHSM\_Mailbox\_CmdId\_Ip.h, [838](#)  
 EHSM\_MAC\_VERIFICATION  
     eHSM\_Mailbox\_CmdId\_Ip.h, [838](#)  
 EHSM\_MD5  
     eHSM\_Mailbox\_CmdId\_Ip.h, [838](#)  
 eHSM\_Mailbox\_CmdId\_Ip.h, [824](#)  
     cmd\_type\_e, [844](#)  
 EHSM\_AES\_128, [826](#)  
 EHSM\_AES\_192, [826](#)  
 EHSM\_AES\_256, [826](#)  
 EHSM\_AES\_CTRDRBG, [826](#)  
 EHSM\_CBC\_MAC\_MODE, [827](#)  
 EHSM\_CBC\_MODE, [827](#)  
 EHSM\_CFB\_MODE, [827](#)  
 EHSM\_CMAC\_MODE, [827](#)  
 EHSM\_CMD\_AEAD\_CCM, [827](#)  
 EHSM\_CMD\_AEAD\_GCM, [827](#)  
 EHSM\_CMD\_CHANGE\_CONTROL\_FIELD, [828](#)  
 EHSM\_CMD\_CHANGE\_LIFECYCLE, [828](#)  
 EHSM\_CMD\_CLOSE\_DEBUG, [828](#)  
 EHSM\_CMD\_COPY\_EVITA\_KEY, [828](#)  
 EHSM\_CMD\_CREATE\_COUNTER, [828](#)  
 EHSM\_CMD\_CREATE\_DH\_KEY, [828](#)  
 EHSM\_CMD\_DEBUG\_AUTHENTICATION, [829](#)  
 EHSM\_CMD\_DELETE\_COUNTER, [829](#)  
 EHSM\_CMD\_DERIVE\_KEY, [829](#)  
 EHSM\_CMD\_ECCP\_GEN\_KEY, [829](#)  
 EHSM\_CMD\_ECDSA, [829](#)  
 EHSM\_CMD\_ECIES, [829](#)  
 EHSM\_CMD\_EXPORT\_KEY, [830](#)  
 EHSM\_CMD\_FW\_ENCRYPT\_KEY, [830](#)  
 EHSM\_CMD\_FW\_GET\_RANDOM\_KEY, [830](#)  
 EHSM\_CMD\_GEN\_DH\_KEY\_PAIR, [830](#)  
 EHSM\_CMD\_GET\_CHALLENGE, [830](#)  
 EHSM\_CMD\_GET\_PUB\_FROM\_PRIV, [830](#)  
 EHSM\_CMD\_GET\_SHE\_ID, [831](#)  
 EHSM\_CMD\_GET\_SHE\_STATUS, [831](#)  
 EHSM\_CMD\_HASH, [831](#)  
 EHSM\_CMD\_IMAGE\_UPGRADE, [831](#)  
 EHSM\_CMD\_IMAGE\_VERIFY, [831](#)  
 EHSM\_CMD\_IMPORT\_KEY, [831](#)  
 EHSM\_CMD\_INCREASE\_COUNTER, [832](#)  
 EHSM\_CMD\_KEY\_REMOVE, [832](#)  
 EHSM\_CMD\_KEY\_STATUS, [832](#)  
 EHSM\_CMD\_LOW\_POWER, [832](#)  
 EHSM\_CMD\_MAC, [832](#)  
 EHSM\_CMD\_MODULE\_STATUS, [832](#)  
 EHSM\_CMD\_READ\_COUNTER, [833](#)  
 EHSM\_CMD\_READ\_OTP\_DATA, [833](#)  
 EHSM\_CMD\_RESET\_FIRMWARE, [833](#)  
 EHSM\_CMD\_RNG\_GENERATE, [833](#)  
 EHSM\_CMD\_RSA\_CIPHER, [833](#)  
 EHSM\_CMD\_RSA\_GEN\_KEY, [833](#)  
 EHSM\_CMD\_RSA\_SIGN, [834](#)  
 EHSM\_CMD\_SELF\_TEST, [834](#)



- EHSM\_CMD\_SENSOR\_RESP\_INIT, [834](#)
- EHSM\_CMD\_SET\_BAUDRATE, [834](#)
- EHSM\_CMD\_SHE\_LOAD\_KEY, [834](#)
- EHSM\_CMD\_SHE\_LOAD\_PLAIN\_KEY, [834](#)
- EHSM\_CMD\_SHE\_RAM\_KEY\_EXPORT, [835](#)
- EHSM\_CMD\_SM2\_CIPHER, [835](#)
- EHSM\_CMD\_SM2\_GEN\_KEY, [835](#)
- EHSM\_CMD\_SM2\_SIGN, [835](#)
- EHSM\_CMD\_SOC\_BOOT\_STATUS, [835](#)
- EHSM\_CMD\_SOC\_IMAGE\_VERIFY, [835](#)
- EHSM\_CMD\_SYM\_CIPHER, [836](#)
- EHSM\_CMD\_SYM\_GEN\_KEY, [836](#)
- EHSM\_CMD\_UART\_COMMAND, [836](#)
- EHSM\_CMD\_WRITE\_OTP\_DATA, [836](#)
- EHSM\_CRT\_MODE, [836](#)
- EHSM\_CTR\_MODE, [837](#)
- EHSM\_DECRYPTION, [837](#)
- EHSM\_DES, [837](#)
- EHSM\_ECB\_MODE, [837](#)
- EHSM\_ENCRYPTION, [837](#)
- EHSM\_FINISH, [837](#)
- EHSM\_GMAC\_MODE, [838](#)
- EHSM\_INVALID\_ALG, [838](#)
- EHSM\_INVALID\_DIR, [838](#)
- EHSM\_MAC\_GENERATION, [838](#)
- EHSM\_MAC\_VERIFICATION, [838](#)
- EHSM\_MD5, [838](#)
- EHSM\_NO\_TIME\_STAMP, [839](#)
- EHSM\_NONE\_CRT, [839](#)
- EHSM\_NOPADDING, [839](#)
- EHSM\_OFB\_MODE, [839](#)
- EHSM\_ONEPASS, [839](#)
- EHSM\_ONEWITHZEROS, [839](#)
- EHSM\_PKCS7, [840](#)
- EHSM\_RSASSA\_PPS, [840](#)
- EHSM\_SHA1, [840](#)
- EHSM\_SHA224, [840](#)
- EHSM\_SHA256, [840](#)
- EHSM\_SHA384, [840](#)
- EHSM\_SHA3\_224, [841](#)
- EHSM\_SHA3\_256, [841](#)
- EHSM\_SHA3\_384, [841](#)
- EHSM\_SHA3\_512, [841](#)
- EHSM\_SHA512, [841](#)
- EHSM\_SHA512\_224, [841](#)
- EHSM\_SHA512\_256, [842](#)
- EHSM\_SIGN\_GENERATION, [842](#)
- EHSM\_SIGN\_VERIFICATION, [842](#)
- EHSM\_SM3, [842](#)
- EHSM\_SM4, [842](#)
- EHSM\_SM4\_CTRDRBG, [842](#)
- EHSM\_START, [843](#)
- EHSM\_STREAMSTART, [843](#)
- EHSM\_TDES\_128, [843](#)
- EHSM\_TDES\_192, [843](#)
- EHSM\_UPDATE, [843](#)
- EHSM\_USE\_TIME\_STAMP, [843](#)
- EHSM\_XOR, [844](#)
- EHSM\_XTS\_MODE, [844](#)
- eHSM\_Mailbox\_lp.c, [844](#)
- \_\_attribute\_\_, [845](#)
- ehsm\_is\_cmd\_addr\_null, [846](#)
- ehsm\_mbox\_init, [846](#)
- ehsm\_mbox\_polling, [846](#)
- ehsm\_mbox\_send\_cmd, [846](#)
- HSMMBX\_IRQ\_PRIO, [845](#)
- MAILBOX\_Handler, [846](#)
- mbox\_channel, [847](#)
- eHSM\_Mailbox\_lp.h, [847](#)
- ehsm\_get\_tick, [848](#)
- ehsm\_is\_cmd\_addr\_null, [848](#)
- ehsm\_mbox\_init, [848](#)
- ehsm\_mbox\_send\_cmd, [848](#)
- ehsm\_tick\_form\_ms, [848](#)
- mailbox\_callback, [847](#)
- mailbox\_channel\_st, [847](#)
- eHSM\_Mailbox\_Prtcl\_lp.h, [849](#)
- CMD\_TAG\_BYTE\_SIZE, [852](#)
- CMD\_TAG\_WORD\_INDEX, [852](#)
- CMD\_TAG\_WORD\_SIZE, [852](#)
- EHSM\_CANCEL\_CERT\_TYPE\_CMD, [852](#)
- EHSM\_CANCEL\_SINGLE\_CMD, [853](#)
- EHSM\_CMD\_CIPHER\_KEY\_TYPE\_EVITA, [853](#)
- EHSM\_CMD\_CIPHER\_KEY\_TYPE\_SHE, [853](#)
- EHSM\_LOW\_POWER\_MODE, [853](#)
- EHSM\_NORMAL\_MODE, [853](#)
- ehsm\_change\_control\_field\_cmd\_st, [859](#)
- ehsm\_change\_lifecycle\_cmd\_st, [859](#)
- ehsm\_close\_debug\_cmd\_st, [860](#)
- ehsm\_copy\_key\_cmd\_st, [860](#)
- ehsm\_create\_dh\_key\_cmd\_st, [860](#)
- ehsm\_create\_dh\_sm2\_ext\_param\_st, [860](#)
- ehsm\_debug\_authentication\_cmd\_st, [860](#)
- ehsm\_derive\_key\_cmd\_st, [860](#)
- ehsm\_export\_key\_cmd\_st, [860](#)
- ehsm\_fw\_encrypt\_key\_cmd\_st, [860](#)
- ehsm\_fw\_get\_random\_key\_cmd\_st, [861](#)
- ehsm\_gen\_key\_cmd\_st, [861](#)
- ehsm\_gen\_sm9\_userpriv\_key\_cmd\_st, [861](#)
- ehsm\_get\_challenge\_cmd\_st, [861](#)
- ehsm\_get\_emu\_cmd\_st, [861](#)
- ehsm\_get\_pub\_from\_priv\_cmd\_st, [861](#)
- ehsm\_get\_she\_id\_cmd\_st, [861](#)
- ehsm\_image\_upgrade\_cmd\_st, [861](#)
- ehsm\_image\_verify\_cmd\_st, [862](#)
- ehsm\_import\_key\_cmd\_st, [862](#)
- ehsm\_key\_remove\_cmd\_st, [862](#)
- ehsm\_key\_status\_cmd\_st, [862](#)
- ehsm\_low\_power\_cmd\_st, [862](#)
- ehsm\_mailbox\_req\_st, [862](#)
- ehsm\_mbox\_cancel\_channel\_req\_st, [862](#)
- ehsm\_mbox\_cancel\_channel\_rps\_st, [862](#)
- ehsm\_mbox\_mgr\_channel\_req\_st, [863](#)
- ehsm\_module\_status\_cmd\_st, [863](#)
- ehsm\_otp\_read\_cmd\_st, [863](#)
- ehsm\_otp\_write\_cmd\_st, [863](#)
- ehsm\_rng\_generate\_cmd\_st, [863](#)
- ehsm\_self\_test\_cmd\_st, [863](#)
- ehsm\_sensor\_resp\_init\_cmd\_st, [863](#)
- ehsm\_set\_baudrate\_cmd\_st, [863](#)
- ehsm\_she\_load\_export\_key\_cmd\_st, [864](#)
- ehsm\_she\_load\_plain\_key\_cmd\_st, [864](#)

- ehsm\_sm9\_exchg\_gen\_usertmp\_cmd\_st, 864
- ehsm\_sm9\_exchg\_key\_cmd\_child\_st, 864
- ehsm\_sm9\_exchg\_key\_cmd\_st, 864
- ehsm\_sm9\_export\_key\_cmd\_st, 864
- ehsm\_sm9\_get\_mast\_pubkey\_cmd\_st, 864
- ehsm\_sm9\_get\_tmp\_pubkey\_cmd\_st, 864
- ehsm\_sm9\_import\_key\_cmd\_st, 865
- ehsm\_sm9\_remove\_key\_cmd\_st, 865
- ehsm\_sm9\_unwrap\_key\_cmd\_st, 865
- ehsm\_sm9\_wrap\_key\_cmd\_st, 865
- ehsm\_soc\_image\_verify\_cmd\_st, 865
- ehsm\_uart\_cmd\_st, 865
- H2S\_SRV\_CMD\_CANCEL\_NOTE\_BIT, 853
- H2S\_SRV\_CMD\_CANCEL\_WORD\_INDEX, 854
- H2S\_SRV\_CMD\_CANCEL\_WORD\_SIZE, 854
- H2S\_SRV\_CMD\_JTAG\_END\_BIT, 854
- H2S\_SRV\_CMD\_JTAG\_NOTE\_BIT, 854
- H2S\_SRV\_GENERAL\_NOTE\_BIT, 854
- H2S\_SRV\_GENERAL\_WORD\_INDEX, 854
- H2S\_SRV\_GENERAL\_WORD\_SIZE, 855
- H2S\_SRV\_JTAG\_WORD\_INDEX, 855
- H2S\_SRV\_JTAG\_WORD\_SIZE, 855
- H2S\_SRV\_MGR\_NOTE\_BIT, 855
- H2S\_SRV\_MGR\_WORD\_INDEX, 855
- H2S\_SRV\_MGR\_WORD\_SIZE, 855
- HOST\_ADDRESS\_SIZE, 856
- MAILBOX\_CMD\_MAX\_SIZE, 856
- MAX\_IMPORT\_KEY\_SIZE, 856
- MAX\_KEY\_AUTH\_VALUE\_SIZE, 856
- MAX\_KEY\_DERIVED\_PWD\_SIZE, 856
- MAX\_SM9\_CIPHER\_KEY\_SIZE, 856
- MAX\_SM9\_USER\_ID\_SIZE, 857
- MAX\_SM9\_WARP\_KEY\_SIZE, 857
- mailbox\_channel\_e, 865
- RESPONSE\_TAG\_INDEX, 857
- S2H\_SRV\_CMD\_CANCEL\_NOTE\_BIT, 857
- S2H\_SRV\_CMD\_CANCEL\_WORD\_INDEX, 857
- S2H\_SRV\_CMD\_CANCEL\_WORD\_SIZE, 857
- S2H\_SRV\_CMD\_JTAG\_END\_BIT, 858
- S2H\_SRV\_CMD\_JTAG\_NOTE\_BIT, 858
- S2H\_SRV\_CMD\_JTAG\_WORD\_INDEX, 858
- S2H\_SRV\_GENERAL\_NOTE\_BIT, 858
- S2H\_SRV\_GENERAL\_WORD\_INDEX, 858
- S2H\_SRV\_GENERAL\_WORD\_SIZE, 858
- S2H\_SRV\_JTAG\_WORD\_SIZE, 859
- S2H\_SRV\_MGR\_NOTE\_BIT, 859
- S2H\_SRV\_MGR\_WORD\_INDEX, 859
- S2H\_SRV\_MGR\_WORD\_SIZE, 859
- eHSM\_Mailbox\_Reg\_Ip.h, 866
- HOST2HSM\_ACCESS\_PATH, 867
- HSM\_STATUS\_IN1, 867
- HSM\_STATUS\_IN, 867
- KBUF\_BASE, 867
- KMU\_BASE, 868
- MB\_H2S\_NOTE, 868
- MB\_H2S\_SOC\_INT\_EN, 868
- MB\_H2S\_SOC\_INT, 868
- MB\_HSM\_STATUS0, 868
- MB\_HSM\_STATUS1, 868
- MB\_S2H\_NOTE, 869
- MB\_S2H\_SOC\_INT\_EN, 869
- MB\_S2H\_SOC\_INT, 869
- MBOX\_HOST2HSM\_HOST\_INT\_EN, 869
- MBOX\_HOST2HSM\_HOST\_INT, 869
- MBOX\_HOST2HSM\_HSM\_INT\_EN, 870
- MBOX\_HOST2HSM\_HSM\_INT, 869
- MBOX\_HSM2HOST\_HOST\_INT\_EN, 870
- MBOX\_HSM2HOST\_HOST\_INT, 870
- MBOX\_HSMHOST\_HSM\_INT\_EN, 870
- MBOX\_HSMHOST\_HSM\_INT, 870
- MBOX\_SOCBASE, 870
- rATTR\_D0, 871
- rATTR\_D1, 871
- rATTR\_D2, 871
- rERR\_ST, 871
- rKBUF, 871
- rKMU\_CTRL, 871
- rKMU\_INT\_EN, 872
- rKMU\_STA, 872
- rSN\_D0, 872
- rSN\_D1, 872
- rSOCMBOX\_CMD\_D0, 872
- rSOCMBOX\_CMD\_D1, 872
- rSOCMBOX\_CMD\_D10, 873
- rSOCMBOX\_CMD\_D11, 873
- rSOCMBOX\_CMD\_D12, 873
- rSOCMBOX\_CMD\_D13, 873
- rSOCMBOX\_CMD\_D14, 873
- rSOCMBOX\_CMD\_D15, 873
- rSOCMBOX\_CMD\_D2, 874
- rSOCMBOX\_CMD\_D3, 874
- rSOCMBOX\_CMD\_D4, 874
- rSOCMBOX\_CMD\_D5, 874
- rSOCMBOX\_CMD\_D6, 874
- rSOCMBOX\_CMD\_D7, 874
- rSOCMBOX\_CMD\_D8, 875
- rSOCMBOX\_CMD\_D9, 875
- rSOCMBOX\_RSP\_D0, 875
- rSOCMBOX\_RSP\_D01, 875
- rVER\_D0, 875
- rVER\_D1, 875
- rVER\_D2, 876
- rVER\_D3, 876
- STATUS\_BASE, 876
- eHSM\_Mgr\_Cipher\_Ctx\_Find
- eHSM\_Mgr\_Ctx\_Ip.c, 878
- eHSM\_Mgr\_Ctx\_Ip.h, 881
- eHSM\_Mgr\_Cipher\_Ctx\_Free
- eHSM\_Mgr\_Ctx\_Ip.c, 878
- eHSM\_Mgr\_Ctx\_Ip.h, 882
- eHSM\_Mgr\_Cipher\_Ctx\_Get\_Free
- eHSM\_Mgr\_Ctx\_Ip.c, 878
- eHSM\_Mgr\_Ctx\_Ip.h, 882
- eHSM\_Mgr\_Ctx\_Ip.c, 876
- eHSM\_Mgr\_Cipher\_Ctx\_Find, 878
- eHSM\_Mgr\_Cipher\_Ctx\_Free, 878
- eHSM\_Mgr\_Cipher\_Ctx\_Get\_Free, 878
- ehsm\_cipher\_ctx\_data\_check, 877
- ehsm\_disjunction\_finish, 877
- ehsm\_disjunction\_updata, 877
- ehsm\_init\_block\_mgr, 877
- ehsm\_mgr\_ctx\_find, 878

- ehsm\_mgr\_ctx\_free, [878](#)
- ehsm\_mgr\_ctx\_get\_free, [879](#)
- eHSM\_Mgr\_Ctx\_Ip.h, [879](#)
- EHSM\_ERR\_CTX\_MGR\_BUFFER\_DATA\_VALID, [880](#)
- EHSM\_ERR\_CTX\_MGR\_DATA\_NOT\_READY, [880](#)
- eHSM\_Mgr\_Cipher\_Ctx\_Find, [881](#)
- eHSM\_Mgr\_Cipher\_Ctx\_Free, [882](#)
- eHSM\_Mgr\_Cipher\_Ctx\_Get\_Free, [882](#)
- ehsm\_aead\_data\_ptr\_st, [880](#)
- ehsm\_asym\_alg\_e, [880](#)
- ehsm\_cipher\_ctx\_data\_check, [881](#)
- ehsm\_ctx\_block\_mgr\_st, [880](#)
- ehsm\_disjunction\_finish, [881](#)
- ehsm\_disjunction\_updata, [881](#)
- ehsm\_init\_block\_mgr, [881](#)
- ehsm\_mgr\_ctx\_find, [882](#)
- ehsm\_mgr\_ctx\_free, [882](#)
- ehsm\_mgr\_ctx\_get\_free, [882](#)
- EHSM\_NO\_TIME\_STAMP
- eHSM\_Mailbox\_CmdId\_Ip.h, [839](#)
- EHSM\_NONE\_CRT
- eHSM\_Mailbox\_CmdId\_Ip.h, [839](#)
- EHSM\_NOPADDING
- eHSM\_Mailbox\_CmdId\_Ip.h, [839](#)
- EHSM\_NORMAL\_MODE
- eHSM\_Mailbox\_Prtcl\_Ip.h, [853](#)
- EHSM\_OFB\_MODE
- eHSM\_Mailbox\_CmdId\_Ip.h, [839](#)
- EHSM\_ONEPASS
- eHSM\_Mailbox\_CmdId\_Ip.h, [839](#)
- EHSM\_ONEWITHZEROS
- eHSM\_Mailbox\_CmdId\_Ip.h, [839](#)
- EHSM\_PKCS7
- eHSM\_Mailbox\_CmdId\_Ip.h, [840](#)
- EHSM\_RSA\_DH\_KEY\_PAIR\_MAX\_SIZE
- eHSM\_If\_Evita\_Types\_Ip.h, [690](#)
- EHSM\_RSA\_KEY\_PAIR\_MAX\_SIZE
- eHSM\_If\_Evita\_Types\_Ip.h, [690](#)
- EHSM\_RSASSA\_PPS
- eHSM\_Mailbox\_CmdId\_Ip.h, [840](#)
- EHSM\_SELF\_TEST\_ALL
- eHSM\_Com\_Struct\_Ip.h, [415](#)
- EHSM\_SELF\_TEST\_HASH\_MD5
- eHSM\_Com\_Struct\_Ip.h, [415](#)
- EHSM\_SELF\_TEST\_HASH\_SHA1
- eHSM\_Com\_Struct\_Ip.h, [415](#)
- EHSM\_SELF\_TEST\_HASH\_SHA2
- eHSM\_Com\_Struct\_Ip.h, [416](#)
- EHSM\_SELF\_TEST\_HASH\_SHA256
- eHSM\_Com\_Struct\_Ip.h, [416](#)
- EHSM\_SELF\_TEST\_HASH\_SHA3
- eHSM\_Com\_Struct\_Ip.h, [416](#)
- EHSM\_SELF\_TEST\_HASH\_SM3
- eHSM\_Com\_Struct\_Ip.h, [416](#)
- EHSM\_SELF\_TEST\_HASH
- eHSM\_Com\_Struct\_Ip.h, [415](#)
- EHSM\_SELF\_TEST\_PKE\_ECC
- eHSM\_Com\_Struct\_Ip.h, [416](#)
- EHSM\_SELF\_TEST\_PKE\_RSA
- eHSM\_Com\_Struct\_Ip.h, [417](#)
- EHSM\_SELF\_TEST\_PKE\_SM2
- eHSM\_Com\_Struct\_Ip.h, [417](#)
- EHSM\_SELF\_TEST\_PKE\_SM9
- eHSM\_Com\_Struct\_Ip.h, [417](#)
- EHSM\_SELF\_TEST\_PKE
- eHSM\_Com\_Struct\_Ip.h, [416](#)
- EHSM\_SELF\_TEST\_SKE\_AES
- eHSM\_Com\_Struct\_Ip.h, [417](#)
- EHSM\_SELF\_TEST\_SKE\_DES
- eHSM\_Com\_Struct\_Ip.h, [417](#)
- EHSM\_SELF\_TEST\_SKE\_SM4
- eHSM\_Com\_Struct\_Ip.h, [418](#)
- EHSM\_SELF\_TEST\_SKE\_TDES
- eHSM\_Com\_Struct\_Ip.h, [418](#)
- EHSM\_SELF\_TEST\_SKE
- eHSM\_Com\_Struct\_Ip.h, [417](#)
- EHSM\_SELF\_TEST\_TRNG
- eHSM\_Com\_Struct\_Ip.h, [418](#)
- EHSM\_SHA1
- eHSM\_Mailbox\_CmdId\_Ip.h, [840](#)
- EHSM\_SHA224
- eHSM\_Mailbox\_CmdId\_Ip.h, [840](#)
- EHSM\_SHA256
- eHSM\_Mailbox\_CmdId\_Ip.h, [840](#)
- EHSM\_SHA384
- eHSM\_Mailbox\_CmdId\_Ip.h, [840](#)
- EHSM\_SHA3\_224
- eHSM\_Mailbox\_CmdId\_Ip.h, [841](#)
- EHSM\_SHA3\_256
- eHSM\_Mailbox\_CmdId\_Ip.h, [841](#)
- EHSM\_SHA3\_384
- eHSM\_Mailbox\_CmdId\_Ip.h, [841](#)
- EHSM\_SHA3\_512
- eHSM\_Mailbox\_CmdId\_Ip.h, [841](#)
- EHSM\_SHA512
- eHSM\_Mailbox\_CmdId\_Ip.h, [841](#)
- EHSM\_SHA512\_224
- eHSM\_Mailbox\_CmdId\_Ip.h, [841](#)
- EHSM\_SHA512\_256
- eHSM\_Mailbox\_CmdId\_Ip.h, [842](#)
- EHSM\_SHE\_KEY\_MAC\_VERIFY\_ONLY
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_MAX\_SIZE
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_PROP\_BOOT\_PRT
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_PROP\_DEBUG\_PRT
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_PROP\_KEY\_USAGE
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_PROP\_WILDCARD
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_PROP\_WR\_PRT
- eHSM\_If\_She\_Types\_Ip.h, [786](#)
- EHSM\_SHE\_KEY\_SIZE
- eHSM\_If\_She\_Types\_Ip.h, [787](#)
- EHSM\_SHE\_M1\_MAX\_SIZE
- eHSM\_Com\_Struct\_Ip.h, [418](#)
- EHSM\_SHE\_M1\_STD\_SIZE
- eHSM\_If\_She\_Types\_Ip.h, [787](#)
- EHSM\_SHE\_M2\_MAX\_SIZE



- eHSM\_Com\_Struct\_lp.h, 418
- EHSM\_SHE\_M3\_MAX\_SIZE
  - eHSM\_Com\_Struct\_lp.h, 418
- EHSM\_SHE\_M4\_MAX\_SIZE
  - eHSM\_Com\_Struct\_lp.h, 419
- EHSM\_SHE\_M4\_STD\_SIZE
  - eHSM\_If\_She\_Types\_lp.h, 787
- EHSM\_SHE\_M5\_MAX\_SIZE
  - eHSM\_Com\_Struct\_lp.h, 419
- EHSM\_SHE\_NVM\_KEY\_NUM
  - eHSM\_If\_She\_Types\_lp.h, 787
- EHSM\_SHE\_OTP\_KEY\_ATTR\_SIZE
  - eHSM\_If\_She\_Types\_lp.h, 787
- EHSM\_SHE\_UID\_MAX\_SIZE
  - eHSM\_If\_She\_Types\_lp.h, 787
- EHSM\_SIGN\_GENERATION
  - eHSM\_Mailbox\_CmdId\_lp.h, 842
- EHSM\_SIGN\_VERIFICATION
  - eHSM\_Mailbox\_CmdId\_lp.h, 842
- EHSM\_SM2\_SM2\_KEY\_MAX\_SIZE
  - eHSM\_If\_Evita\_Types\_lp.h, 690
- EHSM\_SM3
  - eHSM\_Mailbox\_CmdId\_lp.h, 842
- EHSM\_SM4
  - eHSM\_Mailbox\_CmdId\_lp.h, 842
- EHSM\_SM4\_CTRDRBG
  - eHSM\_Mailbox\_CmdId\_lp.h, 842
- EHSM\_SOC\_BOOT\_STATUS\_FAIL
  - eHSM\_If\_She\_lp.c, 758
- EHSM\_SOC\_BOOT\_STATUS\_OK
  - eHSM\_If\_She\_lp.c, 758
- EHSM\_START
  - eHSM\_Mailbox\_CmdId\_lp.h, 843
- EHSM\_STREAMSTART
  - eHSM\_Mailbox\_CmdId\_lp.h, 843
- EHSM\_SYM\_KEY\_PAIR\_MAX\_SIZE
  - eHSM\_If\_Evita\_Types\_lp.h, 691
- eHSM\_Srv\_AsymCper\_lp.c, 883
  - srv\_crypto\_pke, 883
- eHSM\_Srv\_Ciper\_lp.c, 883
  - \_srv\_crypto\_cipher\_reqhdl, 884
  - \_srv\_crypto\_cipher\_rsphdl, 884
  - srv\_crypto\_ske, 884
- eHSM\_Srv\_Cipher\_lp.h, 885
  - \_srv\_crypto\_cipher\_reqhdl, 885
  - \_srv\_crypto\_cipher\_rsphdl, 885
  - ehsm\_cmd\_cipher\_with\_rps\_st, 885
- eHSM\_Srv\_CmdReq\_lp.h, 886
  - cmd\_release\_cb, 887
  - cmd\_req\_cb, 887
  - EHSM\_CMD\_PRIORITY\_DEFAULT, 886
  - ehsm\_cmd\_req\_st, 887
  - ehsm\_cmd\_req\_state\_e, 887
  - ehsm\_cmd\_req\_type\_e, 887
  - MAX\_RESPONSE\_DATA\_SIZE, 886
- eHSM\_Srv\_Counter\_lp.c, 889
- eHSM\_Srv\_Ext\_lp.c, 889
  - g\_srv\_soc\_image\_verify, 890
  - srv\_asr\_candle\_cmd, 890
  - srv\_bootloader\_cmd, 890
  - srv\_change\_control\_field, 891
  - srv\_change\_lifecycle, 891
  - srv\_close\_debug, 891
  - srv\_debug\_auth, 892
  - srv\_fw\_encrypt\_key, 892
  - srv\_fw\_get\_random\_key, 892
  - srv\_get\_challenge, 893
  - srv\_get\_module\_status, 893
  - srv\_get\_she\_id, 893
  - srv\_get\_she\_status, 894
  - srv\_image\_upgrade, 894
  - srv\_image\_verify, 894
  - srv\_low\_power, 895
  - srv\_read\_otp\_data, 895
  - srv\_self\_test, 895
  - srv\_set\_baudrate, 896
  - srv\_she\_candle\_cmd, 896
  - srv\_soc\_boot\_status, 896
  - srv\_write\_otp\_data, 897
- eHSM\_Srv\_Hash\_lp.c, 897
  - srv\_crypto\_hash, 897
- eHSM\_Srv\_Key\_lp.c, 898
  - DEFAULT\_RSA\_E\_SIZE, 899
  - srv\_certificate\_parse, 899
  - srv\_certificate\_verify, 899
  - srv\_create\_dh\_key, 899
  - srv\_create\_random\_key, 900
  - srv\_derive\_key, 900
  - srv\_export\_key, 900
  - srv\_get\_pub\_from\_priv, 901
  - srv\_import\_key, 901
  - srv\_key\_copy, 901
  - srv\_key\_remove, 902
  - srv\_key\_status, 902
  - srv\_she\_load\_key, 902
  - srv\_she\_load\_plain\_key, 903
  - srv\_she\_ram\_key\_export, 903
- eHSM\_Srv\_Mgr\_lp.c, 903
  - COMMAND\_REQ\_QUANTITY, 905
  - ehsm\_cancel\_single\_service, 905
  - ehsm\_cmd\_req\_buffer\_st, 905
  - ehsm\_process\_asr\_service, 906
  - ehsm\_process\_sync\_service, 906
  - ehsm\_register\_service, 906
  - ehsm\_service\_init, 906
  - ehsm\_set\_address\_pointer, 906
  - g\_cmd\_req\_buffer, 907
  - g\_srv\_soc\_image\_verify, 907
  - hw\_interrupt\_disable, 907
  - hw\_interrupt\_enable, 907
  - service\_table, 907
  - srv\_asr\_candle\_cmd, 907
  - srv\_bootloader\_cmd, 908
  - srv\_certificate\_parse, 908
  - srv\_certificate\_verify, 908
  - srv\_change\_control\_field, 908
  - srv\_change\_lifecycle, 908
  - srv\_close\_debug, 908
  - srv\_create\_dh\_key, 909
  - srv\_create\_random\_key, 909
  - srv\_crypto\_hash, 909
  - srv\_crypto\_pke, 909

- srv\_crypto\_randomgenerate, 909
- srv\_crypto\_ske, 909
- srv\_debug\_auth, 910
- srv\_derive\_key, 910
- srv\_export\_key, 910
- srv\_fw\_encrypt\_key, 910
- srv\_fw\_get\_random\_key, 910
- srv\_get\_challenge, 910
- srv\_get\_module\_status, 911
- srv\_get\_pub\_from\_priv, 911
- srv\_get\_she\_id, 911
- srv\_get\_she\_status, 911
- srv\_image\_upgrade, 911
- srv\_image\_verify, 911
- srv\_import\_key, 912
- srv\_key\_copy, 912
- srv\_key\_remove, 912
- srv\_key\_status, 912
- srv\_low\_power, 912
- srv\_read\_otp\_data, 912
- srv\_set\_baudrate, 913
- srv\_she\_cancel\_cmd, 913
- srv\_she\_load\_key, 913
- srv\_she\_load\_plain\_key, 913
- srv\_she\_ram\_key\_export, 913
- srv\_soc\_boot\_status, 913
- srv\_write\_otp\_data, 914
- eHSM\_Srv\_Mgr\_Ip.h, 914
  - ehsm\_cancel\_single\_service, 917
  - ehsm\_cmd\_ext\_type\_e, 916
  - ehsm\_process\_asr\_service, 917
  - ehsm\_process\_sync\_service, 918
  - ehsm\_register\_service, 918
  - ehsm\_service\_info\_st, 915
  - ehsm\_service\_init, 918
  - ehsm\_service\_st, 915
  - ehsm\_set\_address\_pointer, 918
  - service\_reqhdl, 915
  - service\_rsphdl, 915
- eHSM\_Srv\_Rng\_Ip.c, 919
  - srv\_crypto\_randomgenerate, 919
  - srv\_crypto\_randomseed, 919
  - srv\_rng\_extend\_seed, 919
  - srv\_rng\_init, 920
- eHSM\_Srv\_Timer\_Ip.c, 920
- EHSM\_TDES\_128
  - eHSM\_Mailbox\_CmdId\_Ip.h, 843
- EHSM\_TDES\_192
  - eHSM\_Mailbox\_CmdId\_Ip.h, 843
- eHSM\_Types\_Ip.h, 920
  - EHSM\_ARRAY\_SIZE, 921
  - ehsm\_addr\_t, 922
  - ehsm\_bool\_t, 922
  - ehsm\_handle\_t, 922
  - ehsm\_int16\_t, 922
  - ehsm\_int32\_t, 922
  - ehsm\_int64\_t, 922
  - ehsm\_int8\_t, 922
  - ehsm\_uint16\_t, 923
  - ehsm\_uint32\_t, 923
  - ehsm\_uint64\_t, 923
  - ehsm\_uint8\_t, 923
  - false, 921
  - NULL, 921
  - true, 921
- EHSM\_UPDATE
  - eHSM\_Mailbox\_CmdId\_Ip.h, 843
- EHSM\_USE\_TIME\_STAMP
  - eHSM\_Mailbox\_CmdId\_Ip.h, 843
- EHSM\_XOR
  - eHSM\_Mailbox\_CmdId\_Ip.h, 844
- EHSM\_XTS\_MODE
  - eHSM\_Mailbox\_CmdId\_Ip.h, 844
- ERC\_BUSY
  - eHSM\_If\_She\_Types\_Ip.h, 788
- ERC\_GENERAL\_ERROR
  - eHSM\_If\_She\_Types\_Ip.h, 788
- ERC\_KEY\_EMPTY
  - eHSM\_If\_She\_Types\_Ip.h, 788
- ERC\_KEY\_INVALID
  - eHSM\_If\_She\_Types\_Ip.h, 788
- ERC\_KEY\_NOT\_AVAILABLE
  - eHSM\_If\_She\_Types\_Ip.h, 788
- ERC\_KEY\_UPDATE\_ERROR
  - eHSM\_If\_She\_Types\_Ip.h, 789
- ERC\_KEY\_WRITE\_PROTECTED
  - eHSM\_If\_She\_Types\_Ip.h, 789
- ERC\_MEMORY\_FAILURE
  - eHSM\_If\_She\_Types\_Ip.h, 789
- ERC\_NO\_DEBUGGING
  - eHSM\_If\_She\_Types\_Ip.h, 789
- ERC\_NO\_ERROR
  - eHSM\_If\_She\_Types\_Ip.h, 789
- ERC\_NO\_SECURE\_BOOT
  - eHSM\_If\_She\_Types\_Ip.h, 790
- ERC\_RNG\_SEED
  - eHSM\_If\_She\_Types\_Ip.h, 790
- ERC\_SEQUENCE\_ERROR
  - eHSM\_If\_She\_Types\_Ip.h, 790
- EVITA\_ALGORITHM\_ERROR
  - eHSM\_If\_Evita\_Types\_Ip.h, 691
- EVITA\_ALL\_COUNTERS\_OCCUPIED
  - eHSM\_If\_Evita\_Types\_Ip.h, 691
- EVITA\_ALL\_KEY\_SPACE\_OCCUPIED
  - eHSM\_If\_Evita\_Types\_Ip.h, 691
- EVITA\_ALL\_SESSIONS\_OCCUPIED
  - eHSM\_If\_Evita\_Types\_Ip.h, 691
- EVITA\_AUTH\_TYPE\_NONE
  - eHSM\_If\_Evita\_Types\_Ip.h, 691
- EVITA\_AUTH\_TYPE\_PASSWD
  - eHSM\_If\_Evita\_Types\_Ip.h, 692
- EVITA\_AUTHORIZATION\_FAILED
  - eHSM\_If\_Evita\_Types\_Ip.h, 692
- EVITA\_CLOCK\_NOT\_SYNCHRONIZED
  - eHSM\_If\_Evita\_Types\_Ip.h, 692
- EVITA\_GENERAL\_ERROR
  - eHSM\_If\_Evita\_Types\_Ip.h, 692
- EVITA\_HASH\_BUF\_SIZE
  - eHSM\_If\_Evita\_Types\_Ip.h, 692
- EVITA\_INVALID\_COUNTER\_INCREMENTATION
  - eHSM\_If\_Evita\_Types\_Ip.h, 692
- EVITA\_INVALID\_KEY\_FLAG

eHSM\_If\_Evita\_Types\_lp.h, [693](#)  
EVITA\_INVALID\_KEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [693](#)  
EVITA\_INVALID\_MSG\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [693](#)  
EVITA\_INVALID\_TIME\_STAMP  
eHSM\_If\_Evita\_Types\_lp.h, [693](#)  
EVITA\_INVALID\_UTC\_TIME  
eHSM\_If\_Evita\_Types\_lp.h, [693](#)  
EVITA\_KEY\_DERIVE\_KDFX963  
eHSM\_If\_Evita\_Types\_lp.h, [693](#)  
EVITA\_KEY\_DERIVE\_PBKDF2  
eHSM\_If\_Evita\_Types\_lp.h, [694](#)  
EVITA\_KEY\_MAX\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [694](#)  
EVITA\_KEY\_SIGNATTRUE\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [694](#)  
EVITA\_KEY\_TRNSP\_EXT  
eHSM\_If\_Evita\_Types\_lp.h, [694](#)  
EVITA\_KEY\_TRNSP\_INI  
eHSM\_If\_Evita\_Types\_lp.h, [694](#)  
EVITA\_KEY\_TRNSP\_MIG  
eHSM\_If\_Evita\_Types\_lp.h, [694](#)  
EVITA\_KEY\_TRNSP\_OEM  
eHSM\_If\_Evita\_Types\_lp.h, [695](#)  
EVITA\_KEY\_USE\_FLAG\_DECRYPT  
eHSM\_If\_Asr\_Types\_lp.h, [614](#)  
eHSM\_If\_Evita\_Types\_lp.h, [695](#)  
EVITA\_KEY\_USE\_FLAG\_DHKE  
eHSM\_If\_Asr\_Types\_lp.h, [614](#)  
EVITA\_KEY\_USE\_FLAG\_ENCRYPT  
eHSM\_If\_Asr\_Types\_lp.h, [615](#)  
eHSM\_If\_Evita\_Types\_lp.h, [695](#)  
EVITA\_KEY\_USE\_FLAG\_KEYCREATION  
eHSM\_If\_Evita\_Types\_lp.h, [695](#)  
EVITA\_KEY\_USE\_FLAG\_REMOVE  
eHSM\_If\_Asr\_Types\_lp.h, [615](#)  
eHSM\_If\_Evita\_Types\_lp.h, [695](#)  
EVITA\_KEY\_USE\_FLAG\_SECUREBOOT  
eHSM\_If\_Asr\_Types\_lp.h, [615](#)  
eHSM\_If\_Evita\_Types\_lp.h, [695](#)  
EVITA\_KEY\_USE\_FLAG\_SECURESTORAGE  
eHSM\_If\_Asr\_Types\_lp.h, [615](#)  
eHSM\_If\_Evita\_Types\_lp.h, [696](#)  
EVITA\_KEY\_USE\_FLAG\_SIGN  
eHSM\_If\_Asr\_Types\_lp.h, [615](#)  
eHSM\_If\_Evita\_Types\_lp.h, [696](#)  
EVITA\_KEY\_USE\_FLAG\_TIMESTAMP  
eHSM\_If\_Asr\_Types\_lp.h, [615](#)  
eHSM\_If\_Evita\_Types\_lp.h, [696](#)  
EVITA\_KEY\_USE\_FLAG\_TRANSPORT  
eHSM\_If\_Asr\_Types\_lp.h, [616](#)  
eHSM\_If\_Evita\_Types\_lp.h, [696](#)  
EVITA\_KEY\_USE\_FLAG\_UTCSYNC  
eHSM\_If\_Asr\_Types\_lp.h, [616](#)  
eHSM\_If\_Evita\_Types\_lp.h, [696](#)  
EVITA\_KEY\_USE\_FLAG\_VERIFY  
eHSM\_If\_Asr\_Types\_lp.h, [616](#)  
eHSM\_If\_Evita\_Types\_lp.h, [696](#)  
EVITA\_LOG\_DEBUG  
eHSM\_Debug\_lp.h, [491](#)  
EVITA\_LOG\_ERROR  
eHSM\_Debug\_lp.h, [491](#)  
EVITA\_LOG\_INFO  
eHSM\_Debug\_lp.h, [491](#)  
EVITA\_LOG\_WARN  
eHSM\_Debug\_lp.h, [491](#)  
EVITA\_MAC\_BUF\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [697](#)  
EVITA\_MAC\_LENGTH\_OVERSIZE  
eHSM\_If\_Evita\_Types\_lp.h, [697](#)  
EVITA\_MAX\_CHUNK\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [697](#)  
EVITA\_MAX\_OTP\_KEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [697](#)  
EVITA\_MAX\_RANDOM\_KEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [697](#)  
EVITA\_OTP\_ASYM\_KEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [698](#)  
EVITA\_OTP\_PRIVKEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [698](#)  
EVITA\_OTP\_PUBKEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [698](#)  
EVITA\_OTP\_SYM\_KEY\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [698](#)  
EVITA\_OK  
eHSM\_If\_Evita\_Types\_lp.h, [697](#)  
EVITA\_PRNG\_REQUEST\_OVERSIZE  
eHSM\_If\_Evita\_Types\_lp.h, [698](#)  
EVITA\_PURE\_LOG\_DEBUG  
eHSM\_Debug\_lp.h, [492](#)  
EVITA\_REMOVE\_IMPOSSIBLE  
eHSM\_If\_Evita\_Types\_lp.h, [698](#)  
EVITA\_SALT\_VALUE\_MAX\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [699](#)  
EVITA\_SIGNATURE\_BUF\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [699](#)  
EVITA\_STATUS\_TYPE\_NOT\_AVAILABLE  
eHSM\_If\_Evita\_Types\_lp.h, [699](#)  
EVITA\_TEST\_CASE\_FAILED  
eHSM\_If\_Evita\_Types\_lp.h, [699](#)  
EVITA\_TEST\_CASE\_NOT\_AVAILABLE  
eHSM\_If\_Evita\_Types\_lp.h, [699](#)  
EVITA\_TRANSPORT\_IMPOSSIBLE  
eHSM\_If\_Evita\_Types\_lp.h, [699](#)  
EVITA\_TRNG\_SEED\_FAILURE  
eHSM\_If\_Evita\_Types\_lp.h, [700](#)  
EVITA\_UNKNOWN\_COUNTER\_ID  
eHSM\_If\_Evita\_Types\_lp.h, [700](#)  
EVITA\_UTC\_CHALLENGE\_EXPIRED  
eHSM\_If\_Evita\_Types\_lp.h, [700](#)  
EVITA\_UTC\_SYNCHRONIZATION\_FAILED  
eHSM\_If\_Evita\_Types\_lp.h, [700](#)  
EVITA\_WRONG\_AUTHORIZATION  
eHSM\_If\_Evita\_Types\_lp.h, [700](#)  
EVITA\_WRONG\_CERT\_KEY\_HANDLE  
eHSM\_If\_Evita\_Types\_lp.h, [700](#)  
EVITA\_WRONG\_CHUNK\_SIZE  
eHSM\_If\_Evita\_Types\_lp.h, [701](#)  
EVITA\_WRONG\_ECR\_INDEX  
eHSM\_If\_Evita\_Types\_lp.h, [701](#)  
EVITA\_WRONG\_IV

- eHSM\_If\_Evita\_Types\_lp.h, [701](#)
- EVITA\_WRONG\_KEY\_COMBINATION
  - eHSM\_If\_Evita\_Types\_lp.h, [701](#)
- EVITA\_WRONG\_KEY\_HANDLE
  - eHSM\_If\_Evita\_Types\_lp.h, [701](#)
- EVITA\_WRONG\_REMOTE\_KEY\_HANDLE
  - eHSM\_If\_Evita\_Types\_lp.h, [701](#)
- EVITA\_WRONG\_SESSION\_HANDLE
  - eHSM\_If\_Evita\_Types\_lp.h, [702](#)
- Ecc
  - Hsm\_PubKeyDataType\_, [327](#)
- ecc
  - ehsm\_pubkey\_data\_, [219](#)
- ecc\_k
  - ehsm\_prikey\_data\_, [216](#)
- EccKey
  - Hsm\_PriKeyDataType\_, [326](#)
- ecies\_testvec, [59](#)
  - cipher\_part1, [60](#)
  - cipher\_part2\_part3\_with\_s1, [60](#)
  - cipher\_part2\_part3\_with\_s1\_s2, [60](#)
  - cipher\_part2\_part3\_with\_s2, [60](#)
  - cipher\_part2\_part3\_without\_s1\_s2, [60](#)
  - curve\_id, [60](#)
  - kdf\_hash\_alg, [61](#)
  - mac\_hash\_alg, [61](#)
  - mac\_k\_bytes, [61](#)
  - msg, [61](#)
  - msg\_bytes, [61](#)
  - point\_form, [61](#)
  - receiver\_pri\_key, [62](#)
  - receiver\_pri\_key\_sz, [62](#)
  - receiver\_pub\_key, [62](#)
  - receiver\_pub\_key\_sz, [62](#)
  - sender\_tmp\_pri\_key, [62](#)
  - sender\_tmp\_pri\_key\_sz, [62](#)
  - shared\_info1, [63](#)
  - shared\_info1\_bytes, [63](#)
  - shared\_info2, [63](#)
  - shared\_info2\_bytes, [63](#)
- ehsm\_SM9\_exchg\_key\_role\_e
  - eHSM\_Com\_Struct\_lp.h, [431](#)
- ehsm\_add\_cmd\_to\_priority\_queue
  - eHSM\_Dspt\_CryObj\_lp.c, [496](#)
  - eHSM\_Dspt\_CryObj\_lp.h, [500](#)
- ehsm\_add\_cmd\_to\_sent\_queue
  - eHSM\_Dspt\_CryObj\_lp.c, [496](#)
  - eHSM\_Dspt\_CryObj\_lp.h, [500](#)
- ehsm\_addr\_t
  - eHSM\_Types\_lp.h, [922](#)
- ehsm\_aead\_data\_ptr, [63](#)
  - input\_data, [64](#)
  - output\_data, [64](#)
- ehsm\_aead\_data\_ptr\_st
  - eHSM\_Mgr\_Ctx\_lp.h, [880](#)
- ehsm\_aead\_request
  - eHSM\_If\_Asr\_Cipher\_lp.c, [545](#)
  - eHSM\_If\_Asr\_Cipher\_lp.h, [548](#)
- ehsm\_api\_type\_e
  - eHSM\_Com\_Struct\_lp.h, [426](#)
- ehsm\_asym\_alg\_e
  - eHSM\_Mgr\_Ctx\_lp.h, [880](#)
- ehsm\_autosar\_convert\_ret\_code
  - eHSM\_If\_Asr\_ErrCode\_lp.c, [551](#)
  - eHSM\_If\_Asr\_ErrCode\_lp.h, [552](#)
- ehsm\_bitmap\_clr
  - eHSM\_Compt\_Bitmap.c, [433](#)
  - eHSM\_Compt\_Bitmap.h, [435](#)
- ehsm\_bitmap\_count
  - eHSM\_Compt\_Bitmap.c, [433](#)
  - eHSM\_Compt\_Bitmap.h, [436](#)
- ehsm\_bitmap\_create
  - eHSM\_Compt\_Bitmap.h, [436](#)
- ehsm\_bitmap\_destroy
  - eHSM\_Compt\_Bitmap.h, [436](#)
- ehsm\_bitmap\_first
  - eHSM\_Compt\_Bitmap.c, [433](#)
  - eHSM\_Compt\_Bitmap.h, [436](#)
- ehsm\_bitmap\_init
  - eHSM\_Compt\_Bitmap.c, [433](#)
  - eHSM\_Compt\_Bitmap.h, [437](#)
- ehsm\_bitmap\_last
  - eHSM\_Compt\_Bitmap.c, [434](#)
  - eHSM\_Compt\_Bitmap.h, [437](#)
- ehsm\_bitmap\_reset
  - eHSM\_Compt\_Bitmap.c, [434](#)
  - eHSM\_Compt\_Bitmap.h, [437](#)
- ehsm\_bitmap\_set
  - eHSM\_Compt\_Bitmap.c, [434](#)
  - eHSM\_Compt\_Bitmap.h, [437](#)
- ehsm\_bool\_t
  - eHSM\_Types\_lp.h, [922](#)
- ehsm\_calcpubval\_with\_job
  - eHSM\_If\_Asr\_Key\_lp.h, [569](#)
- ehsm\_calcsecret\_with\_job
  - eHSM\_If\_Asr\_Key\_lp.h, [569](#)
- ehsm\_cancel\_single\_service
  - eHSM\_Srv\_Mgr\_lp.c, [905](#)
  - eHSM\_Srv\_Mgr\_lp.h, [917](#)
- ehsm\_certificate\_parse
  - eHSM\_If\_Asr\_lp.h, [553](#)
- ehsm\_certificate\_verify
  - eHSM\_If\_Asr\_lp.h, [554](#)
- ehsm\_certificate\_verify\_st, [64](#)
  - certificate\_info, [65](#)
  - verify, [65](#)
- ehsm\_challenge\_type\_e
  - eHSM\_Com\_Struct\_lp.h, [426](#)
- ehsm\_change\_control\_field\_cmd, [65](#)
  - reserved, [65](#)
  - size, [65](#)
  - type, [66](#)
  - value\_addr, [66](#)
- ehsm\_change\_control\_field\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, [859](#)
- ehsm\_change\_control\_field\_st, [66](#)
  - rev, [66](#)
  - size, [67](#)
  - type, [67](#)
  - value, [67](#)
- ehsm\_change\_controlfield
  - eHSM\_If\_Ext\_lp.h, [715](#)

- eHSM\_If\_Ext\_SysMgr\_lp.c, 740
- ehsm\_change\_lifecycle
  - eHSM\_If\_Ext\_lp.h, 716
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 741
- ehsm\_change\_lifecycle\_cmd, 67
  - type, 68
- ehsm\_change\_lifecycle\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 859
- ehsm\_cipher\_ctx\_data\_check
  - eHSM\_Mgr\_Ctx\_lp.c, 877
  - eHSM\_Mgr\_Ctx\_lp.h, 881
- ehsm\_close\_debug
  - eHSM\_If\_Ext\_lp.h, 716
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 741
- ehsm\_close\_debug\_cmd, 68
  - type, 68
- ehsm\_close\_debug\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 860
- ehsm\_cmd, 68
  - ehsm\_mailbox\_req, 198
  - ehsm\_mbox\_mgr\_channel\_req, 206
  - output\_size, 69
  - req\_cipher, 69
- ehsm\_cmd\_aead\_ptr\_st, 69
  - aad\_ptr, 70
  - data\_ptr, 70
  - tag\_ptr, 70
- ehsm\_cmd\_cipher\_st, 70
  - cmd\_id, 71
  - context\_addr, 71
  - context\_size, 71
  - input\_addr, 71
  - input\_size, 71
  - key\_addr, 71
  - key\_handle, 72
  - key\_size, 72
  - output\_addr, 72
  - output\_size, 72
  - rev1, 72
  - sec\_input\_addr, 72
  - sec\_input\_size, 73
  - u\_hdr, 73
- ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, 354
  - hdr\_eccp\_keygen, 354
  - hdr\_ecise, 355
  - hdr\_pke, 355
  - hdr\_rng, 355
  - hdr\_rsa\_keygen, 355
  - hdr\_ske, 355
  - hdr\_sm9, 355
- ehsm\_cmd\_cipher\_with\_rps\_st
  - eHSM\_Srv\_Cipher\_lp.h, 885
- ehsm\_cmd\_ext\_type\_e
  - eHSM\_Srv\_Mgr\_lp.h, 916
- ehsm\_cmd\_hdr\_eccp\_keygen\_st, 73
  - curve\_id, 73
  - hdr\_rev1, 74
  - type, 74
- ehsm\_cmd\_hdr\_ecise\_st, 74
  - cipher\_alg, 75
  - curve\_id, 75
  - direction, 75
  - hdr\_rev1, 75
  - hdr\_rev2, 75
  - kdf\_alg, 75
  - mac\_alg, 75
  - mac\_k\_byte, 76
- ehsm\_cmd\_hdr\_pke\_st, 76
  - algorithm, 76
  - direction, 76
  - hdr\_rev1, 77
  - padding, 77
  - process\_mode, 77
  - rsa crt\_mode, 77
  - time\_stamp, 77
- ehsm\_cmd\_hdr\_rng\_st, 78
  - algorithm, 78
  - hdr\_rev1, 78
  - hdr\_rev2, 78
- ehsm\_cmd\_hdr\_rsa\_keygen\_st, 79
  - e\_bit\_size, 79
  - hdr\_rev1, 79
  - is crt, 79
  - n\_bit\_size, 79
  - type, 80
- ehsm\_cmd\_hdr\_ske\_st, 80
  - algorithm, 80
  - cipher\_mode, 81
  - direction, 81
  - key\_type, 81
  - padding, 81
  - process\_mode, 81
  - tag\_size, 81
  - time\_stamp, 82
- ehsm\_cmd\_hdr\_sm9\_st, 82
  - direction, 82
  - enc\_type, 82
  - hdr\_rev1, 83
  - hdr\_rev2, 83
  - hid, 83
  - key2\_size, 83
  - padding, 83
- ehsm\_cmd\_req, 84
  - api\_type, 84
  - channel, 84
  - cmd\_data, 84
  - cmd\_id, 85
  - cmd\_size, 85
  - cmd\_state, 85
  - error\_code, 85
  - list, 85
  - object\_type, 85
  - priority, 86
  - release\_cb, 86
  - req\_cb, 86
  - req\_ctx, 86
  - req\_type, 86
  - rps\_data, 86
  - timeout, 87
- ehsm\_cmd\_req\_buffer, 87
  - bitmap, 87
  - cmd\_req, 87

- ehsm\_cmd\_req\_buffer\_st
  - eHSM\_Srv\_Mgr\_Ip.c, [905](#)
- ehsm\_cmd\_req\_st
  - eHSM\_Srv\_CmdReq\_Ip.h, [887](#)
- ehsm\_cmd\_req\_state\_e
  - eHSM\_Srv\_CmdReq\_Ip.h, [887](#)
- ehsm\_cmd\_req\_type\_e
  - eHSM\_Srv\_CmdReq\_Ip.h, [887](#)
- ehsm\_cmd\_sm9\_sig\_vry\_output\_ptr\_st, [88](#)
  - h, [88](#)
  - Sig, [88](#)
- ehsm\_code\_upgrade\_alg\_e
  - eHSM\_Com\_Struct\_Ip.h, [426](#)
- ehsm\_code\_verify\_alg\_e
  - eHSM\_Com\_Struct\_Ip.h, [427](#)
- ehsm\_control\_field\_type\_e
  - eHSM\_Com\_Struct\_Ip.h, [427](#)
- ehsm\_copy\_key\_cmd, [88](#)
  - key\_auth\_size, [89](#)
  - key\_auth\_value, [89](#)
  - key\_handle, [89](#)
  - key\_usage, [89](#)
  - key\_usage\_size, [89](#)
  - reserved1, [90](#)
- ehsm\_copy\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [860](#)
- ehsm\_counter\_value\_st
  - eHSM\_If\_Evita\_Types\_Ip.h, [702](#)
- ehsm\_create\_dh\_key\_cmd, [90](#)
  - algorithm, [90](#)
  - dh\_mode, [91](#)
  - key\_size, [91](#)
  - key\_usage, [91](#)
  - key\_usage\_size, [91](#)
  - local\_key\_auth\_size, [91](#)
  - local\_key\_auth\_value, [91](#)
  - local\_key\_handle, [92](#)
  - parent\_alg, [92](#)
  - remote\_key\_auth\_size, [92](#)
  - remote\_key\_auth\_value, [92](#)
  - remote\_key\_handle, [92](#)
  - reserved2, [92](#)
  - reserved3, [93](#)
  - sm2\_ext\_param, [93](#)
  - ss\_addr, [93](#)
  - type, [93](#)
  - valid\_until, [93](#)
- ehsm\_create\_dh\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [860](#)
- ehsm\_create\_dh\_key\_param, [94](#)
  - dh\_mode, [94](#)
  - key\_element\_data, [94](#)
  - key\_element\_size, [94](#)
  - key\_handle, [95](#)
  - key\_size, [95](#)
  - local\_key\_auth\_size, [95](#)
  - local\_key\_auth\_value, [95](#)
  - local\_key\_handle, [95](#)
  - parent\_alg, [95](#)
  - remote\_key\_auth\_or\_pub\_key\_size, [96](#)
  - remote\_key\_auth\_value\_or\_pub\_key, [96](#)
  - remote\_key\_handle, [96](#)
  - sm2\_ext\_param, [96](#)
  - ss\_addr, [96](#)
  - target\_algorithm\_identifier, [96](#)
  - type, [97](#)
- ehsm\_create\_dh\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [421](#)
- ehsm\_create\_dh\_sm2\_ext\_param, [97](#)
  - local\_tmp\_key\_auth\_size, [97](#)
  - local\_tmp\_key\_auth\_value, [97](#)
  - local\_tmp\_key\_handle, [98](#)
  - peer\_temp\_pubkey, [98](#)
  - reserved2, [98](#)
  - s1\_s2\_value, [98](#)
  - sa\_sb\_value, [98](#)
  - sm2\_role, [98](#)
- ehsm\_create\_dh\_sm2\_ext\_param\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [860](#)
- ehsm\_create\_evita\_key\_param, [99](#)
  - gen\_key\_param, [99](#)
  - key\_element\_data, [99](#)
  - key\_element\_size, [99](#)
  - key\_handle, [100](#)
  - key\_size, [100](#)
  - type, [100](#)
- ehsm\_create\_evita\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [421](#)
- ehsm\_create\_random\_key\_param, [100](#)
  - g, [101](#)
  - g\_size, [101](#)
  - key\_element\_data, [101](#)
  - key\_element\_size, [101](#)
  - key\_handle, [101](#)
  - key\_size, [101](#)
  - p, [102](#)
  - p\_size, [102](#)
  - q, [102](#)
  - q\_size, [102](#)
  - target\_algorithm\_identifier, [102](#)
  - type, [102](#)
- ehsm\_create\_random\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [421](#)
- ehsm\_crypto\_key, [103](#)
  - crypto\_key, [103](#)
  - ehsm\_key\_type, [103](#)
  - valid, [103](#)
- ehsm\_crypto\_key\_st
  - eHSM\_If\_Asr\_Types\_Ip.h, [617](#)
- ehsm\_crypto\_object\_get\_cmd\_done
  - eHSM\_Dspt\_CryObj\_Ip.c, [496](#)
  - eHSM\_Dspt\_CryObj\_Ip.h, [500](#)
- ehsm\_crypto\_object\_get\_job
  - eHSM\_Dspt\_CryObj\_Ip.c, [497](#)
  - eHSM\_Dspt\_CryObj\_Ip.h, [500](#)
- ehsm\_crypto\_object\_init
  - eHSM\_Dspt\_CryObj\_Ip.c, [497](#)
  - eHSM\_Dspt\_CryObj\_Ip.h, [500](#)
- ehsm\_crypto\_object\_is\_free
  - eHSM\_Dspt\_CryObj\_Ip.c, [497](#)
  - eHSM\_Dspt\_CryObj\_Ip.h, [501](#)
- ehsm\_crypto\_object\_submit\_cmd



- eHSM\_Dspt\_CryObj\_lp.h, 501
- ehsm\_crypto\_randomgenerate\_param, 104
  - algorithm, 104
  - random\_data\_addr, 104
  - request\_size, 104
- ehsm\_crypto\_randomgenerate\_param\_st
  - eHSM\_Com\_Struct\_lp.h, 421
- ehsm\_ctx\_block\_mgr, 105
  - block\_buf, 105
  - block\_sz, 105
  - remain\_data\_sz, 105
- ehsm\_ctx\_block\_mgr\_st
  - eHSM\_Mgr\_Ctx\_lp.h, 880
- ehsm\_ctx\_session\_st, 106
  - block\_sz, 106
  - max\_chunk\_size, 106
  - session\_id, 106
- ehsm\_debug\_auth
  - eHSM\_If\_Ext\_lp.h, 717
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 742
- ehsm\_debug\_auth\_alg\_e
  - eHSM\_Com\_Struct\_lp.h, 427
- ehsm\_debug\_auth\_st, 107
  - alg, 107
  - public\_key, 107
  - public\_key\_size, 107
  - signature, 108
  - signature\_size, 108
  - type, 108
- ehsm\_debug\_authentication\_cmd, 108
  - algrithm, 109
  - pub\_addr, 109
  - pub\_size, 109
  - rev1, 109
  - rev2, 109
  - sign\_addr, 109
  - sign\_size, 110
  - type, 110
- ehsm\_debug\_authentication\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 860
- ehsm\_del\_cmd\_from\_priority\_queue
  - eHSM\_Dspt\_CryObj\_lp.c, 497
  - eHSM\_Dspt\_CryObj\_lp.h, 501
- ehsm\_del\_cmd\_from\_sent\_queue
  - eHSM\_Dspt\_CryObj\_lp.c, 498
  - eHSM\_Dspt\_CryObj\_lp.h, 501
- ehsm\_derive\_key\_cmd, 110
  - derive\_type, 111
  - itera\_times, 111
  - key\_deriv\_func, 111
  - key\_size, 111
  - key\_usage, 111
  - key\_usage\_size, 111
  - parent\_key\_auth\_size, 112
  - parent\_key\_auth\_value, 112
  - parent\_key\_handle, 112
  - pw\_data, 112
  - pw\_size, 112
  - reserved1, 112
  - reserved2, 113
  - salt\_data, 113
  - salt\_size, 113
  - type, 113
  - valid\_until, 113
- ehsm\_derive\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 860
- ehsm\_dh\_mode\_e
  - eHSM\_Com\_Struct\_lp.h, 428
- ehsm\_dh\_param, 114
  - g, 114
  - p, 114
  - q, 114
- ehsm\_dh\_param\_size\_info, 114
  - g\_size, 115
  - p\_size, 115
  - q\_size, 115
- ehsm\_dh\_param\_size\_info\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 702
- ehsm\_dh\_param\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 702
- ehsm\_dh\_prikey, 115
  - priv, 116
- ehsm\_dh\_prikey\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 703
- ehsm\_dh\_pubkey, 116
  - dh\_param, 116
  - pub, 116
- ehsm\_dh\_pubkey\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 703
- ehsm\_disjunction\_finish
  - eHSM\_Mgr\_Ctx\_lp.c, 877
  - eHSM\_Mgr\_Ctx\_lp.h, 881
- ehsm\_disjunction\_updata
  - eHSM\_Mgr\_Ctx\_lp.c, 877
  - eHSM\_Mgr\_Ctx\_lp.h, 881
- ehsm\_dispatcher\_init
  - eHSM\_Dspt\_lp.c, 503
  - eHSM\_Dspt\_lp.h, 505
- ehsm\_dst\_addr
  - ehsm\_otp\_write\_cmd, 214
- ehsm\_ecc\_key\_size, 117
  - priv\_key\_size, 117
  - pub\_key\_size, 117
- ehsm\_ecc\_key\_size\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 703
- ehsm\_ecc\_pubkey, 118
  - p, 118
- ehsm\_ecc\_pubkey\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 703
- ehsm\_emu\_status\_st
  - eHSM\_Com\_Struct\_lp.h, 421
- ehsm\_emu\_status\_st, 118
  - o\_hsm\_err\_fw, 118
  - o\_hsm\_err\_hw, 119
  - o\_hsm\_err\_sensor, 119
  - o\_hsm\_status, 119
- ehsm\_encrypt\_key
  - eHSM\_If\_Ext\_lp.h, 717
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 743
- ehsm\_encrypt\_request
  - eHSM\_If\_Asr\_Cipher\_lp.c, 545
  - eHSM\_If\_Asr\_Cipher\_lp.h, 548

- ehsm\_evita\_convert\_ret\_code
  - eHSM\_If\_Evita\_ErrCode\_lp.c, 624
  - eHSM\_If\_Evita\_ErrCode\_lp.h, 626
- ehsm\_evita\_key\_export, 119
  - authenticity\_key\_author\_size, 120
  - authenticity\_key\_author\_value, 120
  - authenticity\_key\_handle, 120
  - encrypted\_key, 120
  - encrypted\_key\_size, 120
  - key\_auth\_code, 120
  - key\_auth\_code\_size, 121
  - key\_handle, 121
  - transport\_key\_author\_size, 121
  - transport\_key\_author\_value, 121
  - transport\_key\_handle, 121
  - use\_flags, 121
- ehsm\_evita\_key\_export\_st
  - eHSM\_Com\_Struct\_lp.h, 422
- ehsm\_evita\_key\_handle\_e
  - eHSM\_If\_Evita\_Types\_lp.h, 705
- ehsm\_evita\_key\_import\_st, 122
  - authenticity\_key\_author\_size, 122
  - authenticity\_key\_author\_value, 122
  - authenticity\_key\_handle, 123
  - encrypted\_key, 123
  - encrypted\_key\_size, 123
  - key\_auth\_code, 123
  - key\_auth\_code\_size, 123
  - key\_handle, 123
  - transport\_key\_author\_size, 124
  - transport\_key\_author\_value, 124
  - transport\_key\_handle, 124
  - type, 124
- ehsm\_evita\_memory\_info\_st
  - eHSM\_Com\_Struct\_lp.h, 422
- ehsm\_evita\_memory\_info\_st\_, 124
  - nvm\_free\_size, 125
  - nvm\_total\_size, 125
  - ram\_free\_size, 125
  - ram\_total\_size, 125
- ehsm\_exchange\_sm9\_key\_param, 126
  - fp12g, 126
  - key\_handle, 126
  - key\_size, 126
  - kgc\_pub\_key, 127
  - peer\_id, 127
  - peer\_id\_size, 127
  - peer\_tmp\_pub, 127
  - role, 127
  - s1\_s2, 127
  - sa\_sb, 128
  - self\_id, 128
  - self\_id\_size, 128
  - type, 128
  - user\_priv\_key\_handle, 128
  - user\_tmp\_key\_handle, 128
- ehsm\_exchange\_sm9\_key\_param\_st
  - eHSM\_Com\_Struct\_lp.h, 422
- ehsm\_export\_key\_cmd, 129
  - authenticity\_key\_auth\_size, 129
  - authenticity\_key\_auth\_value, 129
  - authenticity\_key\_handle, 130
  - encrypted\_key, 130
  - encrypted\_key\_size, 130
  - key\_auth\_size, 130
  - key\_auth\_value, 130
  - key\_handle, 130
  - transport\_key\_auth\_size, 131
  - transport\_key\_auth\_value, 131
  - transport\_key\_handle, 131
  - use\_flags, 131
- ehsm\_export\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 860
- ehsm\_export\_pub\_key\_, 131
  - algo\_id, 132
  - dh\_pubkey\_bytes\_size, 132
  - key, 132
  - rsa\_e\_bytes\_size, 132
  - size\_info, 132
- ehsm\_external\_key\_, 133
  - auth\_sign\_data, 133
  - evita\_internal\_key, 133
- ehsm\_fast\_cmac\_st, 133
  - algorithm, 134
  - direction, 134
  - in, 134
  - key\_auth\_size, 134
  - key\_auth\_value, 135
  - key\_handle, 135
  - key\_type, 135
  - mac, 135
  - size, 135
- ehsm\_fetch\_cmd\_from\_crypto\_object
  - eHSM\_Dspt\_CryObj\_lp.c, 498
  - eHSM\_Dspt\_CryObj\_lp.h, 502
- ehsm\_fw\_encrypt\_key, 136
  - key\_data, 136
  - key\_size, 136
  - key\_slot, 136
  - key\_type, 136
- ehsm\_fw\_encrypt\_key\_cmd, 137
  - input\_addr, 137
  - input\_size, 137
  - key\_slot, 137
  - key\_type, 138
  - rev1, 138
  - rev2, 138
  - rev3, 138
- ehsm\_fw\_encrypt\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 860
- ehsm\_fw\_encrypt\_key\_slot\_e
  - eHSM\_Com\_Struct\_lp.h, 428
- ehsm\_fw\_encrypt\_key\_st
  - eHSM\_Com\_Struct\_lp.h, 422
- ehsm\_fw\_encrypt\_key\_type\_e
  - eHSM\_Com\_Struct\_lp.h, 428
- ehsm\_fw\_get\_random\_key\_cmd, 138
  - key\_slot, 139
  - key\_type, 139
  - reserved1, 139
- ehsm\_fw\_get\_random\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 861



- ehsm\_fw\_random\_key, 139
  - key\_slot, 140
  - key\_type, 140
- ehsm\_fw\_random\_key\_slot\_e
  - eHSM\_Com\_Struct\_Ip.h, 429
- ehsm\_fw\_random\_key\_st
  - eHSM\_Com\_Struct\_Ip.h, 422
- ehsm\_fw\_random\_key\_type\_e
  - eHSM\_Com\_Struct\_Ip.h, 429
- ehsm\_gen\_dh\_key\_param\_st, 140
  - g, 140
  - g\_size, 141
  - p, 141
  - p\_size, 141
  - q, 141
  - q\_size, 141
- ehsm\_gen\_key
  - ehsm\_mailbox\_req, 198
- ehsm\_gen\_key\_alg\_e
  - eHSM\_If\_Ext\_Types\_Ip.h, 754
- ehsm\_gen\_key\_cmd, 142
  - alg\_or\_crt, 142
  - e\_size, 142
  - g, 142
  - g\_size, 143
  - key\_usage, 143
  - key\_usage\_size, 143
  - n\_size, 143
  - p, 143
  - p\_size, 143
  - q, 144
  - q\_size, 144
  - reserved1, 144
  - reserved2, 144
  - reserved3, 144
  - type, 144
  - valid\_until, 145
- ehsm\_gen\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_gen\_key\_param\_st, 145
  - algo\_id, 145
  - dh\_key\_param, 145
  - param, 146
  - rsa\_e\_bit\_size, 146
- ehsm\_gen\_sm9\_key\_param, 146
  - hid, 146
  - key\_handle, 147
  - key\_param, 147
  - master\_key, 147
  - priv\_key, 147
  - rev, 147
  - sm9\_key\_type, 147
- ehsm\_gen\_sm9\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, 422
- ehsm\_gen\_sm9\_key\_type\_e
  - eHSM\_Com\_Struct\_Ip.h, 429
- ehsm\_gen\_sm9\_master\_key\_param, 148
  - master\_key\_type, 148
- ehsm\_gen\_sm9\_master\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, 422
- ehsm\_gen\_sm9\_userpriv\_key
  - ehsm\_mailbox\_req, 198
- ehsm\_gen\_sm9\_userpriv\_key\_cmd, 148
  - id\_addr, 149
  - id\_size, 149
  - rev1, 149
  - rev\_key\_auth, 149
  - rev\_key\_auth\_size, 149
  - rev\_key\_handle, 150
  - type, 150
- ehsm\_gen\_sm9\_userpriv\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_gen\_sm9\_userpriv\_key\_param, 150
  - kgc\_pubkey, 150
  - priv\_key\_type, 151
  - type, 151
  - user\_id\_size, 151
  - user\_id\_value, 151
  - with\_pubkey, 151
- ehsm\_gen\_sm9\_userpriv\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, 422
- ehsm\_get\_challenge
  - eHSM\_If\_Ext\_Ip.h, 718
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 743
- ehsm\_get\_challenge\_cmd, 152
  - output\_addr, 152
  - reserved1, 152
  - type, 152
- ehsm\_get\_challenge\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_get\_challenge\_st, 152
  - buf, 153
  - size, 153
  - type, 153
- ehsm\_get\_crypto\_driver\_object
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 572
  - eHSM\_If\_Asr\_KeyCfg\_Ip.h, 602
- ehsm\_get\_crypto\_ke\_info
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 572
  - eHSM\_If\_Asr\_KeyCfg\_Ip.h, 602
- ehsm\_get\_ehsm\_key\_type
  - eHSM\_If\_Asr\_Key\_Ip.h, 570
- ehsm\_get\_emu\_cmd, 153
  - emu\_addr, 154
  - emu\_size, 154
  - rev1, 154
  - rev2, 154
  - rev3, 154
  - rev\_key\_auth\_addr, 155
  - rev\_key\_auth\_size, 155
  - rev\_key\_handle, 155
- ehsm\_get\_emu\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_get\_emu\_status
  - eHSM\_If\_Ext\_Ip.h, 719
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 744
- ehsm\_get\_emu\_status\_param\_st, 155
  - emu\_addr, 156
  - emu\_size, 156
- ehsm\_get\_pub\_from\_priv
  - eHSM\_If\_Evita\_Ip.h, 646
  - eHSM\_If\_Evita\_Key\_Ip.c, 669

- ehsm\_get\_pub\_from\_priv\_cmd, 156
  - key\_alg\_id, 156
  - key\_auth\_size, 157
  - key\_auth\_value, 157
  - key\_handle, 157
  - public\_key\_addr, 157
  - public\_key\_buffer\_size, 157
  - reserved1, 157
  - reserved2, 158
- ehsm\_get\_pub\_from\_priv\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_get\_pub\_from\_priv\_param, 158
  - key\_alg\_id, 158
  - key\_auth\_size, 158
  - key\_auth\_value, 159
  - key\_handle, 159
  - public\_key\_addr, 159
  - public\_key\_buffer\_size, 159
  - public\_key\_size, 159
- ehsm\_get\_pub\_from\_priv\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, 423
- ehsm\_get\_random\_key
  - eHSM\_If\_Ext\_Ip.h, 719
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 744
- ehsm\_get\_she\_id\_cmd, 160
  - challenge\_addr, 160
  - challenge\_size, 160
  - reserved, 160
  - signatrue\_addr, 160
  - signatrue\_size, 161
  - status\_addr, 161
  - status\_size, 161
- ehsm\_get\_she\_id\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_get\_tick
  - eHSM\_Mailbox\_Ip.h, 848
- ehsm\_get\_version
  - eHSM\_If\_Asr\_Ip.h, 555
- ehsm\_handle\_t
  - eHSM\_Types\_Ip.h, 922
- ehsm\_hash\_hmac
  - eHSM\_If\_Asr\_Cipher\_Ip.c, 546
  - eHSM\_If\_Asr\_Cipher\_Ip.h, 549
- ehsm\_hsm\_fw\_upgrade\_finish
  - eHSM\_If\_Ext\_Ip.h, 720
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 745
- ehsm\_hsm\_fw\_upgrade\_init
  - eHSM\_If\_Ext\_Ip.h, 721
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 746
- ehsm\_hsm\_fw\_upgrade\_update
  - eHSM\_If\_Ext\_Ip.h, 721
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 746
- ehsm\_hsm\_fw\_upgrade\_verify\_finish
  - eHSM\_If\_Ext\_Ip.h, 722
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 747
- ehsm\_hsm\_fw\_upgrade\_verify\_init
  - eHSM\_If\_Ext\_Ip.h, 723
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 748
- ehsm\_hsm\_fw\_upgrade\_verify\_update
  - eHSM\_If\_Ext\_Ip.h, 723
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 748
- ehsm\_image, 161
  - ctx, 162
  - ctx\_size, 162
  - image, 162
  - image\_size, 162
  - process\_mode, 162
  - storage, 162
  - storage\_size, 163
- ehsm\_image\_process\_mode\_e
  - eHSM\_Com\_Struct\_Ip.h, 429
- ehsm\_image\_upgrade\_cmd, 163
  - ctx\_addr, 163
  - ctx\_size, 164
  - image\_addr, 164
  - image\_size, 164
  - process\_mode, 164
  - rev1, 164
  - rev2, 164
  - rev3, 165
  - rev4, 165
  - rev\_key\_auth, 165
  - rev\_key\_auth\_size, 165
  - rev\_key\_handle, 165
  - storage\_addr, 165
- ehsm\_image\_upgrade\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 861
- ehsm\_image\_upgrade\_st
  - eHSM\_Com\_Struct\_Ip.h, 423
- ehsm\_image\_verfiy\_cmd, 166
  - ctx\_addr, 166
  - ctx\_size, 166
  - image\_addr, 167
  - image\_size, 167
  - process\_mode, 167
  - rev1, 167
  - rev2, 167
  - rev3, 167
  - rev4, 168
  - rev5, 168
  - rev\_key\_auth, 168
  - rev\_key\_auth\_size, 168
  - rev\_key\_handle, 168
  - type, 168
- ehsm\_image\_verify\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 862
- ehsm\_image\_verify\_st, 169
  - ctx, 169
  - ctx\_size, 169
  - image, 169
  - image\_size, 170
  - process\_mode, 170
  - storage, 170
  - storage\_size, 170
  - type, 170
- ehsm\_import\_key\_cmd, 171
  - authenticity\_key\_auth\_size, 171
  - authenticity\_key\_auth\_value, 171
  - authenticity\_key\_handle, 171
  - encrypted\_key, 171
  - encrypted\_key\_size, 172
  - key\_auth\_size, 172

- key\_auth\_value, 172
- key\_type, 172
- rev1, 172
- transport\_key\_auth\_size, 172
- transport\_key\_auth\_value, 173
- transport\_key\_handle, 173
- ehsm\_import\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 862
- ehsm\_init
  - eHSM\_If\_Asr\_lp.h, 555
  - eHSM\_If\_Asr\_Job\_lp.c, 566
- ehsm\_init\_block\_mgr
  - eHSM\_Mgr\_Ctx\_lp.c, 877
  - eHSM\_Mgr\_Ctx\_lp.h, 881
- ehsm\_int16\_t
  - eHSM\_Types\_lp.h, 922
- ehsm\_int32\_t
  - eHSM\_Types\_lp.h, 922
- ehsm\_int64\_t
  - eHSM\_Types\_lp.h, 922
- ehsm\_int8\_t
  - eHSM\_Types\_lp.h, 922
- ehsm\_internal\_key\_, 173
  - attr, 173
  - key\_signatrue, 174
  - key\_usage, 174
  - prikey, 174
  - prikey\_enc\_size, 174
  - pubkey, 174
- ehsm\_is\_cmd\_addr\_null
  - eHSM\_Mailbox\_lp.c, 846
  - eHSM\_Mailbox\_lp.h, 848
- ehsm\_job\_cancel
  - eHSM\_If\_Asr\_lp.h, 556
  - eHSM\_If\_Asr\_Job\_lp.c, 566
- ehsm\_job\_submit
  - eHSM\_If\_Asr\_lp.h, 556
  - eHSM\_If\_Asr\_Job\_lp.c, 567
- ehsm\_key\_attr\_data\_, 175
  - algo\_id, 175
  - key\_identifier, 175
  - key\_info, 175
  - key\_signatrue\_off, 175
  - key\_usage\_size, 176
  - valid\_util, 176
- ehsm\_key\_attr\_data\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 703
- ehsm\_key\_copy
  - eHSM\_If\_Asr\_lp.h, 557
- ehsm\_key\_copy\_param, 176
  - key\_auth\_size, 176
  - key\_auth\_value, 177
  - key\_element\_data, 177
  - key\_element\_size, 177
  - parent\_key\_handle, 177
  - target\_key\_handle, 177
- ehsm\_key\_copy\_param\_st
  - eHSM\_Com\_Struct\_lp.h, 423
- ehsm\_key\_derive
  - eHSM\_If\_Asr\_lp.h, 558
- ehsm\_key\_derived\_param, 178
  - derive\_type, 178
  - itera\_times, 178
  - key\_deriv\_func, 178
  - key\_element\_data, 179
  - key\_element\_size, 179
  - key\_handle, 179
  - key\_size, 179
  - parent\_key\_author\_size, 179
  - parent\_key\_author\_value, 179
  - parent\_key\_handle, 180
  - passwd, 180
  - passwd\_size, 180
  - salt\_data, 180
  - salt\_size, 180
  - type, 180
- ehsm\_key\_derived\_param\_st
  - eHSM\_Com\_Struct\_lp.h, 423
- ehsm\_key\_element\_copy
  - eHSM\_If\_Asr\_lp.h, 558
- ehsm\_key\_element\_copy\_partial
  - eHSM\_If\_Asr\_lp.h, 559
- ehsm\_key\_element\_get
  - eHSM\_If\_Asr\_lp.h, 560
- ehsm\_key\_element\_ids\_get
  - eHSM\_If\_Asr\_lp.h, 561
- ehsm\_key\_element\_set
  - eHSM\_If\_Asr\_lp.h, 562
- ehsm\_key\_element\_type\_get\_ex
  - eHSM\_If\_Asr\_Key\_lp.h, 570
- ehsm\_key\_exchange\_caclpubval
  - eHSM\_If\_Asr\_lp.h, 562
- ehsm\_key\_exchange\_calcsecret
  - eHSM\_If\_Asr\_lp.h, 563
- ehsm\_key\_flags\_element\_st, 181
  - auth\_flag, 181
  - auth\_size, 181
  - auth\_value, 181
  - auth\_value\_exist\_flags, 182
  - trnsp\_flags, 182
  - use\_flags, 182
- ehsm\_key\_generate
  - eHSM\_If\_Asr\_lp.h, 564
- ehsm\_key\_is\_valid
  - eHSM\_If\_Asr\_Key\_lp.h, 570
- ehsm\_key\_mem\_type\_e
  - eHSM\_Com\_Struct\_lp.h, 430
- ehsm\_key\_mgr\_init
  - eHSM\_If\_Asr\_Key\_lp.h, 570
- ehsm\_key\_remove\_cmd, 182
  - key\_auth\_size, 183
  - key\_auth\_value, 183
  - key\_handle, 183
  - reserved1, 183
- ehsm\_key\_remove\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 862
- ehsm\_key\_remove\_param, 183
  - key\_auth\_size, 184
  - key\_auth\_value, 184
  - key\_handle, 184
- ehsm\_key\_remove\_param\_st
  - eHSM\_Com\_Struct\_lp.h, 423

- ehsm\_key\_set\_valid
  - eHSM\_If\_Asr\_Ip.h, 564
- ehsm\_key\_signatrue\_st
  - eHSM\_If\_Evita\_Types\_Ip.h, 703
- ehsm\_key\_signature\_, 184
  - sign\_id, 185
  - sign\_info, 185
  - sign\_key\_id, 185
  - signatrue, 185
  - target\_key\_id, 185
- ehsm\_key\_status\_, 186
  - activeUseFlag, 186
  - algo\_id, 186
  - cert\_data, 186
  - cert\_size, 186
  - key\_sign\_data, 187
  - keyId, 187
  - keyIdSize, 187
  - mem\_location, 187
  - pubkey, 187
  - valid\_util, 187
- ehsm\_key\_status\_cmd, 188
  - cert\_key\_auth\_size, 188
  - cert\_key\_auth\_value, 188
  - cert\_key\_handle, 188
  - key\_handle, 189
  - key\_status, 189
  - key\_status\_size, 189
  - reserved1, 189
  - reserved2, 189
- ehsm\_key\_status\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 862
- ehsm\_key\_status\_param, 190
  - certification\_key\_auth\_size, 190
  - certification\_key\_auth\_value, 190
  - certification\_key\_handle, 190
  - key\_handle, 190
  - key\_status, 191
  - key\_status\_buffer\_size, 191
  - key\_status\_size, 191
- ehsm\_key\_status\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, 423
- ehsm\_key\_type
  - ehsm\_crypto\_key, 103
- ehsm\_key\_type\_e
  - eHSM\_If\_Asr\_Types\_Ip.h, 618
- ehsm\_key\_usages\_st, 191
  - decrypt, 192
  - dhkey, 192
  - encrypt, 192
  - remove, 192
  - secureboot, 192
  - securestorage, 192
  - sign, 193
  - timestamp, 193
  - transport, 193
  - utcsync, 193
  - verify, 193
- ehsm\_key\_use\_state\_e
  - eHSM\_If\_Asr\_Types\_Ip.h, 618
- ehsm\_keyderi\_with\_job
  - eHSM\_If\_Asr\_Key\_Ip.h, 570
- ehsm\_keyexchange\_key\_info, 194
  - key\_auth\_size, 194
  - key\_auth\_value, 194
  - key\_handle, 194
- ehsm\_keyexchange\_key\_info\_st
  - eHSM\_Com\_Struct\_Ip.h, 423
- ehsm\_keyexport\_with\_job
  - eHSM\_If\_Asr\_Key\_Ip.h, 570
- ehsm\_keygen\_with\_job
  - eHSM\_If\_Asr\_Key\_Ip.h, 571
- ehsm\_keyimport\_with\_job
  - eHSM\_If\_Asr\_Key\_Ip.h, 571
- ehsm\_keyremove\_with\_job
  - eHSM\_If\_Asr\_Key\_Ip.h, 571
- ehsm\_keysetvalid\_with\_job
  - eHSM\_If\_Asr\_Key\_Ip.h, 571
- ehsm\_lifecycle\_e
  - eHSM\_Com\_Struct\_Ip.h, 430
- ehsm\_low\_power
  - eHSM\_If\_Ext\_Ip.h, 724
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 749
- ehsm\_low\_power\_cmd, 194
  - power\_mode, 195
  - reserved, 195
- ehsm\_low\_power\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 862
- ehsm\_mac\_request
  - eHSM\_If\_Asr\_Cipher\_Ip.c, 546
  - eHSM\_If\_Asr\_Cipher\_Ip.h, 549
- ehsm\_mailbox\_req, 195
  - api\_type, 196
  - close\_debug, 197
  - cmd\_id, 197
  - copy\_key, 197
  - create\_dh\_key, 197
  - debug\_authentication, 197
  - derive\_key, 197
  - ehsm\_cmd, 198
  - ehsm\_gen\_key, 198
  - ehsm\_gen\_sm9\_userpriv\_key, 198
  - export\_key, 198
  - fw\_encrypt\_key, 198
  - fw\_random\_key, 198
  - gen\_usertmp, 199
  - get\_challenge, 199
  - get\_emu, 199
  - get\_mast\_pubkey, 199
  - get\_pub\_from\_priv, 199
  - get\_she\_id, 199
  - get\_tmp\_pubkey, 200
  - image\_upgrade, 200
  - image\_verify, 200
  - import\_key, 200
  - key\_remove, 200
  - key\_status, 200
  - module\_status, 201
  - otp\_read, 201
  - otp\_write, 201
  - rev, 201
  - she\_load\_export\_key, 201

- she\_load\_plain\_key, 201
- sm9\_exchg\_key, 202
- sm9\_export\_key, 202
- sm9\_import\_key, 202
- sm9\_remove\_key, 202
- sm9\_unwrap\_key, 202
- sm9\_wrap\_key, 202
- soc\_boot, 203
- soc\_image\_verify, 203
- uart\_cmd, 203
- ehsm\_mailbox\_req\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 862
- ehsm\_mbox\_cancel\_channel\_req, 203
  - api\_type, 204
  - cancel\_type, 204
  - cmd\_tag, 204
  - rev, 204
- ehsm\_mbox\_cancel\_channel\_req\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 862
- ehsm\_mbox\_cancel\_channel\_rps, 204
  - api\_type, 205
  - cancel\_type, 205
  - cmd\_tag, 205
  - ret\_code, 205
  - rev, 205
- ehsm\_mbox\_cancel\_channel\_rps\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 862
- ehsm\_mbox\_init
  - eHSM\_Mailbox\_Ip.c, 846
  - eHSM\_Mailbox\_Ip.h, 848
- ehsm\_mbox\_mgr\_channel\_req, 206
  - change\_control\_field\_cmd, 206
  - change\_lifecycle\_cmd, 206
  - cmd\_id, 206
  - ehsm\_cmd, 206
  - low\_power\_cmd, 207
  - self\_test\_cmd, 207
  - sensor\_resp\_init\_cmd, 207
  - set\_baudrate\_cmd, 207
- ehsm\_mbox\_mgr\_channel\_req\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 863
- ehsm\_mbox\_polling
  - eHSM\_Dspt\_Ip.c, 503
  - eHSM\_Mailbox\_Ip.c, 846
- ehsm\_mbox\_send\_cmd
  - eHSM\_Mailbox\_Ip.c, 846
  - eHSM\_Mailbox\_Ip.h, 848
- ehsm\_mgr\_ctx\_find
  - eHSM\_Mgr\_Ctx\_Ip.c, 878
  - eHSM\_Mgr\_Ctx\_Ip.h, 882
- ehsm\_mgr\_ctx\_free
  - eHSM\_Mgr\_Ctx\_Ip.c, 878
  - eHSM\_Mgr\_Ctx\_Ip.h, 882
- ehsm\_mgr\_ctx\_get\_free
  - eHSM\_Mgr\_Ctx\_Ip.c, 879
  - eHSM\_Mgr\_Ctx\_Ip.h, 882
- ehsm\_module\_status\_cmd, 207
  - algo\_id, 208
  - key\_auth\_addr, 208
  - key\_auth\_size, 208
  - key\_handle, 208
  - reserved, 208
  - signatrue, 209
  - signatrue\_size, 209
  - status\_addr, 209
  - status\_size, 209
  - type, 209
- ehsm\_module\_status\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 863
- ehsm\_module\_status\_st, 210
  - algo\_id, 210
  - key\_auth\_size, 210
  - key\_auth\_value, 210
  - key\_handle, 211
  - sign, 211
  - sign\_size, 211
  - status, 211
  - status\_size, 211
  - type, 211
- ehsm\_otp\_read\_cmd, 212
  - ehsm\_src\_addr, 212
  - host\_dst\_addr, 212
  - size, 212
- ehsm\_otp\_read\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 863
- ehsm\_otp\_read\_param\_st, 213
  - flash\_read\_addr, 213
  - otp\_data\_addr, 213
  - read\_data\_size, 213
- ehsm\_otp\_write\_cmd, 214
  - ehsm\_dst\_addr, 214
  - host\_src\_addr, 214
  - size, 214
- ehsm\_otp\_write\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 863
- ehsm\_otp\_write\_param\_st, 215
  - flash\_write\_addr, 215
  - otp\_data\_addr, 215
  - write\_data\_size, 215
- ehsm\_power\_mode\_e
  - eHSM\_If\_Ext\_Ip.h, 715
- ehsm\_prikey\_data\_, 216
  - dh, 216
  - dh\_k, 216
  - ecc\_k, 216
  - kdf\_k, 217
  - rsa\_crt, 217
  - rsa\_d, 217
  - sym\_k, 217
- ehsm\_prikey\_data\_st
  - eHSM\_If\_Evita\_Types\_Ip.h, 703
- ehsm\_printf\_ke\_size
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 572
  - eHSM\_If\_Asr\_KeyCfg\_Ip.h, 602
- ehsm\_process\_asr\_service
  - eHSM\_Srv\_Mgr\_Ip.c, 906
  - eHSM\_Srv\_Mgr\_Ip.h, 917
- ehsm\_process\_sync\_service
  - eHSM\_Srv\_Mgr\_Ip.c, 906
  - eHSM\_Srv\_Mgr\_Ip.h, 918
- ehsm\_pub\_key\_, 217
  - algo\_id, 218

- key\_pub\_data, 218
- key\_size\_info, 218
- ehsm\_pubkey\_data\_, 218
  - dh, 219
  - ecc, 219
  - rsa, 219
- ehsm\_pubkey\_data\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 703
- ehsm\_random\_generate
  - eHSM\_If\_Asr\_Cipher\_lp.c, 547
  - eHSM\_If\_Asr\_Cipher\_lp.h, 550
- ehsm\_read\_otp\_data
  - eHSM\_If\_Ext\_lp.h, 725
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 750
- ehsm\_register\_addr.h, 883
- ehsm\_register\_service
  - eHSM\_Srv\_Mgr\_lp.c, 906
  - eHSM\_Srv\_Mgr\_lp.h, 918
- ehsm\_remove\_cmd\_from\_queue
  - eHSM\_Dspt\_lp.c, 503
  - eHSM\_Dspt\_lp.h, 505
- ehsm\_remove\_key\_extend
  - eHSM\_If\_Asr\_lp.h, 565
- ehsm\_rng\_generate\_cmd, 219
  - algorithm, 220
  - random\_data\_addr, 220
  - request\_size, 220
  - rev1, 220
  - rev2, 220
  - rev3, 221
  - rev4, 221
  - rev\_key\_auth\_size, 221
  - rev\_key\_auth\_value, 221
  - rev\_key\_handle, 221
- ehsm\_rng\_generate\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 863
- ehsm\_rsa crt\_param\_, 222
  - dp, 222
  - dq, 222
  - p, 222
  - q, 222
  - u, 223
- ehsm\_rsa\_ctr\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 704
- ehsm\_rsa\_dh\_key\_size\_, 223
  - reserved, 223
  - rsa\_dh\_g\_size, 223
  - rsa\_dh\_p\_size, 224
  - rsa\_dh\_q\_size, 224
- ehsm\_rsa\_dh\_key\_size\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 704
- ehsm\_rsa\_key\_size\_, 224
  - reserved, 224
  - rsa\_d\_size, 225
  - rsa\_e\_size, 225
  - rsa\_n\_size, 225
- ehsm\_rsa\_key\_size\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 704
- ehsm\_rsa\_key\_type\_e
  - eHSM\_Com\_Struct\_lp.h, 430
- ehsm\_rsa\_pubkey, 225
  - e, 226
  - n, 226
- ehsm\_rsa\_pubkey\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 704
- ehsm\_se\_key\_, 226
  - algo\_id, 226
  - auth\_size, 227
  - auth\_value, 227
  - key\_data, 227
  - key\_handle, 227
  - key\_size\_info, 227
  - reserved, 227
- ehsm\_secure\_boot
  - eHSM\_If\_Ext\_lp.h, 725
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 750
- ehsm\_secure\_boot\_st, 228
  - encrypt\_iv\_addr, 228
  - encrypt\_iv\_size, 228
  - header\_addr, 229
  - header\_size, 229
  - image\_addr, 229
  - image\_size, 229
  - need\_encryption, 229
  - pubkey\_addr, 229
  - pubkey\_size, 230
  - rev1, 230
  - sign\_addr, 230
  - sign\_size, 230
  - storage\_alg, 230
  - type, 230
- ehsm\_self\_test
  - eHSM\_If\_Evita\_lp.h, 647
  - eHSM\_If\_Ext\_SysMgr\_lp.c, 751
- ehsm\_self\_test\_cmd, 231
  - test\_type, 231
- ehsm\_self\_test\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 863
- ehsm\_sensor\_init\_param\_st, 231
  - data, 232
  - size, 232
- ehsm\_sensor\_resp\_init\_cmd, 232
  - data\_addr, 232
  - data\_size, 233
- ehsm\_sensor\_resp\_init\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_lp.h, 863
- ehsm\_service, 233
  - reqhdl, 233
  - rsphdl, 233
  - service\_id, 234
  - timeout, 234
- ehsm\_service\_info, 234
  - api\_type, 234
  - cb, 235
  - priority, 235
  - req\_type, 235
  - service\_ctx, 235
- ehsm\_service\_info\_st
  - eHSM\_Srv\_Mgr\_lp.h, 915
- ehsm\_service\_init
  - eHSM\_Srv\_Mgr\_lp.c, 906
  - eHSM\_Srv\_Mgr\_lp.h, 918

- ehsm\_service\_st
  - eHSM\_Srv\_Mgr\_Ip.h, [915](#)
- ehsm\_set\_address\_pointer
  - eHSM\_Srv\_Mgr\_Ip.c, [906](#)
  - eHSM\_Srv\_Mgr\_Ip.h, [918](#)
- ehsm\_set\_baudrate\_cmd, [235](#)
  - baud\_div, [236](#)
- ehsm\_set\_baudrate\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [863](#)
- ehsm\_set\_uart\_baudrate
  - eHSM\_If\_Ext\_Ip.h, [726](#)
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, [751](#)
- ehsm\_she\_cancel
  - eHSM\_If\_She\_Ip.c, [758](#)
  - eHSM\_If\_She\_Ip.h, [772](#)
- ehsm\_she\_convert\_ret\_code
  - eHSM\_If\_She\_ErrCode\_Ip.c, [755](#)
  - eHSM\_If\_She\_ErrCode\_Ip.h, [756](#)
- ehsm\_she\_get\_id\_param\_st, [236](#)
  - challenge, [236](#)
  - challenge\_size, [236](#)
  - signatrue, [237](#)
  - signatrue\_size, [237](#)
  - status, [237](#)
  - status\_size, [237](#)
- ehsm\_she\_key\_handle\_e
  - eHSM\_If\_She\_Types\_Ip.h, [790](#)
- ehsm\_she\_key\_host\_param, [237](#)
  - m1, [238](#)
  - m2, [238](#)
  - m3, [238](#)
  - m4, [238](#)
  - m5, [238](#)
  - she\_ext\_flag, [239](#)
- ehsm\_she\_key\_host\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [423](#)
- ehsm\_she\_key\_param, [239](#)
  - m1, [239](#)
  - m2, [239](#)
  - m3, [240](#)
  - m4, [240](#)
  - m5, [240](#)
  - she\_ext\_flag, [240](#)
- ehsm\_she\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_she\_key\_st, [240](#)
  - counter, [241](#)
  - raw\_key, [241](#)
  - reserved, [241](#)
  - secure\_flag, [241](#)
- ehsm\_she\_load\_export\_key\_cmd, [242](#)
  - m1, [242](#)
  - m2, [242](#)
  - m3, [242](#)
  - m4, [242](#)
  - m5, [243](#)
  - she\_ext\_flag, [243](#)
- ehsm\_she\_load\_export\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_she\_load\_plain\_key\_cmd, [243](#)
  - key\_data, [243](#)
- ehsm\_she\_load\_plain\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_she\_plain\_key\_host\_param, [244](#)
  - key\_data, [244](#)
- ehsm\_she\_plain\_key\_host\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_she\_plain\_key\_param, [244](#)
  - key\_data, [245](#)
- ehsm\_she\_plain\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_she\_status\_type\_
  - eHSM\_If\_She\_Types\_Ip.h, [791](#)
- ehsm\_she\_status\_type\_e
  - eHSM\_If\_She\_Types\_Ip.h, [790](#)
- ehsm\_signature\_gen\_vry
  - eHSM\_If\_Asr\_Cipher\_Ip.c, [547](#)
  - eHSM\_If\_Asr\_Cipher\_Ip.h, [550](#)
- ehsm\_sm9\_cipher
  - eHSM\_If\_Ext\_Ip.h, [726](#)
- ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, [245](#)
  - kgc\_public\_key, [245](#)
  - peer\_id, [245](#)
  - peer\_id\_size, [246](#)
  - rev1, [246](#)
  - rev\_key\_auth\_size, [246](#)
  - rev\_key\_handle, [246](#)
  - type, [246](#)
- ehsm\_sm9\_exchg\_gen\_usertmp\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_sm9\_exchg\_key
  - eHSM\_If\_Ext\_Ip.h, [727](#)
- ehsm\_sm9\_exchg\_key\_cmd, [247](#)
  - info\_stuct, [247](#)
  - key\_size, [247](#)
  - rev1, [247](#)
  - rev2, [247](#)
  - rev3, [248](#)
  - rev4, [248](#)
  - role, [248](#)
  - type, [248](#)
  - user\_priv\_key\_handle, [248](#)
  - user\_tmp\_key\_handle, [248](#)
- ehsm\_sm9\_exchg\_key\_cmd\_child, [249](#)
  - fp12g, [249](#)
  - kgc\_pub\_key, [249](#)
  - peer\_id, [249](#)
  - peer\_id\_size, [250](#)
  - peer\_tmp\_pub, [250](#)
  - s1\_s2, [250](#)
  - sa\_sb, [250](#)
  - self\_id, [250](#)
  - self\_id\_size, [250](#)
- ehsm\_sm9\_exchg\_key\_cmd\_child\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_sm9\_exchg\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_sm9\_exckey\_gen\_tmpkey
  - eHSM\_If\_Ext\_Ip.h, [728](#)
- ehsm\_sm9\_exckey\_gen\_tmpkey\_param, [251](#)
  - key\_handle, [251](#)
  - kgc\_pub\_key, [251](#)



- peer\_id, [251](#)
- peer\_id\_size, [252](#)
- priv\_key\_type, [252](#)
- type, [252](#)
- ehsm\_sm9\_exckey\_gen\_tmpkey\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_sm9\_export\_key
  - eHSM\_If\_Ext\_Ip.h, [729](#)
- ehsm\_sm9\_export\_key\_cmd, [252](#)
  - authenticated\_key, [253](#)
  - authenticated\_key\_size, [253](#)
  - encrypted\_key, [253](#)
  - encrypted\_key\_size, [253](#)
  - key\_handle, [253](#)
  - rev1, [253](#)
  - rev\_key\_auth, [254](#)
  - rev\_key\_auth\_size, [254](#)
- ehsm\_sm9\_export\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_sm9\_gen\_mast\_pubkey, [254](#)
  - key\_type, [254](#)
  - pub\_key, [255](#)
- ehsm\_sm9\_gen\_mast\_pubkey\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_sm9\_gen\_mastpubkey\_from\_mastprivkey
  - eHSM\_If\_Ext\_Ip.h, [730](#)
- ehsm\_sm9\_gen\_tmp\_pubkey\_param, [255](#)
  - id\_size, [255](#)
  - key\_handle, [255](#)
  - pub\_key, [256](#)
  - user\_id, [256](#)
- ehsm\_sm9\_gen\_tmp\_pubkey\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_sm9\_gen\_tmppubkey\_from\_tmpprivkey
  - eHSM\_If\_Ext\_Ip.h, [730](#)
- ehsm\_sm9\_generate\_master\_key
  - eHSM\_If\_Ext\_Ip.h, [732](#)
- ehsm\_sm9\_generate\_priv\_key
  - eHSM\_If\_Ext\_Ip.h, [732](#)
- ehsm\_sm9\_get\_mast\_pubkey\_cmd, [256](#)
  - master\_key\_type, [256](#)
  - public\_key\_addr, [257](#)
  - reserved2, [257](#)
- ehsm\_sm9\_get\_mast\_pubkey\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_sm9\_get\_tmp\_pubkey\_cmd, [257](#)
  - key\_handle, [257](#)
  - public\_key\_addr, [258](#)
  - reserved1, [258](#)
  - reserved2, [258](#)
  - user\_id, [258](#)
  - user\_id\_size, [258](#)
- ehsm\_sm9\_get\_tmp\_pubkey\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [864](#)
- ehsm\_sm9\_import\_key
  - eHSM\_If\_Ext\_Ip.h, [733](#)
- ehsm\_sm9\_import\_key\_cmd, [259](#)
  - authenticated\_key, [259](#)
  - authenticated\_key\_size, [259](#)
  - encrypted\_key, [259](#)
  - encrypted\_key\_size, [259](#)
  - key\_is\_plain, [260](#)
  - rev1, [260](#)
  - rev\_key\_auth, [260](#)
  - rev\_key\_auth\_size, [260](#)
  - rev\_key\_handle, [260](#)
  - type, [260](#)
- ehsm\_sm9\_import\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [865](#)
- ehsm\_sm9\_inexport\_key\_param, [261](#)
  - key\_auth\_size, [261](#)
  - key\_auth\_value, [261](#)
  - key\_blob, [261](#)
  - key\_blob\_size, [262](#)
  - key\_handle, [262](#)
  - key\_is\_plain, [262](#)
  - type, [262](#)
- ehsm\_sm9\_inexport\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_sm9\_master\_key\_type\_e
  - eHSM\_Com\_Struct\_Ip.h, [431](#)
- ehsm\_sm9\_remove\_key
  - eHSM\_If\_Ext\_Ip.h, [734](#)
- ehsm\_sm9\_remove\_key\_cmd, [262](#)
  - key\_handle, [263](#)
  - reserved1, [263](#)
- ehsm\_sm9\_remove\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [865](#)
- ehsm\_sm9\_sign
  - eHSM\_If\_Ext\_Ip.h, [734](#)
- ehsm\_sm9\_unwrap\_key
  - eHSM\_If\_Ext\_Ip.h, [735](#)
- ehsm\_sm9\_unwrap\_key\_cmd, [263](#)
  - cipher\_addr, [264](#)
  - cipher\_size, [264](#)
  - id\_addr, [264](#)
  - id\_size, [264](#)
  - key\_addr, [264](#)
  - key\_size, [264](#)
  - rev1, [265](#)
  - rev\_key\_auth, [265](#)
  - rev\_key\_auth\_size, [265](#)
  - user\_priv\_key\_handle, [265](#)
- ehsm\_sm9\_unwrap\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [865](#)
- ehsm\_sm9\_unwrap\_key\_param, [265](#)
  - cipher\_addr, [266](#)
  - cipher\_size, [266](#)
  - id\_addr, [266](#)
  - id\_size, [266](#)
  - key\_addr, [266](#)
  - key\_size, [267](#)
  - user\_priv\_key\_handle, [267](#)
- ehsm\_sm9\_unwrap\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [424](#)
- ehsm\_sm9\_user\_privkey\_type\_e
  - eHSM\_Com\_Struct\_Ip.h, [431](#)
- ehsm\_sm9\_wrap\_key
  - eHSM\_If\_Ext\_Ip.h, [736](#)
- ehsm\_sm9\_wrap\_key\_cmd, [267](#)
  - fp12g, [268](#)
  - hid, [268](#)



- id\_addr, [268](#)
- id\_size, [268](#)
- key\_addr, [268](#)
- key\_size, [268](#)
- pub\_key, [268](#)
- rev1, [269](#)
- rev2, [269](#)
- rev3, [269](#)
- rev4, [269](#)
- rev\_key\_auth\_size, [269](#)
- rev\_key\_handle, [269](#)
- ehsm\_sm9\_wrap\_key\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [865](#)
- ehsm\_sm9\_wrap\_key\_param, [270](#)
  - fp12g, [270](#)
  - hid, [270](#)
  - id\_size, [270](#)
  - key\_addr, [271](#)
  - key\_size, [271](#)
  - pub\_key, [271](#)
  - user\_id, [271](#)
- ehsm\_sm9\_wrap\_key\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [425](#)
- ehsm\_soc\_image\_upgrade\_info\_st
  - eHSM\_Com\_Struct\_Ip.h, [425](#)
- ehsm\_soc\_image\_upgrade\_input\_st
  - eHSM\_Com\_Struct\_Ip.h, [425](#)
- ehsm\_soc\_image\_verify\_cmd, [271](#)
  - header\_addr, [272](#)
  - header\_size, [272](#)
  - pubkey\_addr, [272](#)
  - pubkey\_size, [272](#)
  - resered1, [273](#)
  - resered2, [273](#)
  - resered3, [273](#)
  - storage\_alg, [273](#)
  - storage\_encryption\_flag, [273](#)
  - storage\_image\_addr, [273](#)
  - storage\_image\_size, [274](#)
  - storage\_iv\_addr, [274](#)
  - storage\_iv\_size, [274](#)
  - storage\_sign\_addr, [274](#)
  - storage\_sign\_size, [274](#)
  - type, [274](#)
- ehsm\_soc\_image\_verify\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [865](#)
- ehsm\_soc\_image\_verify\_info\_st
  - eHSM\_Com\_Struct\_Ip.h, [425](#)
- ehsm\_soc\_image\_verify\_input\_st
  - eHSM\_Com\_Struct\_Ip.h, [425](#)
- ehsm\_soc\_image\_verify\_st, [275](#)
  - encrypt\_iv\_addr, [275](#)
  - encrypt\_iv\_size, [275](#)
  - header\_addr, [276](#)
  - header\_size, [276](#)
  - image\_addr, [276](#)
  - image\_size, [276](#)
  - need\_encryption, [276](#)
  - pubkey\_addr, [276](#)
  - pubkey\_size, [277](#)
  - rev1, [277](#)
  - sign\_addr, [277](#)
  - sign\_size, [277](#)
  - storage\_alg, [277](#)
  - type, [277](#)
- ehsm\_soc\_secure\_boot\_status\_st, [278](#)
  - status, [278](#)
- ehsm\_src\_addr
  - ehsm\_otp\_read\_cmd, [212](#)
- ehsm\_storage\_area\_param\_st, [278](#)
  - addr, [279](#)
  - size, [279](#)
- ehsm\_submit\_cmd\_req
  - eHSM\_Dspt\_Ip.c, [503](#)
  - eHSM\_Dspt\_Ip.h, [505](#)
- ehsm\_sym\_key\_size\_, [279](#)
  - key\_size, [279](#)
  - reserved, [280](#)
- ehsm\_sym\_key\_size\_st
  - eHSM\_If\_Evita\_Types\_Ip.h, [704](#)
- ehsm\_tick\_form\_ms
  - eHSM\_Mailbox\_Ip.h, [848](#)
- ehsm\_tick\_value, [280](#)
  - current\_ticks, [280](#)
  - tick\_accuracy, [280](#)
  - tick\_length, [281](#)
- ehsm\_tick\_value\_st
  - eHSM\_If\_Evita\_Types\_Ip.h, [704](#)
- ehsm\_uart\_baudrate\_e
  - eHSM\_Com\_Struct\_Ip.h, [432](#)
- ehsm\_uart\_cmd, [281](#)
  - uart\_cmd\_buffer, [281](#)
- ehsm\_uart\_cmd\_st
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [865](#)
- ehsm\_uint16\_t
  - eHSM\_Types\_Ip.h, [923](#)
- ehsm\_uint32\_t
  - eHSM\_Types\_Ip.h, [923](#)
- ehsm\_uint64\_t
  - eHSM\_Types\_Ip.h, [923](#)
- ehsm\_uint8\_t
  - eHSM\_Types\_Ip.h, [923](#)
- ehsm\_utc\_time\_t
  - eHSM\_If\_Evita\_Types\_Ip.h, [704](#)
- ehsm\_write\_otp\_data
  - eHSM\_If\_Ext\_Ip.h, [737](#)
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, [752](#)
- element\_data
  - crypto\_copy\_key\_info, [25](#)
  - crypto\_create\_evita\_key\_info, [26](#)
- element\_info
  - CryptoKeyType, [57](#)
- element\_size
  - crypto\_copy\_key\_info, [26](#)
  - crypto\_create\_evita\_key\_info, [27](#)
- emu\_addr
  - ehsm\_get\_emu\_cmd, [154](#)
  - ehsm\_get\_emu\_status\_param\_st, [156](#)
- emu\_size
  - ehsm\_get\_emu\_cmd, [154](#)
  - ehsm\_get\_emu\_status\_param\_st, [156](#)
- enc\_typde

- sm9cipher\_testvec, 365
- enc\_type
  - ehsm\_cmd\_hdr\_sm9\_st, 82
- encrypt
  - ehsm\_key\_usages\_st, 192
  - HSM\_KeyActUseFlagsType, 312
  - HSM\_KeyUsagesType, 320
  - key\_act\_use\_flags\_t, 344
- encrypt\_iv\_addr
  - ehsm\_secure\_boot\_st, 228
  - ehsm\_soc\_image\_verify\_st, 275
- encrypt\_iv\_size
  - ehsm\_secure\_boot\_st, 228
  - ehsm\_soc\_image\_verify\_st, 275
- encrypted\_key
  - crypto\_exported\_key, 29
  - crypto\_import\_evita\_key\_info, 31
  - ehsm\_evita\_key\_export, 120
  - ehsm\_evita\_key\_import\_st, 123
  - ehsm\_export\_key\_cmd, 130
  - ehsm\_import\_key\_cmd, 171
  - ehsm\_sm9\_export\_key\_cmd, 253
  - ehsm\_sm9\_import\_key\_cmd, 259
- encrypted\_key\_buffer\_size
  - crypto\_exported\_key, 29
- encrypted\_key\_size
  - crypto\_exported\_key, 29
  - crypto\_import\_evita\_key\_info, 31
  - ehsm\_evita\_key\_export, 120
  - ehsm\_evita\_key\_import\_st, 123
  - ehsm\_export\_key\_cmd, 130
  - ehsm\_import\_key\_cmd, 172
  - ehsm\_sm9\_export\_key\_cmd, 253
  - ehsm\_sm9\_import\_key\_cmd, 259
- error\_code
  - ehsm\_cmd\_req, 85
- Evita\_Check\_Authorization\_Code
  - eHSM\_If\_Evita\_ErrCode\_Ip.c, 625
  - eHSM\_If\_Evita\_ErrCode\_Ip.h, 627
- Evita\_Check\_Key\_Handle
  - eHSM\_If\_Evita\_ErrCode\_Ip.c, 625
  - eHSM\_If\_Evita\_ErrCode\_Ip.h, 627
- evita\_internal\_key
  - eHSM\_If\_Evita\_Types\_Ip.h, 709
  - ehsm\_external\_key, 133
- exchange\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 574
- Exclusive\_area\_enter
  - eHSM\_Exclusive\_Area.h, 543
- Exclusive\_area\_exit
  - eHSM\_Exclusive\_Area.h, 544
- expected\_a\_public
  - kpp\_testvec, 349
- expected\_a\_public\_size
  - kpp\_testvec, 349
- expected\_ss
  - kpp\_testvec, 349
- expected\_ss\_size
  - kpp\_testvec, 349
- export\_key
  - ehsm\_mailbox\_req, 198
- export\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 574
- ExtParam
  - HSM\_PlainKeyCfgType, 322
- false
  - eHSM\_Types\_Ip.h, 921
- family
  - Crypto\_AlgorithmInfoType, 23
- filename
  - eHSM\_Debug\_Ip.h, 492
- fips\_skip
  - cipher\_testvec, 20
- flash\_read\_addr
  - ehsm\_otp\_read\_param\_st, 213
- flash\_write\_addr
  - ehsm\_otp\_write\_param\_st, 215
- fp12g
  - ehsm\_exchange\_sm9\_key\_param, 126
  - ehsm\_sm9\_exchg\_key\_cmd\_child, 249
  - ehsm\_sm9\_wrap\_key\_cmd, 268
  - ehsm\_sm9\_wrap\_key\_param, 270
- fw\_encrypt\_key
  - ehsm\_mailbox\_req, 198
- fw\_random\_key
  - ehsm\_mailbox\_req, 198
- g
  - crypto\_copy\_key\_dh\_key\_info, 24
  - ehsm\_create\_random\_key\_param, 101
  - ehsm\_dh\_param, 114
  - ehsm\_gen\_dh\_key\_param\_st, 140
  - ehsm\_gen\_key\_cmd, 142
  - HSM\_DhParamType, 296
  - kpp\_testvec, 350
- g\_cmd\_req\_buffer
  - eHSM\_Srv\_Mgr\_Ip.c, 907
- g\_ctx
  - eHSM\_If\_Ext\_SysMgr\_Ip.c, 753
- g\_size
  - crypto\_copy\_key\_dh\_key\_info, 24
  - ehsm\_create\_random\_key\_param, 101
  - ehsm\_dh\_param\_size\_info, 115
  - ehsm\_gen\_dh\_key\_param\_st, 141
  - ehsm\_gen\_key\_cmd, 143
  - kpp\_testvec, 350
- g\_srv\_soc\_image\_verify
  - eHSM\_Srv\_Ext\_Ip.c, 890
  - eHSM\_Srv\_Mgr\_Ip.c, 907
- gen\_key\_param
  - ehsm\_create\_evita\_key\_param, 99
- gen\_usertmp
  - ehsm\_mailbox\_req, 199
- generate\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 574
- genkey
  - kpp\_testvec, 350
- Get\_Tick\_Count
  - eHSM\_If\_Evita\_Ip.h, 647
- Get\_Time\_Sync\_Challenge
  - eHSM\_If\_Evita\_Ip.h, 648

- Get.UTC\_Time
  - eHSM\_If\_Evita\_lp.h, [648](#)
- get\_challenge
  - ehsm\_mailbox\_req, [199](#)
- get\_emu
  - ehsm\_mailbox\_req, [199](#)
- get\_mast\_pubkey
  - ehsm\_mailbox\_req, [199](#)
- get\_pub\_from\_priv
  - ehsm\_mailbox\_req, [199](#)
- get\_she\_id
  - ehsm\_mailbox\_req, [199](#)
- get\_tmp\_pubkey
  - ehsm\_mailbox\_req, [200](#)
- h
  - ehsm\_cmd\_sm9\_sig\_vry\_output\_ptr\_st, [88](#)
  - sm9cipher\_testvec, [365](#)
- H2S\_SRV\_CMD\_CANCEL\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_lp.h, [853](#)
- H2S\_SRV\_CMD\_CANCEL\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_lp.h, [854](#)
- H2S\_SRV\_CMD\_CANCEL\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, [854](#)
- H2S\_SRV\_CMD\_JTAG\_END\_BIT
  - eHSM\_Mailbox\_Prtcl\_lp.h, [854](#)
- H2S\_SRV\_CMD\_JTAG\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_lp.h, [854](#)
- H2S\_SRV\_GENERAL\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_lp.h, [854](#)
- H2S\_SRV\_GENERAL\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_lp.h, [854](#)
- H2S\_SRV\_GENERAL\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, [855](#)
- H2S\_SRV\_JTAG\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_lp.h, [855](#)
- H2S\_SRV\_JTAG\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, [855](#)
- H2S\_SRV\_MGR\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_lp.h, [855](#)
- H2S\_SRV\_MGR\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_lp.h, [855](#)
- H2S\_SRV\_MGR\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, [855](#)
- h2s\_note\_bit
  - mailbox\_channel, [353](#)
- h2s\_start\_addr
  - mailbox\_channel, [353](#)
- h2s\_word\_size
  - mailbox\_channel, [353](#)
- HASH\_SIZE
  - eHSM\_If\_Evita\_Hash\_lp.c, [629](#)
- HAlgo
  - HSM\_AsymCfgType, [286](#)
  - HSM\_HMacCfgType, [303](#)
- HOST2HSM\_ACCESS\_PATH
  - eHSM\_Mailbox\_Reg\_lp.h, [867](#)
- HOST\_ADDRESS\_SIZE
  - eHSM\_Mailbox\_Prtcl\_lp.h, [856](#)
- HOST\_LOG\_DEBUG
  - eHSM\_Debug\_lp.h, [492](#)
- HOST\_LOG\_ERROR
  - eHSM\_Debug\_lp.h, [492](#)
- HOST\_LOG\_INFO
  - eHSM\_Debug\_lp.h, [493](#)
- HOST\_LOG\_WARN
  - eHSM\_Debug\_lp.h, [493](#)
- HOST\_LOG
  - eHSM\_Debug\_lp.h, [492](#)
- HOST\_PURE\_LOG
  - eHSM\_Debug\_lp.h, [493](#)
- HSM\_AsymAlgoType
  - Hsm\_Hal.h, [961](#)
- HSM\_AsymCfgType, [285](#)
  - AsymAlgo, [285](#)
  - CipherDir, [286](#)
  - HAlgo, [286](#)
  - KeyId, [286](#)
  - Padding, [286](#)
  - SignDir, [286](#)
  - Sync, [287](#)
- HSM\_BootCfgType, [287](#)
  - BootAddr, [288](#)
  - HeaderAddr, [288](#)
  - HeaderSize, [288](#)
  - InitLSram, [288](#)
  - InitUSram, [288](#)
  - LogEnable, [288](#)
  - PIIFreq, [289](#)
  - PubKeyAddr, [289](#)
  - Reservel, [289](#)
  - ResetDisable, [289](#)
  - SignAddr, [289](#)
  - StbWaitHSM, [289](#)
  - StbWaitVerify, [290](#)
  - SwitchClock, [290](#)
  - VerifySize, [290](#)
  - VersionAddr, [290](#)
  - VersionUpdateEn, [290](#)
  - WdgTimeout, [290](#)
- HSM\_CMD\_BOOTROM\_SET\_BAUDRATE
  - AC784xx\_Hsm\_Reg.h, [386](#)
- HSM\_CMD\_ENCRYPT\_KEY
  - AC784xx\_Hsm\_Reg.h, [386](#)
- HSM\_CMD\_GET\_RANDOM\_KEY
  - AC784xx\_Hsm\_Reg.h, [386](#)
- HSM\_CMacCfgType, [291](#)
  - KeyId, [291](#)
  - MacDir, [291](#)
  - SymAlgo, [291](#)
  - Sync, [292](#)
- HSM\_ChallengeType
  - Hsm\_Hal.h, [961](#)
- HSM\_CipherDirection
  - Hsm\_Hal.h, [962](#)
- HSM\_CipherMode
  - Hsm\_Hal.h, [962](#)
- HSM\_DEBUG\_MODE
  - Hsm\_Hal.h, [954](#)
- HSM\_DESTROY\_MODE
  - Hsm\_Hal.h, [954](#)
- HSM\_DEVELOP\_MODE

- Hsm\_Hal.h, [954](#)
- HSM\_DISABLE
  - Hsm\_Hal.h, [954](#)
- HSM\_DebugAuthAlgoType
  - Hsm\_Hal.h, [962](#)
- HSM\_DebugAuthConfigType, [292](#)
  - Alg, [292](#)
  - PubKey, [293](#)
  - PubKeySize, [293](#)
  - Signature, [293](#)
  - SignatureSize, [293](#)
  - Type, [293](#)
- HSM\_DeriveKeyCfgType, [294](#)
  - Kdf, [294](#)
  - KeyId, [294](#)
  - KeySize, [294](#)
  - KeyType, [295](#)
  - KeyUsage, [295](#)
  - KeyUsageSize, [295](#)
  - SaltData, [295](#)
  - SaltDataSize, [295](#)
  - ValidUntil, [295](#)
- HSM\_DhParamType, [296](#)
  - g, [296](#)
  - p, [296](#)
  - q, [296](#)
- HSM\_DhPriKeyType, [297](#)
  - Priv, [297](#)
- HSM\_DhPubKeyType
  - Hsm\_Hal.h, [960](#)
- HSM\_EccPubKeyType, [298](#)
  - p, [299](#)
- HSM\_FLASH\_KEY\_BASE
  - Hsm\_Hal.h, [955](#)
- HSM\_FLASH\_KEY\_MAX\_SLOT\_NUM
  - Hsm\_Hal.h, [955](#)
- HSM\_FLASH\_KEY\_OFFSET
  - Hsm\_Hal.h, [955](#)
- HSM\_FLASH\_KEY\_PAGE\_NUM
  - Hsm\_Hal.h, [955](#)
- HSM\_FLASH\_KEY\_SLOT\_LEN
  - Hsm\_Hal.h, [955](#)
- HSM\_FLASH\_PAGE\_NUM\_OFFSET
  - Hsm\_Hal.h, [956](#)
- HSM\_FLASH\_PAGE\_NUM
  - Hsm\_Hal.h, [955](#)
- HSM\_FLASH\_PAGE\_REVERSE\_LEN
  - Hsm\_Hal.h, [956](#)
- HSM\_FLASH\_PAGE\_VALID\_LEN
  - Hsm\_Hal.h, [956](#)
- HSM\_FlashKeyPageType, [299](#)
  - KeyInfo, [299](#)
  - PageReverse, [299](#)
  - PageValid, [300](#)
- HSM\_FlashKeyType, [300](#)
  - HandleInfo, [300](#)
  - IndexInfo, [300](#)
  - SlotInfo, [301](#)
- HSM\_GenKeyAlgo
  - Hsm\_Hal.h, [963](#)
- HSM\_GenKeyCfgType, [301](#)
  - KeyAlgo, [301](#)
  - KeyId, [301](#)
  - KeySize, [302](#)
  - KeyType, [302](#)
  - KeyUsage, [302](#)
  - KeyUsageSize, [302](#)
  - ValidUntil, [302](#)
- HSM\_HMacCfgType, [303](#)
  - HAlgo, [303](#)
  - KeyId, [303](#)
  - MacDir, [303](#)
  - Sync, [303](#)
- HSM\_Hal\_AesCipher
  - Hsm\_Hal.c, [926](#)
  - Hsm\_Hal.h, [972](#)
- HSM\_Hal\_CipherMac
  - Hsm\_Hal.c, [927](#)
  - Hsm\_Hal.h, [972](#)
- HSM\_Hal\_DebugAuth
  - Hsm\_Hal.c, [927](#)
  - Hsm\_Hal.h, [973](#)
- HSM\_Hal\_Deinit
  - Hsm\_Hal.c, [928](#)
  - Hsm\_Hal.h, [973](#)
- HSM\_Hal\_DeriveKey
  - Hsm\_Hal.c, [928](#)
  - Hsm\_Hal.h, [974](#)
- HSM\_Hal\_DisableSecureBoot
  - Hsm\_Hal.c, [929](#)
  - Hsm\_Hal.h, [974](#)
- HSM\_Hal\_EccSign
  - Hsm\_Hal.c, [929](#)
  - Hsm\_Hal.h, [974](#)
- HSM\_Hal\_EnableSecureBoot
  - Hsm\_Hal.c, [930](#)
  - Hsm\_Hal.h, [975](#)
- HSM\_Hal\_GenerateDHKey
  - Hsm\_Hal.c, [930](#)
  - Hsm\_Hal.h, [975](#)
- HSM\_Hal\_GenerateKey
  - Hsm\_Hal.c, [930](#)
  - Hsm\_Hal.h, [976](#)
- HSM\_Hal\_GetChallenge
  - Hsm\_Hal.c, [931](#)
  - Hsm\_Hal.h, [976](#)
- HSM\_Hal\_GetHsmFwVersion
  - Hsm\_Hal.c, [931](#)
  - Hsm\_Hal.h, [977](#)
- HSM\_Hal\_GetKeyStatus
  - Hsm\_Hal.c, [932](#)
  - Hsm\_Hal.h, [977](#)
- HSM\_Hal\_GetLifeCycle
  - Hsm\_Hal.c, [932](#)
  - Hsm\_Hal.h, [978](#)
- HSM\_Hal\_GetLockState
  - Hsm\_Hal.c, [932](#)
  - Hsm\_Hal.h, [978](#)
- HSM\_Hal\_GetPubKeyFromPrvKey
  - Hsm\_Hal.c, [933](#)
  - Hsm\_Hal.h, [978](#)
- HSM\_Hal\_GetRnd

- Hsm\_Hal.c, [933](#)
- Hsm\_Hal.h, [979](#)
- HSM\_Hal\_GetRndKey
  - Hsm\_Hal.c, [934](#)
  - Hsm\_Hal.h, [979](#)
- HSM\_Hal\_GetSecretkey
  - Hsm\_Hal.c, [934](#)
  - Hsm\_Hal.h, [980](#)
- HSM\_Hal\_Hash
  - Hsm\_Hal.c, [935](#)
  - Hsm\_Hal.h, [980](#)
- HSM\_Hal\_HashMac
  - Hsm\_Hal.c, [936](#)
  - Hsm\_Hal.h, [981](#)
- HSM\_Hal\_HostImageSecureUpgrade
  - Hsm\_Hal.c, [936](#)
  - Hsm\_Hal.h, [981](#)
- HSM\_Hal\_HostImageSecureVerify
  - Hsm\_Hal.c, [936](#)
  - Hsm\_Hal.h, [982](#)
- HSM\_Hal\_Init
  - Hsm\_Hal.c, [937](#)
  - Hsm\_Hal.h, [982](#)
- HSM\_Hal\_InstallCallback
  - Hsm\_Hal.c, [937](#)
  - Hsm\_Hal.h, [983](#)
- HSM\_Hal\_Lock
  - Hsm\_Hal.c, [938](#)
  - Hsm\_Hal.h, [983](#)
- HSM\_Hal\_OtpRead
  - Hsm\_Hal.c, [938](#)
  - Hsm\_Hal.h, [984](#)
- HSM\_Hal\_OtpWrite
  - Hsm\_Hal.c, [939](#)
  - Hsm\_Hal.h, [984](#)
- HSM\_Hal\_RemoveKey
  - Hsm\_Hal.c, [939](#)
  - Hsm\_Hal.h, [985](#)
- HSM\_Hal\_RsaCipher
  - Hsm\_Hal.c, [940](#)
  - Hsm\_Hal.h, [985](#)
- HSM\_Hal\_RsaSign
  - Hsm\_Hal.c, [940](#)
  - Hsm\_Hal.h, [986](#)
- HSM\_Hal\_SetImageSecureUpgradeAlgo
  - Hsm\_Hal.c, [941](#)
  - Hsm\_Hal.h, [986](#)
- HSM\_Hal\_SetImageSecureVerifyAlgo
  - Hsm\_Hal.c, [941](#)
  - Hsm\_Hal.h, [987](#)
- HSM\_Hal\_SetLifeCycle
  - Hsm\_Hal.c, [941](#)
  - Hsm\_Hal.h, [988](#)
- HSM\_Hal\_SetOtpExternalKey
  - Hsm\_Hal.c, [942](#)
  - Hsm\_Hal.h, [988](#)
- HSM\_Hal\_SetOtpKeyCipherAlgo
  - Hsm\_Hal.c, [942](#)
  - Hsm\_Hal.h, [989](#)
- HSM\_Hal\_SetOtpRndKey
  - Hsm\_Hal.c, [943](#)
- Hsm\_Hal.h, [989](#)
- HSM\_Hal\_SetPlainKey
  - Hsm\_Hal.c, [943](#)
  - Hsm\_Hal.h, [990](#)
- HSM\_Hal\_SetSecretKey
  - Hsm\_Hal.c, [944](#)
  - Hsm\_Hal.h, [990](#)
- HSM\_Hal\_SetSecureBootCfg
  - Hsm\_Hal.h, [991](#)
- HSM\_Hal\_Sm2Cipher
  - Hsm\_Hal.c, [944](#)
  - Hsm\_Hal.h, [991](#)
- HSM\_Hal\_Sm2Sign
  - Hsm\_Hal.c, [945](#)
  - Hsm\_Hal.h, [992](#)
- HSM\_Hal\_Sm4Cipher
  - Hsm\_Hal.c, [945](#)
  - Hsm\_Hal.h, [992](#)
- HSM\_Hal\_Unlock
  - Hsm\_Hal.c, [946](#)
  - Hsm\_Hal.h, [993](#)
- HSM\_HashAlgoType
  - Hsm\_Hal.h, [963](#)
- HSM\_ImageType
  - Hsm\_Hal.h, [964](#)
- HSM\_ImageVerifyType
  - Hsm\_Hal.h, [960](#)
- HSM\_ImageVerifyType\_, [304](#)
  - HeaderAddr, [304](#)
  - HeaderSize, [305](#)
  - PubkeyAddr, [305](#)
  - PubkeySize, [305](#)
  - StorageAlg, [305](#)
  - StorageEncryptionFlag, [305](#)
  - StorageImageAddr, [305](#)
  - StorageImageSize, [306](#)
  - StorageIvAddr, [306](#)
  - StorageIvSize, [306](#)
  - StorageSignAddr, [306](#)
  - StorageSignSize, [306](#)
  - Type, [306](#)
  - UpdateVersionFlag, [307](#)
  - VersionAddr, [307](#)
  - VersionSize, [307](#)
- HSM\_InOutMacType, [307](#)
  - BasicInOut, [308](#)
  - MacInBuf, [308](#)
  - MacInBufLen, [308](#)
  - Vry, [308](#)
- HSM\_InOutSignType, [309](#)
  - BasicInOut, [309](#)
  - SignInBuf, [309](#)
  - SignInBufLen, [309](#)
  - Vry, [309](#)
- HSM\_InOutType, [310](#)
  - InBuf, [310](#)
  - InBufLen, [310](#)
  - OutBuf, [310](#)
  - OutBufLen, [311](#)
- HSM\_IrqNum
  - Hsm\_Hal.h, [964](#)

- HSM\_KEY\_DATA\_VALID\_TAG
  - Hsm\_Hal.h, [956](#)
- HSM\_KdfType
  - Hsm\_Hal.h, [965](#)
- HSM\_KeyActUseFlagsType, [311](#)
  - createkey, [312](#)
  - decrypt, [312](#)
  - encrypt, [312](#)
  - remove, [312](#)
  - secureboot, [312](#)
  - securestorage, [312](#)
  - sign, [312](#)
  - timestamp, [313](#)
  - transport, [313](#)
  - utcsync, [313](#)
  - verify, [313](#)
- HSM\_KeyAlgoType
  - Hsm\_Hal.h, [965](#)
- HSM\_KeyFlagsElementType, [313](#)
  - auth\_flag, [314](#)
  - auth\_size, [314](#)
  - auth\_value, [314](#)
  - auth\_value\_exist\_flags, [314](#)
  - trnsp\_flags, [314](#)
  - use\_flags, [315](#)
- HSM\_KeyHandleInfoType, [315](#)
  - AuthSize, [315](#)
  - AuthValue, [315](#)
  - KeyHandle, [316](#)
- HSM\_KeyId
  - Hsm\_Hal.h, [965](#)
- HSM\_KeyIndexInfoType, [316](#)
  - KeyIndex, [316](#)
  - KeyValid, [316](#)
- HSM\_KeySlotInfoType, [317](#)
  - SlotIndex, [317](#)
  - SlotValid, [317](#)
- HSM\_KeyStatusType, [318](#)
  - CertificationAuth, [318](#)
  - CertificationAuthSize, [318](#)
  - CertificationKeyId, [318](#)
  - KeyStatus, [318](#)
  - KeyStatusSize, [319](#)
  - TargetKeyId, [319](#)
- HSM\_KeyStorageType
  - Hsm\_Hal.h, [967](#)
- HSM\_KeyUsagesType, [319](#)
  - decrypt, [320](#)
  - dhkey, [320](#)
  - encrypt, [320](#)
  - remove, [320](#)
  - secureboot, [320](#)
  - securestorage, [320](#)
  - sign, [320](#)
  - timestamp, [321](#)
  - transport, [321](#)
  - utcsync, [321](#)
  - verify, [321](#)
- HSM\_LifeCycleType
  - Hsm\_Hal.h, [968](#)
- HSM\_MANU\_MODE
  - Hsm\_Hal.h, [956](#)
- HSM\_MacDirection
  - Hsm\_Hal.h, [968](#)
- HSM\_OtpCtrlAlgoType
  - Hsm\_Hal.h, [968](#)
- HSM\_OtpKeyCipherAlgo
  - Hsm\_Hal.h, [969](#)
- HSM\_OtpKeyId
  - Hsm\_Hal.h, [969](#)
- HSM\_OtpKeyLevel
  - Hsm\_Hal.h, [969](#)
- HSM\_PAGE\_VALID\_TAG
  - Hsm\_Hal.h, [956](#)
- HSM\_PaddingType
  - Hsm\_Hal.h, [970](#)
- HSM\_PlainKeyCfgType, [321](#)
  - AuthValue, [322](#)
  - AuthValueSize, [322](#)
  - ExtParam, [322](#)
  - KeyAlgo, [323](#)
  - KeyId, [323](#)
  - KeyType, [323](#)
  - KeyUsages, [323](#)
  - KeyUsagesCnt, [323](#)
  - PrivKey, [324](#)
  - PrivKeyLen, [324](#)
  - PubKey, [324](#)
  - PubKeyLen, [324](#)
  - RandomKeySize, [324](#)
  - ValidUtil, [325](#)
- HSM\_PriKeyDataType
  - Hsm\_Hal.h, [960](#)
- HSM\_ProcessMode
  - Hsm\_Hal.h, [970](#)
- HSM\_PubKeyDataType
  - Hsm\_Hal.h, [960](#)
- HSM\_RAM\_KEY\_ECC\_MAX\_NUM
  - Hsm\_Hal.h, [957](#)
- HSM\_RAM\_KEY\_MAX\_NUM
  - Hsm\_Hal.h, [957](#)
- HSM\_RAM\_KEY\_MEM\_SIZE
  - Hsm\_Hal.h, [957](#)
- HSM\_RAM\_KEY\_RSA\_MAX\_NUM
  - Hsm\_Hal.h, [957](#)
- HSM\_RAM\_KEY\_SYM\_MAX\_NUM
  - Hsm\_Hal.h, [957](#)
- HSM\_RamKeyHeadType, [328](#)
  - KeyInfo, [328](#)
  - KeyMemStart, [328](#)
  - KeyMemUsedSize, [328](#)
  - KeyNum, [329](#)
- HSM\_RamKeyInfoType, [329](#)
  - KeyHandleAddr, [329](#)
  - KeyIndex, [329](#)
  - Used, [330](#)
- HSM\_RndAlgo
  - Hsm\_Hal.h, [970](#)
- HSM\_RndOtpKeyType
  - Hsm\_Hal.h, [971](#)
- HSM\_RsaPrvCrtType
  - Hsm\_Hal.h, [961](#)

- HSM\_RsaPubKeyType, 331
  - e, 332
  - n, 332
- HSM\_SECURE\_BOOT\_DISABLE
  - Hsm\_Hal.h, 957
- HSM\_SECURE\_BOOT\_ENABLE
  - Hsm\_Hal.h, 958
- HSM\_SLOT\_INVALID\_TAG
  - Hsm\_Hal.h, 958
- HSM\_STATUS\_IN1
  - eHSM\_Mailbox\_Reg\_Ip.h, 867
- HSM\_STATUS\_IN
  - eHSM\_Mailbox\_Reg\_Ip.h, 867
- HSM\_SUCCESS
  - Hsm\_Hal.h, 958
- HSM\_SecretKeyCfgType, 332
  - AuthVAlueSize, 333
  - AuthValue, 333
  - KeyAlgo, 333
  - KeyType, 333
  - SecretKeyBlob, 333
  - SetKeyBlobSize, 333
  - SetKeyUseFlag, 334
- HSM\_SecureUpgradeType
  - Hsm\_Hal.h, 961
- HSM\_SecureUpgradeType\_, 334
  - CheckVersionFlag, 335
  - CtxAddr, 335
  - CtxSize, 335
  - HeaderAddr, 335
  - HeaderSize, 335
  - MacSignAddr, 335
  - MacSignSize, 336
  - ProcessMode, 336
  - Rev, 336
  - StorageAlg, 336
  - StorageEncryptionFlag, 336
  - StorageImageAddr, 336
  - StorageImageSize, 337
  - StorageIvAddr, 337
  - StorageIvSize, 337
  - UpgradeAlg, 337
  - UpgradeDecryptionFlag, 337
  - UpgradelImageAddr, 337
  - UpgradelImageSize, 338
  - UpgradelvAddr, 338
  - UpgradelvSize, 338
  - UpgradePubkeyAddr, 338
  - UpgradePubkeySize, 338
  - UpgradeSignAddr, 338
  - UpgradeSignSize, 339
  - UpgradeVersionAddr, 339
  - UpgradeVersionSize, 339
- HSM\_SignDirection
  - Hsm\_Hal.h, 971
- HSM\_SymAlgoType
  - Hsm\_Hal.h, 971
- HSM\_SymCfgType, 339
  - CMode, 340
  - CipherDir, 340
  - Iv, 340
  - IvLen, 340
  - KeyId, 340
  - Padding, 341
  - SymAlgo, 341
  - Sync, 341
- HSM\_TEST\_MODE
  - Hsm\_Hal.h, 958
- HSM\_UNNORMAL\_MODE
  - Hsm\_Hal.h, 958
- HSM\_USER\_MODE
  - Hsm\_Hal.h, 958
- HSMMBX\_IRQ\_PRI0
  - eHSM\_Mailbox\_Ip.c, 845
- HandleInfo
  - HSM\_FlashKeyType, 300
- Hash\_CryptoPrimitives
  - eHSM\_IIf\_Asr\_KeyCfg\_Ip.c, 575
- Hash\_Finish
  - eHSM\_IIf\_Evita\_Hash\_Ip.c, 630
  - eHSM\_IIf\_Evita\_Ip.h, 649
- Hash\_Init
  - eHSM\_IIf\_Evita\_Hash\_Ip.c, 630
  - eHSM\_IIf\_Evita\_Ip.h, 649
- Hash\_Update
  - eHSM\_IIf\_Evita\_Hash\_Ip.c, 631
  - eHSM\_IIf\_Evita\_Ip.h, 650
- hash\_alg
  - akcipher\_testvec, 12
  - rsacipher\_testvec, 358
- hash\_hmac
  - hash\_hmac\_t, 282
- hash\_hmac\_size
  - hash\_hmac\_t, 282
- hash\_hmac\_st
  - eHSM\_IIf\_Evita\_Types\_Ip.h, 704
- hash\_hmac\_t, 282
  - hash\_hmac, 282
  - hash\_hmac\_size, 282
  - utc\_time, 282
- hash\_mode\_e
  - eHSM\_IIf\_Evita\_Types\_Ip.h, 706
- hash\_testvec, 282
  - digest, 283
  - digest\_error, 283
  - iv, 283
  - iv\_len, 283
  - key, 284
  - ksize, 284
  - np, 284
  - plaintext, 284
  - psize, 284
  - setkey\_error, 284
  - tap, 285
- hdr\_eccp\_keygen
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, 354
- hdr\_ecise
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, 355
- hdr\_pke
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, 355
- hdr\_rev1
  - ehsm\_cmd\_hdr\_eccp\_keygen\_st, 74



- ehsm\_cmd\_hdr\_ecise\_st, [75](#)
- ehsm\_cmd\_hdr\_pke\_st, [77](#)
- ehsm\_cmd\_hdr\_rng\_st, [78](#)
- ehsm\_cmd\_hdr\_rsa\_keygen\_st, [79](#)
- ehsm\_cmd\_hdr\_sm9\_st, [83](#)
- hdr\_rev2
  - ehsm\_cmd\_hdr\_ecise\_st, [75](#)
  - ehsm\_cmd\_hdr\_rng\_st, [78](#)
  - ehsm\_cmd\_hdr\_sm9\_st, [83](#)
- hdr\_rng
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, [355](#)
- hdr\_rsa\_keygen
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, [355](#)
- hdr\_ske
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, [355](#)
- hdr\_sm9
  - ehsm\_cmd\_cipher\_st::osr\_cmd\_hdr\_u, [355](#)
- header
  - soc\_image\_upgrade\_info, [368](#)
  - soc\_image\_verify\_info, [378](#)
- header\_addr
  - ehsm\_secure\_boot\_st, [229](#)
  - ehsm\_soc\_image\_verify\_cmd, [272](#)
  - ehsm\_soc\_image\_verify\_st, [276](#)
  - soc\_image\_upgrade\_input, [374](#)
  - soc\_image\_verify\_input, [382](#)
- header\_size
  - ehsm\_secure\_boot\_st, [229](#)
  - ehsm\_soc\_image\_verify\_cmd, [272](#)
  - ehsm\_soc\_image\_verify\_st, [276](#)
  - soc\_image\_upgrade\_info, [369](#)
  - soc\_image\_upgrade\_input, [374](#)
  - soc\_image\_verify\_info, [379](#)
  - soc\_image\_verify\_input, [382](#)
- HeaderAddr
  - HSM\_BootCfgType, [288](#)
  - HSM\_ImageVerifyType\_, [304](#)
  - HSM\_SecureUpgradeType\_, [335](#)
- HeaderSize
  - HSM\_BootCfgType, [288](#)
  - HSM\_ImageVerifyType\_, [305](#)
  - HSM\_SecureUpgradeType\_, [335](#)
- hid
  - ehsm\_cmd\_hdr\_sm9\_st, [83](#)
  - ehsm\_gen\_sm9\_key\_param, [146](#)
  - ehsm\_sm9\_wrap\_key\_cmd, [268](#)
  - ehsm\_sm9\_wrap\_key\_param, [270](#)
  - sm9cipher\_testvec, [365](#)
- high\_word
  - counter\_value\_64\_t, [22](#)
- host\_dst\_addr
  - ehsm\_otp\_read\_cmd, [212](#)
- host\_src\_addr
  - ehsm\_otp\_write\_cmd, [214](#)
- Hsm\_DhPubKeyType\_, [297](#)
  - DhParam, [298](#)
  - pub, [298](#)
- Hsm\_Hal.c, [923](#)
  - HSM\_Hal\_AesCipher, [926](#)
  - HSM\_Hal\_CipherMac, [927](#)
  - HSM\_Hal\_DebugAuth, [927](#)
  - HSM\_Hal\_Deinit, [928](#)
  - HSM\_Hal\_DeriveKey, [928](#)
  - HSM\_Hal\_DisableSecureBoot, [929](#)
  - HSM\_Hal\_EccSign, [929](#)
  - HSM\_Hal\_EnableSecureBoot, [930](#)
  - HSM\_Hal\_GenerateDHKey, [930](#)
  - HSM\_Hal\_GenerateKey, [930](#)
  - HSM\_Hal\_GetChallenge, [931](#)
  - HSM\_Hal\_GetHsmFwVersion, [931](#)
  - HSM\_Hal\_GetKeyStatus, [932](#)
  - HSM\_Hal\_GetLifeCycle, [932](#)
  - HSM\_Hal\_GetLockState, [932](#)
  - HSM\_Hal\_GetPubKeyFromPrvKey, [933](#)
  - HSM\_Hal\_GetRnd, [933](#)
  - HSM\_Hal\_GetRndKey, [934](#)
  - HSM\_Hal\_GetSecretKey, [934](#)
  - HSM\_Hal\_Hash, [935](#)
  - HSM\_Hal\_HashMac, [936](#)
  - HSM\_Hal\_HostImageSecureUpgrade, [936](#)
  - HSM\_Hal\_HostImageSecureVerify, [936](#)
  - HSM\_Hal\_Init, [937](#)
  - HSM\_Hal\_InstallCallback, [937](#)
  - HSM\_Hal\_Lock, [938](#)
  - HSM\_Hal\_OtpRead, [938](#)
  - HSM\_Hal\_OtpWrite, [939](#)
  - HSM\_Hal\_RemoveKey, [939](#)
  - HSM\_Hal\_RsaCipher, [940](#)
  - HSM\_Hal\_RsaSign, [940](#)
  - HSM\_Hal\_SetImageSecureUpgradeAlgo, [941](#)
  - HSM\_Hal\_SetImageSecureVerifyAlgo, [941](#)
  - HSM\_Hal\_SetLifeCycle, [941](#)
  - HSM\_Hal\_SetOtpExternalKey, [942](#)
  - HSM\_Hal\_SetOtpKeyCipherAlgo, [942](#)
  - HSM\_Hal\_SetOtpRndKey, [943](#)
  - HSM\_Hal\_SetPlainKey, [943](#)
  - HSM\_Hal\_SetSecretKey, [944](#)
  - HSM\_Hal\_Sm2Cipher, [944](#)
  - HSM\_Hal\_Sm2Sign, [945](#)
  - HSM\_Hal\_Sm4Cipher, [945](#)
  - HSM\_Hal\_Unlock, [946](#)
  - ISR, [946](#)
- Hsm\_Hal.h, [947](#)
  - AES128\_CMAC\_WITH\_AES128\_CBC\_UPGRADE, [954](#)
  - AES128\_CMAC\_WITH\_AES128\_CBC\_VERIFY, [954](#)
  - HSM\_AsymAlgoType, [961](#)
  - HSM\_ChallengeType, [961](#)
  - HSM\_CipherDirection, [962](#)
  - HSM\_CipherMode, [962](#)
  - HSM\_DEBUG\_MODE, [954](#)
  - HSM\_DESTROY\_MODE, [954](#)
  - HSM\_DEVELOP\_MODE, [954](#)
  - HSM\_DISABLE, [954](#)
  - HSM\_DebugAuthAlgoType, [962](#)
  - HSM\_DhPubKeyType, [960](#)
  - HSM\_FLASH\_KEY\_BASE, [955](#)
  - HSM\_FLASH\_KEY\_MAX\_SLOT\_NUM, [955](#)
  - HSM\_FLASH\_KEY\_OFFSET, [955](#)
  - HSM\_FLASH\_KEY\_PAGE\_NUM, [955](#)
  - HSM\_FLASH\_KEY\_SLOT\_LEN, [955](#)
  - HSM\_FLASH\_PAGE\_NUM\_OFFSET, [956](#)



HSM\_FLASH\_PAGE\_NUM, 955  
 HSM\_FLASH\_PAGE\_REVERSE\_LEN, 956  
 HSM\_FLASH\_PAGE\_VALID\_LEN, 956  
 HSM\_GenKeyAlgo, 963  
 HSM\_Hal\_AesCipher, 972  
 HSM\_Hal\_CipherMac, 972  
 HSM\_Hal\_DebugAuth, 973  
 HSM\_Hal\_Deinit, 973  
 HSM\_Hal\_DeriveKey, 974  
 HSM\_Hal\_DisableSecureBoot, 974  
 HSM\_Hal\_EccSign, 974  
 HSM\_Hal\_EnableSecureBoot, 975  
 HSM\_Hal\_GenerateDHKey, 975  
 HSM\_Hal\_GenerateKey, 976  
 HSM\_Hal\_GetChallenge, 976  
 HSM\_Hal\_GetHsmFwVersion, 977  
 HSM\_Hal\_GetKeyStatus, 977  
 HSM\_Hal\_GetLifeCycle, 978  
 HSM\_Hal\_GetLockState, 978  
 HSM\_Hal\_GetPubKeyFromPrvKey, 978  
 HSM\_Hal\_GetRnd, 979  
 HSM\_Hal\_GetRndKey, 979  
 HSM\_Hal\_GetSecretkey, 980  
 HSM\_Hal\_Hash, 980  
 HSM\_Hal\_HashMac, 981  
 HSM\_Hal\_HostImageSecureUpgrade, 981  
 HSM\_Hal\_HostImageSecureVerify, 982  
 HSM\_Hal\_Init, 982  
 HSM\_Hal\_InstallCallback, 983  
 HSM\_Hal\_Lock, 983  
 HSM\_Hal\_OtpRead, 984  
 HSM\_Hal\_OtpWrite, 984  
 HSM\_Hal\_RemoveKey, 985  
 HSM\_Hal\_RsaCipher, 985  
 HSM\_Hal\_RsaSign, 986  
 HSM\_Hal\_SetImageSecureUpgradeAlgo, 986  
 HSM\_Hal\_SetImageSecureVerifyAlgo, 987  
 HSM\_Hal\_SetLifeCycle, 988  
 HSM\_Hal\_SetOtpExternalKey, 988  
 HSM\_Hal\_SetOtpKeyCipherAlgo, 989  
 HSM\_Hal\_SetOtpRndKey, 989  
 HSM\_Hal\_SetPlainKey, 990  
 HSM\_Hal\_SetSecretKey, 990  
 HSM\_Hal\_SetSecureBootCfg, 991  
 HSM\_Hal\_Sm2Cipher, 991  
 HSM\_Hal\_Sm2Sign, 992  
 HSM\_Hal\_Sm4Cipher, 992  
 HSM\_Hal\_Unlock, 993  
 HSM\_HashAlgoType, 963  
 HSM\_ImageType, 964  
 HSM\_ImageVerifyType, 960  
 HSM\_IrqNum, 964  
 HSM\_KEY\_DATA\_VALID\_TAG, 956  
 HSM\_KdfType, 965  
 HSM\_KeyAlgoType, 965  
 HSM\_KeyId, 965  
 HSM\_KeyStorageType, 967  
 HSM\_LifeCycleType, 968  
 HSM\_MANU\_MODE, 956  
 HSM\_MacDirection, 968  
 HSM\_OtpCtrlAlgoType, 968  
 HSM\_OtpKeyCipherAlgo, 969  
 HSM\_OtpKeyId, 969  
 HSM\_OtpKeyLevel, 969  
 HSM\_PAGE\_VALID\_TAG, 956  
 HSM\_PaddingType, 970  
 HSM\_PriKeyDataType, 960  
 HSM\_ProcessMode, 970  
 HSM\_PubKeyDataType, 960  
 HSM\_RAM\_KEY\_ECC\_MAX\_NUM, 957  
 HSM\_RAM\_KEY\_MAX\_NUM, 957  
 HSM\_RAM\_KEY\_MEM\_SIZE, 957  
 HSM\_RAM\_KEY\_RSA\_MAX\_NUM, 957  
 HSM\_RAM\_KEY\_SYM\_MAX\_NUM, 957  
 HSM\_RndAlgo, 970  
 HSM\_RndOtpKeyType, 971  
 HSM\_RsaPrvCrtType, 961  
 HSM\_SECURE\_BOOT\_DISABLE, 957  
 HSM\_SECURE\_BOOT\_ENABLE, 958  
 HSM\_SLOT\_INVALID\_TAG, 958  
 HSM\_SUCCESS, 958  
 HSM\_SecureUpgradeType, 961  
 HSM\_SignDirection, 971  
 HSM\_SymAlgoType, 971  
 HSM\_TEST\_MODE, 958  
 HSM\_UNNORMAL\_MODE, 958  
 HSM\_USER\_MODE, 958  
 OTP\_KEY\_CIPHER\_AES\_CIPHER, 959  
 OTP\_KEY\_CIPHER\_SM4\_CIPHER, 959  
 RSA\_2048\_WITH\_AES128\_CBC\_UPGRADE, 959  
 RSA\_2048\_WITH\_AES128\_CBC\_VERIFY, 959  
 SM2\_WITH\_SM4\_CBC\_UPGRADE, 959  
 SM2\_WITH\_SM4\_CBC\_VERIFY, 959  
 SM4\_CMAC\_WITH\_SM4\_CBC\_UPGRADE, 960  
 SM4\_CMAC\_WITH\_SM4\_CBC\_VERIFY, 960  
 Hsm\_PriKeyDataType\_, 325  
   Dh, 325  
   DhKey, 326  
   EccKey, 326  
   KdfKey, 326  
   RsaCtr, 326  
   RsaD, 326  
   SymKey, 326  
 Hsm\_PubKeyDataType\_, 327  
   Dh, 327  
   Ecc, 327  
   Rsa, 327  
 Hsm\_RsaCrtType\_, 330  
   dp, 330  
   dq, 331  
   p, 331  
   q, 331  
   u, 331  
 hw\_interrupt\_disable  
   eHSM\_Dspt\_CryObj\_Ip.c, 498  
   eHSM\_Dspt\_Ip.c, 504  
   eHSM\_Srv\_Mgr\_Ip.c, 907  
 hw\_interrupt\_enable  
   eHSM\_Dspt\_CryObj\_Ip.c, 498  
   eHSM\_Dspt\_Ip.c, 504  
   eHSM\_Srv\_Mgr\_Ip.c, 907

- IMAGE\_ANALYSIS\_CODE
  - eHSM\_Com\_Struct\_Ip.h, [419](#)
- IMAGE\_DECRYPT\_CODE
  - eHSM\_Com\_Struct\_Ip.h, [419](#)
- IMAGE\_ENCRYPT\_CODE
  - eHSM\_Com\_Struct\_Ip.h, [419](#)
- IMAGE\_PUBLIC\_KEY\_MAX\_LENGTH
  - eHSM\_Com\_Struct\_Ip.h, [419](#)
- IMAGE\_SIGNATURE\_MAX\_LENGTH
  - eHSM\_Com\_Struct\_Ip.h, [420](#)
- ISR
  - Hsm\_Hal.c, [946](#)
- IV\_BUFF\_SIZE
  - eHSM\_If\_Asr\_Cipher\_Ip.c, [545](#)
- id
  - sm9cipher\_testvec, [365](#)
- id\_addr
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, [149](#)
  - ehsm\_sm9\_unwrap\_key\_cmd, [264](#)
  - ehsm\_sm9\_unwrap\_key\_param, [266](#)
  - ehsm\_sm9\_wrap\_key\_cmd, [268](#)
- id\_size
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, [149](#)
  - ehsm\_sm9\_gen\_tmp\_pubkey\_param, [255](#)
  - ehsm\_sm9\_unwrap\_key\_cmd, [264](#)
  - ehsm\_sm9\_unwrap\_key\_param, [266](#)
  - ehsm\_sm9\_wrap\_key\_cmd, [268](#)
  - ehsm\_sm9\_wrap\_key\_param, [270](#)
- id\_sz
  - sm9cipher\_testvec, [365](#)
- image
  - ehsm\_image, [162](#)
  - ehsm\_image\_verify\_st, [169](#)
- image\_addr
  - ehsm\_image\_upgrade\_cmd, [164](#)
  - ehsm\_image\_verifiy\_cmd, [167](#)
  - ehsm\_secure\_boot\_st, [229](#)
  - ehsm\_soc\_image\_verify\_st, [276](#)
- image\_size
  - ehsm\_image, [162](#)
  - ehsm\_image\_upgrade\_cmd, [164](#)
  - ehsm\_image\_verifiy\_cmd, [167](#)
  - ehsm\_image\_verify\_st, [170](#)
  - ehsm\_secure\_boot\_st, [229](#)
  - ehsm\_soc\_image\_verify\_st, [276](#)
- image\_upgrade
  - ehsm\_mailbox\_req, [200](#)
- image\_verify
  - ehsm\_mailbox\_req, [200](#)
- import\_key
  - ehsm\_mailbox\_req, [200](#)
- import\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, [575](#)
- in
  - ehsm\_fast\_cmac\_st, [134](#)
- InBuf
  - HSM\_InOutType, [310](#)
- InBufLen
  - HSM\_InOutType, [310](#)
- Increment\_Counter
  - eHSM\_If\_Evita\_Ip.h, [651](#)
- IndexInfo
  - HSM\_FlashKeyType, [300](#)
- info\_stuct
  - ehsm\_sm9\_exchg\_key\_cmd, [247](#)
- InitLSram
  - HSM\_BootCfgType, [288](#)
- InitUSram
  - HSM\_BootCfgType, [288](#)
- input64
  - Crypto\_JobPrimitiveInputOutputType, [35](#)
- input\_addr
  - ehsm\_cmd\_cipher\_st, [71](#)
  - ehsm\_fw\_encrypt\_key\_cmd, [137](#)
- input\_data
  - ehsm\_aead\_data\_ptr, [64](#)
- input\_size
  - ehsm\_cmd\_cipher\_st, [71](#)
  - ehsm\_fw\_encrypt\_key\_cmd, [137](#)
- inputKeyElementId
  - Crypto\_JobRedirectionInfoType, [38](#)
- inputKeyId
  - Crypto\_JobRedirectionInfoType, [38](#)
- inputLength
  - Crypto\_JobPrimitiveInputOutputType, [35](#)
- inputPtr
  - Crypto\_JobPrimitiveInputOutputType, [35](#)
- is\_crt
  - ehsm\_cmd\_hdr\_rsa\_keygen\_st, [79](#)
- is\_crt\_mode
  - rsacipher\_testvec, [358](#)
- is\_hmac
  - cipher\_session\_st, [16](#)
- itera\_times
  - crypto\_key\_derive\_info, [42](#)
  - ehsm\_derive\_key\_cmd, [111](#)
  - ehsm\_key\_derived\_param, [178](#)
- lv
  - HSM\_SymCfgType, [340](#)
- iv
  - aead\_testvec, [9](#)
  - cipher\_testvec, [20](#)
  - hash\_testvec, [283](#)
- iv\_len
  - hash\_testvec, [283](#)
- iv\_out
  - cipher\_testvec, [20](#)
- lvLen
  - HSM\_SymCfgType, [340](#)
- ivlen
  - aead\_testvec, [9](#)
- job\_id
  - cipher\_session\_st, [16](#)
- jobId
  - Crypto\_JobInfoType, [33](#)
  - Crypto\_JobType, [41](#)
- jobInfo
  - Crypto\_JobType, [41](#)
- jobPrimitiveInfo
  - Crypto\_JobType, [41](#)
- jobPrimitiveInputOutput

- Crypto\_JobType, [41](#)
- jobPriority
  - Crypto\_JobInfoType, [33](#)
- jobRedirectionInfoRef
  - Crypto\_JobType, [41](#)
- jobState
  - Crypto\_JobType, [42](#)
- K2len
  - sm9cipher\_testvec, [365](#)
- KBUF\_BASE
  - eHSM\_Mailbox\_Reg\_lp.h, [867](#)
- KEY\_ELEMENT\_NUM
  - eHSM\_If\_Asr\_KeyCfg\_lp.h, [600](#)
- KEY\_ELEMENT\_RESERVER
  - eHSM\_If\_Asr\_KeyCfg\_lp.h, [600](#)
- KEY\_ELEMENT\_VALUE\_BUFFER\_RESERVER
  - eHSM\_If\_Asr\_KeyCfg\_lp.h, [601](#)
- KEY\_ELEMENT\_VALUE\_BUFFER\_SIZE
  - eHSM\_If\_Asr\_KeyCfg\_lp.h, [601](#)
- KMU\_BASE
  - eHSM\_Mailbox\_Reg\_lp.h, [868](#)
- Kdf
  - HSM\_DeriveKeyCfgType, [294](#)
- kdf\_alg
  - ehsm\_cmd\_hdr\_ecise\_st, [75](#)
- kdf\_hash\_alg
  - ecies\_testvec, [61](#)
- kdf\_k
  - ehsm\_prikey\_data\_, [217](#)
- kdf\_key\_size
  - key\_info\_st, [346](#)
- kdf\_testvec, [341](#)
  - key, [342](#)
  - ksize, [342](#)
  - plaintext, [342](#)
  - psize, [342](#)
  - salt, [342](#)
  - salt\_size, [343](#)
  - shared\_info, [343](#)
  - shared\_info\_size, [343](#)
- KdfKey
  - Hsm\_PriKeyDataType\_, [326](#)
- key
  - aead\_testvec, [9](#)
  - akcipher\_testvec, [12](#)
  - cipher\_testvec, [20](#)
  - eHSM\_If\_Evita\_Types\_lp.h, [710](#)
  - ehsm\_export\_pub\_key\_, [132](#)
  - hash\_testvec, [284](#)
  - kdf\_testvec, [342](#)
- key2\_size
  - ehsm\_cmd\_hdr\_sm9\_st, [83](#)
- Key\_CryptoPrimitives
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, [575](#)
- Key\_Export
  - eHSM\_If\_Evita\_lp.h, [651](#)
  - eHSM\_If\_Evita\_Key\_lp.c, [670](#)
- Key\_Import
  - eHSM\_If\_Evita\_lp.h, [652](#)
  - eHSM\_If\_Evita\_Key\_lp.c, [671](#)
- Key\_Remove
  - eHSM\_If\_Evita\_lp.h, [653](#)
  - eHSM\_If\_Evita\_Key\_lp.c, [672](#)
- Key\_Status
  - eHSM\_If\_Evita\_lp.h, [654](#)
  - eHSM\_If\_Evita\_Key\_lp.c, [673](#)
- key\_act\_use\_flags\_t, [343](#)
  - createkey, [344](#)
  - decrypt, [344](#)
  - encrypt, [344](#)
  - remove, [344](#)
  - secureboot, [344](#)
  - securestorage, [344](#)
  - sign, [345](#)
  - timestamp, [345](#)
  - transport, [345](#)
  - utcsync, [345](#)
  - verify, [345](#)
- key\_addr
  - ehsm\_cmd\_cipher\_st, [71](#)
  - ehsm\_sm9\_unwrap\_key\_cmd, [264](#)
  - ehsm\_sm9\_unwrap\_key\_param, [266](#)
  - ehsm\_sm9\_wrap\_key\_cmd, [268](#)
  - ehsm\_sm9\_wrap\_key\_param, [271](#)
- key\_alg\_id
  - ehsm\_get\_pub\_from\_priv\_cmd, [156](#)
  - ehsm\_get\_pub\_from\_priv\_param, [158](#)
- key\_auth\_addr
  - cipher\_session\_st, [17](#)
  - ehsm\_module\_status\_cmd, [208](#)
- key\_auth\_code
  - crypto\_exported\_key, [29](#)
  - ehsm\_evita\_key\_export, [120](#)
  - ehsm\_evita\_key\_import\_st, [123](#)
- key\_auth\_code\_buffer\_size
  - crypto\_exported\_key, [29](#)
- key\_auth\_code\_size
  - crypto\_exported\_key, [30](#)
  - ehsm\_evita\_key\_export, [121](#)
  - ehsm\_evita\_key\_import\_st, [123](#)
- key\_auth\_size
  - cipher\_session\_st, [17](#)
  - ehsm\_copy\_key\_cmd, [89](#)
  - ehsm\_export\_key\_cmd, [130](#)
  - ehsm\_fast\_cmac\_st, [134](#)
  - ehsm\_get\_pub\_from\_priv\_cmd, [157](#)
  - ehsm\_get\_pub\_from\_priv\_param, [158](#)
  - ehsm\_import\_key\_cmd, [172](#)
  - ehsm\_key\_copy\_param, [176](#)
  - ehsm\_key\_remove\_cmd, [183](#)
  - ehsm\_key\_remove\_param, [184](#)
  - ehsm\_keyexchange\_key\_info, [194](#)
  - ehsm\_module\_status\_cmd, [208](#)
  - ehsm\_module\_status\_st, [210](#)
  - ehsm\_sm9\_inexport\_key\_param, [261](#)
- key\_auth\_value
  - ehsm\_copy\_key\_cmd, [89](#)
  - ehsm\_export\_key\_cmd, [130](#)
  - ehsm\_fast\_cmac\_st, [135](#)
  - ehsm\_get\_pub\_from\_priv\_cmd, [157](#)
  - ehsm\_get\_pub\_from\_priv\_param, [159](#)

- ehsm\_import\_key\_cmd, 172
- ehsm\_key\_copy\_param, 177
- ehsm\_key\_remove\_cmd, 183
- ehsm\_key\_remove\_param, 184
- ehsm\_keyexchange\_key\_info, 194
- ehsm\_module\_status\_st, 210
- ehsm\_sm9\_inexport\_key\_param, 261
- key\_authenticity\_code
  - crypto\_import\_evita\_key\_info, 31
- key\_authenticity\_code\_size
  - crypto\_import\_evita\_key\_info, 31
- key\_blob
  - ehsm\_sm9\_inexport\_key\_param, 261
- key\_blob\_size
  - ehsm\_sm9\_inexport\_key\_param, 262
- key\_data
  - eHSM\_If\_Evita\_Types\_Ip.h, 710
  - ehsm\_fw\_encrypt\_key, 136
  - ehsm\_se\_key\_, 227
  - ehsm\_she\_load\_plain\_key\_cmd, 243
  - ehsm\_she\_plain\_key\_host\_param, 244
  - ehsm\_she\_plain\_key\_param, 245
- key\_deriv\_func
  - ehsm\_derive\_key\_cmd, 111
  - ehsm\_key\_derived\_param, 178
- key\_element\_data
  - ehsm\_create\_dh\_key\_param, 94
  - ehsm\_create\_evita\_key\_param, 99
  - ehsm\_create\_random\_key\_param, 101
  - ehsm\_key\_copy\_param, 177
  - ehsm\_key\_derived\_param, 179
- key\_element\_size
  - ehsm\_create\_dh\_key\_param, 94
  - ehsm\_create\_evita\_key\_param, 99
  - ehsm\_create\_random\_key\_param, 101
  - ehsm\_key\_copy\_param, 177
  - ehsm\_key\_derived\_param, 179
- key\_handle
  - cipher\_session\_st, 17
  - crypto\_copy\_key\_info, 26
  - crypto\_evita\_key\_info, 28
  - crypto\_key\_export\_info, 46
  - crypto\_key\_status\_info, 48
  - eHSM\_If\_Evita\_Types\_Ip.h, 710
  - ehsm\_cmd\_cipher\_st, 72
  - ehsm\_copy\_key\_cmd, 89
  - ehsm\_create\_dh\_key\_param, 95
  - ehsm\_create\_evita\_key\_param, 100
  - ehsm\_create\_random\_key\_param, 101
  - ehsm\_evita\_key\_export, 121
  - ehsm\_evita\_key\_import\_st, 123
  - ehsm\_exchange\_sm9\_key\_param, 126
  - ehsm\_export\_key\_cmd, 130
  - ehsm\_fast\_cmac\_st, 135
  - ehsm\_gen\_sm9\_key\_param, 147
  - ehsm\_get\_pub\_from\_priv\_cmd, 157
  - ehsm\_get\_pub\_from\_priv\_param, 159
  - ehsm\_key\_derived\_param, 179
  - ehsm\_key\_remove\_cmd, 183
  - ehsm\_key\_remove\_param, 184
  - ehsm\_key\_status\_cmd, 189
  - ehsm\_key\_status\_param, 190
  - ehsm\_keyexchange\_key\_info, 194
  - ehsm\_module\_status\_cmd, 208
  - ehsm\_module\_status\_st, 211
  - ehsm\_se\_key\_, 227
  - ehsm\_sm9\_exckey\_gen\_tmpkey\_param, 251
  - ehsm\_sm9\_export\_key\_cmd, 253
  - ehsm\_sm9\_gen\_tmp\_pubkey\_param, 255
  - ehsm\_sm9\_get\_tmp\_pubkey\_cmd, 257
  - ehsm\_sm9\_inexport\_key\_param, 262
  - ehsm\_sm9\_remove\_key\_cmd, 263
  - key\_info\_st, 346
- key\_identifier
  - ehsm\_key\_attr\_data\_, 175
- key\_info
  - crypto\_key\_derive\_info, 43
  - ehsm\_key\_attr\_data\_, 175
- key\_info\_st, 346
  - dh\_key\_size, 346
  - dh\_param, 346
  - kdf\_key\_size, 346
  - key\_handle, 346
  - random\_key\_size, 347
  - rsa\_e\_bytes\_size, 347
  - storage\_info, 347
  - storage\_key\_type, 347
- key\_is\_plain
  - ehsm\_sm9\_import\_key\_cmd, 260
  - ehsm\_sm9\_inexport\_key\_param, 262
- key\_len
  - akcipher\_testvec, 12
- key\_param
  - ehsm\_gen\_sm9\_key\_param, 147
- key\_pub\_data
  - eHSM\_If\_Evita\_Types\_Ip.h, 710
  - ehsm\_pub\_key\_, 218
- key\_remove
  - ehsm\_mailbox\_req, 200
- key\_remove\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, 575
- key\_sign\_data
  - eHSM\_If\_Evita\_Types\_Ip.h, 710
  - ehsm\_key\_status\_, 187
- key\_signatrue
  - eHSM\_If\_Evita\_Types\_Ip.h, 710
  - ehsm\_internal\_key\_, 174
- key\_signatrue\_off
  - ehsm\_key\_attr\_data\_, 175
- key\_size
  - crypto\_create\_evita\_key\_info, 27
  - ehsm\_cmd\_cipher\_st, 72
  - ehsm\_create\_dh\_key\_cmd, 91
  - ehsm\_create\_dh\_key\_param, 95
  - ehsm\_create\_evita\_key\_param, 100
  - ehsm\_create\_random\_key\_param, 101
  - ehsm\_derive\_key\_cmd, 111
  - ehsm\_exchange\_sm9\_key\_param, 126
  - ehsm\_fw\_encrypt\_key, 136
  - ehsm\_key\_derived\_param, 179
  - ehsm\_sm9\_exchg\_key\_cmd, 247
  - ehsm\_sm9\_unwrap\_key\_cmd, 264

- ehsm\_sm9\_unwrap\_key\_param, 267
- ehsm\_sm9\_wrap\_key\_cmd, 268
- ehsm\_sm9\_wrap\_key\_param, 271
- ehsm\_sym\_key\_size\_, 279
- key\_size\_info
  - eHSM\_If\_Evita\_Types\_lp.h, 711
  - ehsm\_pub\_key\_, 218
  - ehsm\_se\_key\_, 227
- key\_slot
  - ehsm\_fw\_encrypt\_key, 136
  - ehsm\_fw\_encrypt\_key\_cmd, 137
  - ehsm\_fw\_get\_random\_key\_cmd, 139
  - ehsm\_fw\_random\_key, 140
- key\_status
  - ehsm\_key\_status\_cmd, 189
  - ehsm\_key\_status\_param, 191
  - ehsm\_mailbox\_req, 200
- key\_status\_buffer\_size
  - ehsm\_key\_status\_param, 191
- key\_status\_elements
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, 576
- key\_status\_size
  - ehsm\_key\_status\_cmd, 189
  - ehsm\_key\_status\_param, 191
- key\_storage\_type\_e
  - eHSM\_If\_Asr\_Types\_lp.h, 618
- key\_type
  - cipher\_session\_st, 17
  - ehsm\_cmd\_hdr\_ske\_st, 81
  - ehsm\_fast\_cmac\_st, 135
  - ehsm\_fw\_encrypt\_key, 136
  - ehsm\_fw\_encrypt\_key\_cmd, 138
  - ehsm\_fw\_get\_random\_key\_cmd, 139
  - ehsm\_fw\_random\_key, 140
  - ehsm\_import\_key\_cmd, 172
  - ehsm\_sm9\_gen\_mast\_pubkey, 254
- key\_usage
  - eHSM\_If\_Evita\_Types\_lp.h, 711
  - ehsm\_copy\_key\_cmd, 89
  - ehsm\_create\_dh\_key\_cmd, 91
  - ehsm\_derive\_key\_cmd, 111
  - ehsm\_gen\_key\_cmd, 143
  - ehsm\_internal\_key\_, 174
- key\_usage\_size
  - ehsm\_copy\_key\_cmd, 89
  - ehsm\_create\_dh\_key\_cmd, 91
  - ehsm\_derive\_key\_cmd, 111
  - ehsm\_gen\_key\_cmd, 143
  - ehsm\_key\_attr\_data\_, 176
- KeyAlgo
  - HSM\_GenKeyCfgType, 301
  - HSM\_PlainKeyCfgType, 323
  - HSM\_SecretKeyCfgType, 333
- KeyHandle
  - HSM\_KeyHandleInfoType, 316
- KeyHandleAddr
  - HSM\_RamKeyInfoType, 329
- KeyId
  - HSM\_AsymCfgType, 286
  - HSM\_CMacCfgType, 291
  - HSM\_DeriveKeyCfgType, 294
  - HSM\_GenKeyCfgType, 301
  - HSM\_HMacCfgType, 303
  - HSM\_PlainKeyCfgType, 323
  - HSM\_SymCfgType, 340
- keyId
  - eHSM\_If\_Evita\_Types\_lp.h, 711
  - ehsm\_key\_status\_, 187
- keyIdSize
  - eHSM\_If\_Evita\_Types\_lp.h, 711
  - ehsm\_key\_status\_, 187
- KeyIndex
  - HSM\_KeyIndexInfoType, 316
  - HSM\_RamKeyInfoType, 329
- KeyInfo
  - HSM\_FlashKeyPageType, 299
  - HSM\_RamKeyHeadType, 328
- keyLength
  - Crypto\_AlgorithmInfoType, 23
- KeyMemStart
  - HSM\_RamKeyHeadType, 328
- KeyMemUsedSize
  - HSM\_RamKeyHeadType, 328
- KeyNum
  - HSM\_RamKeyHeadType, 329
- KeySize
  - HSM\_DeriveKeyCfgType, 294
  - HSM\_GenKeyCfgType, 302
- KeyStatus
  - HSM\_KeyStatusType, 318
- KeyStatusSize
  - HSM\_KeyStatusType, 319
- KeyType
  - CryptoKey, 54
  - HSM\_DeriveKeyCfgType, 295
  - HSM\_GenKeyCfgType, 302
  - HSM\_PlainKeyCfgType, 323
  - HSM\_SecretKeyCfgType, 333
- KeyUsage
  - HSM\_DeriveKeyCfgType, 295
  - HSM\_GenKeyCfgType, 302
- KeyUsageSize
  - HSM\_DeriveKeyCfgType, 295
  - HSM\_GenKeyCfgType, 302
- KeyUsages
  - HSM\_PlainKeyCfgType, 323
- KeyUsagesCnt
  - HSM\_PlainKeyCfgType, 323
- KeyValid
  - HSM\_KeyIndexInfoType, 316
- keyelement\_arr
  - CryptoKeyType, 57
- keyelement\_num
  - CryptoKeyType, 57
- kgc\_pub\_key
  - ehsm\_exchange\_sm9\_key\_param, 127
  - ehsm\_sm9\_exchg\_key\_cmd\_child, 249
  - ehsm\_sm9\_exckey\_gen\_tmpkey\_param, 251
- kgc\_pubkey
  - ehsm\_gen\_sm9\_userpriv\_key\_param, 150
- kgc\_public\_key
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, 245

- klen
  - aead\_testvec, [10](#)
  - cipher\_testvec, [20](#)
- kpp\_testvec, [347](#)
  - b\_public, [348](#)
  - b\_public\_size, [348](#)
  - b\_secret, [349](#)
  - b\_secret\_size, [349](#)
  - expected\_a\_public, [349](#)
  - expected\_a\_public\_size, [349](#)
  - expected\_ss, [349](#)
  - expected\_ss\_size, [349](#)
  - g, [350](#)
  - g\_size, [350](#)
  - genkey, [350](#)
  - p, [350](#)
  - p\_size, [350](#)
  - q, [350](#)
  - q\_size, [351](#)
  - secret, [351](#)
  - secret\_size, [351](#)
- ksize
  - hash\_testvec, [284](#)
  - kdf\_testvec, [342](#)
- len
  - cipher\_testvec, [21](#)
- list
  - ehsm\_cmd\_req, [85](#)
- local\_key\_auth\_size
  - ehsm\_create\_dh\_key\_cmd, [91](#)
  - ehsm\_create\_dh\_key\_param, [95](#)
- local\_key\_auth\_value
  - ehsm\_create\_dh\_key\_cmd, [91](#)
  - ehsm\_create\_dh\_key\_param, [95](#)
- local\_key\_handle
  - ehsm\_create\_dh\_key\_cmd, [92](#)
  - ehsm\_create\_dh\_key\_param, [95](#)
- local\_tmp\_key\_auth\_size
  - ehsm\_create\_dh\_sm2\_ext\_param, [97](#)
  - sm2\_ext\_param, [362](#)
- local\_tmp\_key\_auth\_value
  - ehsm\_create\_dh\_sm2\_ext\_param, [97](#)
  - sm2\_ext\_param, [362](#)
- local\_tmp\_key\_handle
  - ehsm\_create\_dh\_sm2\_ext\_param, [98](#)
  - sm2\_ext\_param, [363](#)
- LogEnable
  - HSM\_BootCfgType, [288](#)
- low\_power\_cmd
  - ehsm\_mbox\_mgr\_channel\_req, [207](#)
- low\_word
  - counter\_value\_64\_t, [22](#)
- m
  - akcipher\_testvec, [12](#)
  - rsacipher\_testvec, [358](#)
  - sm9cipher\_testvec, [366](#)
- m1
  - crypto\_she\_key, [51](#)
  - ehsm\_she\_key\_host\_param, [238](#)
  - ehsm\_she\_key\_param, [239](#)
  - ehsm\_she\_load\_export\_key\_cmd, [242](#)
- m2
  - crypto\_she\_key, [51](#)
  - ehsm\_she\_key\_host\_param, [238](#)
  - ehsm\_she\_key\_param, [239](#)
  - ehsm\_she\_load\_export\_key\_cmd, [242](#)
- m3
  - crypto\_she\_key, [51](#)
  - ehsm\_she\_key\_host\_param, [238](#)
  - ehsm\_she\_key\_param, [240](#)
  - ehsm\_she\_load\_export\_key\_cmd, [242](#)
- m4
  - crypto\_she\_key, [52](#)
  - ehsm\_she\_key\_host\_param, [238](#)
  - ehsm\_she\_key\_param, [240](#)
  - ehsm\_she\_load\_export\_key\_cmd, [242](#)
- m5
  - crypto\_she\_key, [52](#)
  - ehsm\_she\_key\_host\_param, [238](#)
  - ehsm\_she\_key\_param, [240](#)
  - ehsm\_she\_load\_export\_key\_cmd, [243](#)
- m\_size
  - akcipher\_testvec, [12](#)
  - rsacipher\_testvec, [359](#)
- m\_sz
  - sm9cipher\_testvec, [366](#)
- MAC\_Finish
  - eHSM\_If\_Evita\_Ip.h, [655](#)
  - eHSM\_If\_Evita\_SymCper\_Ip.c, [681](#)
- MAC\_Init
  - eHSM\_If\_Evita\_Ip.h, [655](#)
  - eHSM\_If\_Evita\_SymCper\_Ip.c, [683](#)
- MAC\_Update
  - eHSM\_If\_Evita\_Ip.h, [656](#)
  - eHSM\_If\_Evita\_SymCper\_Ip.c, [684](#)
- MAILBOX\_CMD\_MAX\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [856](#)
- MAILBOX\_Handler
  - eHSM\_Mailbox\_Ip.c, [846](#)
- MAX\_IMPORT\_KEY\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [856](#)
- MAX\_KEY\_AUTH\_VALUE\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [856](#)
- MAX\_KEY\_DERIVED\_PWD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [856](#)
- MAX\_RESPONSE\_DATA\_SIZE
  - eHSM\_Srv\_CmdReq\_Ip.h, [886](#)
- MAX\_SM9\_CIPHER\_KEY\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [856](#)
- MAX\_SM9\_USER\_ID\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [857](#)
- MAX\_SM9\_WARP\_KEY\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [857](#)
- MAX\_TAP
  - utest\_vecs\_st.h, [994](#)
- MB\_H2S\_NOTE
  - eHSM\_Mailbox\_Reg\_Ip.h, [868](#)
- MB\_H2S\_SOC\_INT\_EN
  - eHSM\_Mailbox\_Reg\_Ip.h, [868](#)
- MB\_H2S\_SOC\_INT



- eHSM\_Mailbox\_Reg\_lp.h, [868](#)
- MB\_HSM\_STATUS0
  - eHSM\_Mailbox\_Reg\_lp.h, [868](#)
- MB\_HSM\_STATUS1
  - eHSM\_Mailbox\_Reg\_lp.h, [868](#)
- MB\_S2H\_NOTE
  - eHSM\_Mailbox\_Reg\_lp.h, [869](#)
- MB\_S2H\_SOC\_INT\_EN
  - eHSM\_Mailbox\_Reg\_lp.h, [869](#)
- MB\_S2H\_SOC\_INT
  - eHSM\_Mailbox\_Reg\_lp.h, [869](#)
- MBOX\_HOST2HSM\_HOST\_INT\_EN
  - eHSM\_Mailbox\_Reg\_lp.h, [869](#)
- MBOX\_HOST2HSM\_HOST\_INT
  - eHSM\_Mailbox\_Reg\_lp.h, [869](#)
- MBOX\_HOST2HSM\_HSM\_INT\_EN
  - eHSM\_Mailbox\_Reg\_lp.h, [870](#)
- MBOX\_HOST2HSM\_HSM\_INT
  - eHSM\_Mailbox\_Reg\_lp.h, [869](#)
- MBOX\_HSM2HOST\_HOST\_INT\_EN
  - eHSM\_Mailbox\_Reg\_lp.h, [870](#)
- MBOX\_HSM2HOST\_HOST\_INT
  - eHSM\_Mailbox\_Reg\_lp.h, [870](#)
- MBOX\_HSMHOST\_HSM\_INT\_EN
  - eHSM\_Mailbox\_Reg\_lp.h, [870](#)
- MBOX\_HSMHOST\_HSM\_INT
  - eHSM\_Mailbox\_Reg\_lp.h, [870](#)
- MBOX\_SOCBASE
  - eHSM\_Mailbox\_Reg\_lp.h, [870](#)
- mac
  - ehsm\_fast\_cmac\_st, [135](#)
- mac\_alg
  - ehsm\_cmd\_hdr\_ecise\_st, [75](#)
- mac\_bytes
  - cipher\_session\_st, [17](#)
- mac\_hash\_alg
  - ecies\_testvec, [61](#)
- mac\_k\_byte
  - ehsm\_cmd\_hdr\_ecise\_st, [76](#)
- mac\_k\_bytes
  - ecies\_testvec, [61](#)
- mac\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, [576](#)
- mac\_mode\_e
  - eHSM\_If\_Evita\_Types\_lp.h, [706](#)
- mac\_sign
  - soc\_image\_upgrade\_info, [369](#)
- mac\_sign\_addr
  - soc\_image\_upgrade\_input, [374](#)
- mac\_sign\_size
  - soc\_image\_upgrade\_info, [369](#)
  - soc\_image\_upgrade\_input, [374](#)
- mac\_size
  - mac\_t, [352](#)
- mac\_st
  - eHSM\_If\_Evita\_Types\_lp.h, [705](#)
- mac\_t, [351](#)
  - mac\_size, [352](#)
  - mac\_value, [352](#)
  - utc\_time, [352](#)
- mac\_value
  - mac\_t, [352](#)
- MacDir
  - HSM\_CMacCfgType, [291](#)
  - HSM\_HMacCfgType, [303](#)
- MacInBuf
  - HSM\_InOutMacType, [308](#)
- MacInBufLen
  - HSM\_InOutMacType, [308](#)
- MacSignAddr
  - HSM\_SecureUpgradeType\_\_\_, [335](#)
- MacSignSize
  - HSM\_SecureUpgradeType\_\_\_, [336](#)
- mailbox\_callback
  - eHSM\_Mailbox\_lp.h, [847](#)
- mailbox\_channel, [352](#)
  - cmd\_inprogres, [353](#)
  - h2s\_note\_bit, [353](#)
  - h2s\_start\_addr, [353](#)
  - h2s\_word\_size, [353](#)
  - s2h\_note\_bit, [353](#)
  - s2h\_start\_addr, [353](#)
  - s2h\_word\_size, [353](#)
  - surpport\_asyn, [354](#)
  - type, [354](#)
- mailbox\_channel\_e
  - eHSM\_Mailbox\_Prtcl\_lp.h, [865](#)
- mailbox\_channel\_st
  - eHSM\_Mailbox\_lp.h, [847](#)
- master\_key
  - ehsm\_gen\_sm9\_key\_param, [147](#)
- master\_key\_type
  - ehsm\_gen\_sm9\_master\_key\_param, [148](#)
  - ehsm\_sm9\_get\_mast\_pubkey\_cmd, [256](#)
- max\_chunk\_size
  - ehsm\_ctx\_session\_st, [106](#)
- mbox\_channel
  - eHSM\_Mailbox\_lp.c, [847](#)
- mem\_location
  - eHSM\_If\_Evita\_Types\_lp.h, [711](#)
  - ehsm\_key\_status\_, [187](#)
- mode
  - Crypto\_AlgorithmInfoType, [23](#)
  - Crypto\_JobPrimitiveInputOutputType, [36](#)
- Module\_Status
  - eHSM\_If\_Evita\_lp.h, [657](#)
  - eHSM\_If\_Evita\_Key\_lp.c, [674](#)
- module\_status
  - ehsm\_mailbox\_req, [201](#)
- msg
  - ecies\_testvec, [61](#)
- msg\_bytes
  - ecies\_testvec, [61](#)
- n
  - ehsm\_rsa\_pubkey, [226](#)
  - HSM\_RsaPubKeyType, [332](#)
  - rsacipher\_testvec, [359](#)
- n\_bit\_size
  - ehsm\_cmd\_hdr\_rsa\_keygen\_st, [79](#)
- n\_byte\_size
  - rsacipher\_testvec, [359](#)

- n\_size
  - ehsm\_gen\_key\_cmd, [143](#)
- NULL
  - eHSM\_Types\_Ip.h, [921](#)
- name
  - crypto\_object, [49](#)
- need\_encryption
  - ehsm\_secure\_boot\_st, [229](#)
  - ehsm\_soc\_image\_verify\_st, [276](#)
- next
  - dlist\_head, [59](#)
- novrfy
  - aead\_testvec, [10](#)
- np
  - hash\_testvec, [284](#)
- nvm\_free\_size
  - ehsm\_evita\_memory\_info\_st\_, [125](#)
- nvm\_total\_size
  - ehsm\_evita\_memory\_info\_st\_, [125](#)
- o\_hsm\_err\_fw
  - ehsm\_emu\_status\_st\_, [118](#)
- o\_hsm\_err\_hw
  - ehsm\_emu\_status\_st\_, [119](#)
- o\_hsm\_err\_sensor
  - ehsm\_emu\_status\_st\_, [119](#)
- o\_hsm\_status
  - ehsm\_emu\_status\_st\_, [119](#)
- OFFSET
  - eHSM\_Compt\_List.h, [440](#)
- OTP\_BASE\_ADDR
  - AC784xx\_Hsm\_Reg.h, [386](#)
- OTP\_CONTROL\_FILED\_BYTE\_SIZE
  - eHSM\_Com\_Struct\_Ip.h, [420](#)
- OTP\_ERR\_RSP\_CTRL\_ADDR
  - AC784xx\_Hsm\_Reg.h, [386](#)
- OTP\_FW\_CTRL\_FIELD\_ADDR\_H
  - AC784xx\_Hsm\_Reg.h, [387](#)
- OTP\_FW\_CTRL\_FIELD\_ADDR\_L
  - AC784xx\_Hsm\_Reg.h, [387](#)
- OTP\_HOST\_CTRL\_FIELD\_ADDR\_H
  - AC784xx\_Hsm\_Reg.h, [387](#)
- OTP\_HOST\_CTRL\_FIELD\_ADDR\_L
  - AC784xx\_Hsm\_Reg.h, [387](#)
- OTP\_HSM\_ENABLE\_ADDR
  - AC784xx\_Hsm\_Reg.h, [387](#)
- OTP\_HSM\_VERSION\_ADDR
  - AC784xx\_Hsm\_Reg.h, [387](#)
- OTP\_HW\_CTRL\_FIELD\_ADDR
  - AC784xx\_Hsm\_Reg.h, [388](#)
- OTP\_KEY\_ADDR
  - AC784xx\_Hsm\_Reg.h, [388](#)
- OTP\_KEY\_ATTR\_BYTE\_LENGTH
  - AC784xx\_Hsm\_Reg.h, [388](#)
- OTP\_KEY\_ATTR\_ENCODE\_EACH\_LENGTH
  - AC784xx\_Hsm\_Reg.h, [388](#)
- OTP\_KEY\_ATTR\_LC
  - AC784xx\_Hsm\_Reg.h, [388](#)
- OTP\_KEY\_CIPHER\_AES\_CIPHER
  - Hsm\_Hal.h, [959](#)
- OTP\_KEY\_CIPHER\_SM4\_CIPHER
  - Hsm\_Hal.h, [959](#)
- OTP\_KEY\_CRC\_SIZE
  - AC784xx\_Hsm\_Reg.h, [389](#)
- OTP\_KEY\_CRC
  - AC784xx\_Hsm\_Reg.h, [388](#)
- OTP\_KEY\_SIZE
  - AC784xx\_Hsm\_Reg.h, [389](#)
- OTP\_LIFE\_CYCLE\_ADDR
  - AC784xx\_Hsm\_Reg.h, [389](#)
- OTP\_SECURE\_BOOT\_ADDR
  - AC784xx\_Hsm\_Reg.h, [389](#)
- OTP\_SIZE
  - AC784xx\_Hsm\_Reg.h, [389](#)
- OTP\_SOC\_VERSION\_ADDR
  - AC784xx\_Hsm\_Reg.h, [389](#)
- OTP\_UID\_ADDR
  - AC784xx\_Hsm\_Reg.h, [390](#)
- OTP\_VERSION\_ENCODE\_LENGTH
  - AC784xx\_Hsm\_Reg.h, [390](#)
- OTP\_VERSION\_LENGTH
  - AC784xx\_Hsm\_Reg.h, [390](#)
- object\_type
  - ehsm\_cmd\_req, [85](#)
- operation\_mode\_e
  - eHSM\_If\_Evita\_Types\_Ip.h, [706](#)
- otp\_data\_addr
  - ehsm\_otp\_read\_param\_st, [213](#)
  - ehsm\_otp\_write\_param\_st, [215](#)
- otp\_read
  - ehsm\_mailbox\_req, [201](#)
- otp\_write
  - ehsm\_mailbox\_req, [201](#)
- OutBuf
  - HSM\_InOutType, [310](#)
- OutBufLen
  - HSM\_InOutType, [311](#)
- output64Ptr
  - Crypto\_JobPrimitiveInputOutputType, [36](#)
- output\_addr
  - ehsm\_cmd\_cipher\_st, [72](#)
  - ehsm\_get\_challenge\_cmd, [152](#)
- output\_data
  - ehsm\_aead\_data\_ptr, [64](#)
- output\_size
  - ehsm\_cmd, [69](#)
  - ehsm\_cmd\_cipher\_st, [72](#)
- outputKeyElementId
  - Crypto\_JobRedirectionInfoType, [39](#)
- outputKeyId
  - Crypto\_JobRedirectionInfoType, [39](#)
- outputLengthPtr
  - Crypto\_JobPrimitiveInputOutputType, [36](#)
- outputPtr
  - Crypto\_JobPrimitiveInputOutputType, [36](#)
- p
  - crypto\_copy\_key\_dh\_key\_info, [24](#)
  - ehsm\_create\_random\_key\_param, [102](#)
  - ehsm\_dh\_param, [114](#)
  - ehsm\_ecc\_pubkey, [118](#)
  - ehsm\_gen\_dh\_key\_param\_st, [141](#)



- ehsm\_gen\_key\_cmd, [143](#)
- ehsm\_rsa crt\_param\_, [222](#)
- HSM\_DhParamType, [296](#)
- HSM\_EccPubKeyType, [299](#)
- Hsm\_RsaCrtType\_, [331](#)
- kpp\_testvec, [350](#)
- rsacipher\_testvec, [359](#)
- p\_byte\_size
  - rsacipher\_testvec, [359](#)
- p\_size
  - crypto\_copy\_key\_dh\_key\_info, [24](#)
  - ehsm\_create\_random\_key\_param, [102](#)
  - ehsm\_dh\_param\_size\_info, [115](#)
  - ehsm\_gen\_dh\_key\_param\_st, [141](#)
  - ehsm\_gen\_key\_cmd, [143](#)
  - kpp\_testvec, [350](#)
- PERFORMANCE\_LOG\_DEBUG
  - eHSM\_Debug\_Ip.h, [493](#)
- PERFORMANCE\_LOG\_ERROR
  - eHSM\_Debug\_Ip.h, [493](#)
- PERFORMANCE\_LOG\_INFO
  - eHSM\_Debug\_Ip.h, [493](#)
- PERFORMANCE\_LOG\_WARN
  - eHSM\_Debug\_Ip.h, [494](#)
- PERFORMANCE\_PURE\_LOG\_DEBUG
  - eHSM\_Debug\_Ip.h, [494](#)
- PURE\_LOG
  - eHSM\_Debug\_Ip.h, [494](#)
- Padding
  - HSM\_AsymCfgType, [286](#)
  - HSM\_SymCfgType, [341](#)
- padding
  - cipher\_session\_st, [17](#)
  - ehsm\_cmd\_hdr\_pke\_st, [77](#)
  - ehsm\_cmd\_hdr\_ske\_st, [81](#)
  - ehsm\_cmd\_hdr\_sm9\_st, [83](#)
  - sm9cipher\_testvec, [366](#)
- padding\_scheme\_e
  - eHSM\_If\_Evita\_Types\_Ip.h, [707](#)
- PageReverse
  - HSM\_FlashKeyPageType, [299](#)
- PageValid
  - HSM\_FlashKeyPageType, [300](#)
- param
  - ehsm\_gen\_key\_param\_st, [146](#)
- param\_len
  - akcipher\_testvec, [13](#)
- params
  - akcipher\_testvec, [13](#)
- parent\_alg
  - ehsm\_create\_dh\_key\_cmd, [92](#)
  - ehsm\_create\_dh\_key\_param, [95](#)
- parent\_key\_auth\_size
  - ehsm\_derive\_key\_cmd, [112](#)
- parent\_key\_auth\_value
  - ehsm\_derive\_key\_cmd, [112](#)
- parent\_key\_author\_size
  - ehsm\_key\_derived\_param, [179](#)
- parent\_key\_author\_value
  - ehsm\_key\_derived\_param, [179](#)
- parent\_key\_handle
  - ehsm\_derive\_key\_cmd, [112](#)
  - ehsm\_key\_copy\_param, [177](#)
  - ehsm\_key\_derived\_param, [180](#)
- passwd
  - crypto\_key\_derive\_info, [43](#)
  - ehsm\_key\_derived\_param, [180](#)
- passwd\_size
  - crypto\_key\_derive\_info, [43](#)
  - ehsm\_key\_derived\_param, [180](#)
- peer\_id
  - ehsm\_exchange\_sm9\_key\_param, [127](#)
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, [245](#)
  - ehsm\_sm9\_exchg\_key\_cmd\_child, [249](#)
  - ehsm\_sm9\_exckey\_gen\_tmpkey\_param, [251](#)
- peer\_id\_size
  - ehsm\_exchange\_sm9\_key\_param, [127](#)
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, [246](#)
  - ehsm\_sm9\_exchg\_key\_cmd\_child, [250](#)
  - ehsm\_sm9\_exckey\_gen\_tmpkey\_param, [252](#)
- peer\_pubkey
  - sm2\_ext\_param, [363](#)
- peer\_temp\_pubkey
  - ehsm\_create\_dh\_sm2\_ext\_param, [98](#)
  - sm2\_ext\_param, [363](#)
- peer\_tmp\_pub
  - ehsm\_exchange\_sm9\_key\_param, [127](#)
  - ehsm\_sm9\_exchg\_key\_cmd\_child, [250](#)
- Pke\_CryptoPrimitives
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, [577](#)
- plaintext
  - hash\_testvec, [284](#)
  - kdf\_testvec, [342](#)
- plen
  - aead\_testvec, [10](#)
- PIIFreq
  - HSM\_BootCfgType, [289](#)
- point\_form
  - ecies\_testvec, [61](#)
- power\_mode
  - ehsm\_low\_power\_cmd, [195](#)
- Ppub
  - sm9cipher\_testvec, [366](#)
- prev
  - dlist\_head, [59](#)
- prikey
  - eHSM\_If\_Evita\_Types\_Ip.h, [711](#)
  - ehsm\_internal\_key\_, [174](#)
- prikey\_enc\_size
  - eHSM\_If\_Evita\_Types\_Ip.h, [712](#)
  - ehsm\_internal\_key\_, [174](#)
- primitiveInfo
  - Crypto\_JobPrimitiveInfoType, [34](#)
- priority
  - ehsm\_cmd\_req, [86](#)
  - ehsm\_service\_info, [235](#)
- Priv
  - HSM\_DhPriKeyType, [297](#)
- priv
  - ehsm\_dh\_prikey, [116](#)
  - sm9cipher\_testvec, [366](#)
- priv\_key

- ehsm\_gen\_sm9\_key\_param, [147](#)
- priv\_key\_size
  - ehsm\_ecc\_key\_size\_, [117](#)
- priv\_key\_type
  - ehsm\_gen\_sm9\_userpriv\_key\_param, [151](#)
  - ehsm\_sm9\_exckey\_gen\_tmpkey\_param, [252](#)
- PrivKey
  - HSM\_PlainKeyCfgType, [324](#)
- PrivKeyLen
  - HSM\_PlainKeyCfgType, [324](#)
- process\_mode
  - ehsm\_cmd\_hdr\_pke\_st, [77](#)
  - ehsm\_cmd\_hdr\_ske\_st, [81](#)
  - ehsm\_image, [162](#)
  - ehsm\_image\_upgrade\_cmd, [164](#)
  - ehsm\_image\_verifiy\_cmd, [167](#)
  - ehsm\_image\_verify\_st, [170](#)
  - soc\_image\_upgrade\_info, [369](#)
  - soc\_image\_upgrade\_input, [374](#)
- ProcessMode
  - HSM\_SecureUpgradeType\_, [336](#)
- processingType
  - Crypto\_JobPrimitiveInfoType, [34](#)
- psize
  - hash\_testvec, [284](#)
  - kdf\_testvec, [342](#)
- pctest\_time\_counting\_end
  - eHSM\_Dspt\_lp.c, [504](#)
- pctest\_time\_counting\_get\_state
  - eHSM\_Dspt\_lp.c, [504](#)
- pctest
  - aead\_testvec, [10](#)
  - cipher\_testvec, [21](#)
- pub
  - ehsm\_dh\_pubkey, [116](#)
  - Hsm\_DhPubKeyType\_, [298](#)
- pub\_addr
  - ehsm\_debug\_authentication\_cmd, [109](#)
- pub\_key
  - ehsm\_sm9\_gen\_mast\_pubkey, [255](#)
  - ehsm\_sm9\_gen\_tmp\_pubkey\_param, [256](#)
  - ehsm\_sm9\_wrap\_key\_cmd, [268](#)
  - ehsm\_sm9\_wrap\_key\_param, [271](#)
- pub\_key\_size
  - ehsm\_ecc\_key\_size\_, [117](#)
- pub\_size
  - ehsm\_debug\_authentication\_cmd, [109](#)
- PubKey
  - HSM\_DebugAuthConfigType, [293](#)
  - HSM\_PlainKeyCfgType, [324](#)
- PubKeyAddr
  - HSM\_BootCfgType, [289](#)
- PubKeyLen
  - HSM\_PlainKeyCfgType, [324](#)
- PubKeySize
  - HSM\_DebugAuthConfigType, [293](#)
- pubkey
  - eHSM\_If\_Evita\_Types\_lp.h, [712](#)
  - ehsm\_internal\_key\_, [174](#)
  - ehsm\_key\_status\_, [187](#)
  - soc\_image\_verify\_info, [379](#)
- pubkey\_addr
  - ehsm\_secure\_boot\_st, [229](#)
  - ehsm\_soc\_image\_verify\_cmd, [272](#)
  - ehsm\_soc\_image\_verify\_st, [276](#)
  - soc\_image\_verify\_input, [382](#)
- pubkey\_size
  - ehsm\_secure\_boot\_st, [230](#)
  - ehsm\_soc\_image\_verify\_cmd, [272](#)
  - ehsm\_soc\_image\_verify\_st, [277](#)
  - soc\_image\_verify\_info, [379](#)
  - soc\_image\_verify\_input, [382](#)
- PubkeyAddr
  - HSM\_ImageVerifyType\_, [305](#)
- PubkeySize
  - HSM\_ImageVerifyType\_, [305](#)
- public\_key
  - ehsm\_debug\_auth\_st, [107](#)
- public\_key\_addr
  - ehsm\_get\_pub\_from\_priv\_cmd, [157](#)
  - ehsm\_get\_pub\_from\_priv\_param, [159](#)
  - ehsm\_sm9\_get\_mast\_pubkey\_cmd, [257](#)
  - ehsm\_sm9\_get\_tmp\_pubkey\_cmd, [258](#)
- public\_key\_buffer\_size
  - ehsm\_get\_pub\_from\_priv\_cmd, [157](#)
  - ehsm\_get\_pub\_from\_priv\_param, [159](#)
- public\_key\_size
  - ehsm\_debug\_auth\_st, [107](#)
  - ehsm\_get\_pub\_from\_priv\_param, [159](#)
- public\_key\_vec
  - akcipher\_testvec, [13](#)
  - rsacipher\_testvec, [359](#)
- pw\_data
  - ehsm\_derive\_key\_cmd, [112](#)
- pw\_size
  - ehsm\_derive\_key\_cmd, [112](#)
- q
  - crypto\_copy\_key\_dh\_key\_info, [24](#)
  - ehsm\_create\_random\_key\_param, [102](#)
  - ehsm\_dh\_param, [114](#)
  - ehsm\_gen\_dh\_key\_param\_st, [141](#)
  - ehsm\_gen\_key\_cmd, [144](#)
  - ehsm\_rsa crt\_param\_, [222](#)
  - HSM\_DhParamType, [296](#)
  - Hsm\_RsaCrtType\_, [331](#)
  - kpp\_testvec, [350](#)
  - rsacipher\_testvec, [360](#)
- q\_byte\_size
  - rsacipher\_testvec, [360](#)
- q\_size
  - crypto\_copy\_key\_dh\_key\_info, [24](#)
  - ehsm\_create\_random\_key\_param, [102](#)
  - ehsm\_dh\_param\_size\_info, [115](#)
  - ehsm\_gen\_dh\_key\_param\_st, [141](#)
  - ehsm\_gen\_key\_cmd, [144](#)
  - kpp\_testvec, [351](#)
- queue\_capacity
  - crypto\_object, [49](#)
- r
  - sm9cipher\_testvec, [366](#)

r\_sz  
     sm9cipher\_testvec, [367](#)  
 rATTR\_D0  
     eHSM\_Mailbox\_Reg\_lp.h, [871](#)  
 rATTR\_D1  
     eHSM\_Mailbox\_Reg\_lp.h, [871](#)  
 rATTR\_D2  
     eHSM\_Mailbox\_Reg\_lp.h, [871](#)  
 rERR\_ST  
     eHSM\_Mailbox\_Reg\_lp.h, [871](#)  
 RESPONSE\_TAG\_INDEX  
     eHSM\_Mailbox\_Prtcl\_lp.h, [857](#)  
 rKBUF  
     eHSM\_Mailbox\_Reg\_lp.h, [871](#)  
 rKMU\_CTRL  
     eHSM\_Mailbox\_Reg\_lp.h, [871](#)  
 rKMU\_INT\_EN  
     eHSM\_Mailbox\_Reg\_lp.h, [872](#)  
 rKMU\_STA  
     eHSM\_Mailbox\_Reg\_lp.h, [872](#)  
 RNG\_Get\_Random  
     eHSM\_If\_Evita\_lp.h, [658](#)  
     eHSM\_If\_Evita\_Rng\_lp.c, [675](#)  
 RNG\_REQUEST\_MAX  
     eHSM\_If\_Evita\_Types\_lp.h, [702](#)  
 RSA\_2048\_WITH\_AES128\_CBC\_UPGRADE  
     Hsm\_Hal.h, [959](#)  
 RSA\_2048\_WITH\_AES128\_CBC\_VERIFY  
     Hsm\_Hal.h, [959](#)  
 rSN\_D0  
     eHSM\_Mailbox\_Reg\_lp.h, [872](#)  
 rSN\_D1  
     eHSM\_Mailbox\_Reg\_lp.h, [872](#)  
 rSOCMBOX\_CMD\_D0  
     eHSM\_Mailbox\_Reg\_lp.h, [872](#)  
 rSOCMBOX\_CMD\_D1  
     eHSM\_Mailbox\_Reg\_lp.h, [872](#)  
 rSOCMBOX\_CMD\_D10  
     eHSM\_Mailbox\_Reg\_lp.h, [873](#)  
 rSOCMBOX\_CMD\_D11  
     eHSM\_Mailbox\_Reg\_lp.h, [873](#)  
 rSOCMBOX\_CMD\_D12  
     eHSM\_Mailbox\_Reg\_lp.h, [873](#)  
 rSOCMBOX\_CMD\_D13  
     eHSM\_Mailbox\_Reg\_lp.h, [873](#)  
 rSOCMBOX\_CMD\_D14  
     eHSM\_Mailbox\_Reg\_lp.h, [873](#)  
 rSOCMBOX\_CMD\_D15  
     eHSM\_Mailbox\_Reg\_lp.h, [873](#)  
 rSOCMBOX\_CMD\_D2  
     eHSM\_Mailbox\_Reg\_lp.h, [874](#)  
 rSOCMBOX\_CMD\_D3  
     eHSM\_Mailbox\_Reg\_lp.h, [874](#)  
 rSOCMBOX\_CMD\_D4  
     eHSM\_Mailbox\_Reg\_lp.h, [874](#)  
 rSOCMBOX\_CMD\_D5  
     eHSM\_Mailbox\_Reg\_lp.h, [874](#)  
 rSOCMBOX\_CMD\_D6  
     eHSM\_Mailbox\_Reg\_lp.h, [874](#)  
 rSOCMBOX\_CMD\_D7  
     eHSM\_Mailbox\_Reg\_lp.h, [874](#)  
 rSOCMBOX\_CMD\_D8  
     eHSM\_Mailbox\_Reg\_lp.h, [875](#)  
 rSOCMBOX\_CMD\_D9  
     eHSM\_Mailbox\_Reg\_lp.h, [875](#)  
 rSOCMBOX\_RSP\_D0  
     eHSM\_Mailbox\_Reg\_lp.h, [875](#)  
 rSOCMBOX\_RSP\_D01  
     eHSM\_Mailbox\_Reg\_lp.h, [875](#)  
 rVER\_D0  
     eHSM\_Mailbox\_Reg\_lp.h, [875](#)  
 rVER\_D1  
     eHSM\_Mailbox\_Reg\_lp.h, [875](#)  
 rVER\_D2  
     eHSM\_Mailbox\_Reg\_lp.h, [876](#)  
 rVER\_D3  
     eHSM\_Mailbox\_Reg\_lp.h, [876](#)  
 ram\_free\_size  
     ehsm\_evita\_memory\_info\_st\_, [125](#)  
 ram\_total\_size  
     ehsm\_evita\_memory\_info\_st\_, [125](#)  
 random\_data\_addr  
     ehsm\_crypto\_randomgenerate\_param, [104](#)  
     ehsm\_rng\_generate\_cmd, [220](#)  
 random\_key\_size  
     key\_info\_st, [347](#)  
 RandomKeySize  
     HSM\_PlainKeyCfgType, [324](#)  
 raw\_key  
     ehsm\_she\_key\_st, [241](#)  
 Read\_Counter  
     eHSM\_If\_Evita\_lp.h, [658](#)  
 read\_data\_size  
     ehsm\_otp\_read\_param\_st, [213](#)  
 receiver\_pri\_key  
     ecies\_testvec, [62](#)  
 receiver\_pri\_key\_sz  
     ecies\_testvec, [62](#)  
 receiver\_pub\_key  
     ecies\_testvec, [62](#)  
 receiver\_pub\_key\_sz  
     ecies\_testvec, [62](#)  
 redirectionConfig  
     Crypto\_JobRedirectionInfoType, [39](#)  
 release\_cb  
     ehsm\_cmd\_req, [86](#)  
 remain\_data\_sz  
     ehsm\_ctx\_block\_mgr, [105](#)  
 remote\_key\_auth\_or\_pub\_key\_size  
     ehsm\_create\_dh\_key\_param, [96](#)  
 remote\_key\_auth\_size  
     ehsm\_create\_dh\_key\_cmd, [92](#)  
 remote\_key\_auth\_value  
     ehsm\_create\_dh\_key\_cmd, [92](#)  
 remote\_key\_auth\_value\_or\_pub\_key  
     ehsm\_create\_dh\_key\_param, [96](#)  
 remote\_key\_handle  
     ehsm\_create\_dh\_key\_cmd, [92](#)  
     ehsm\_create\_dh\_key\_param, [96](#)  
 remove  
     ehsm\_key\_usages\_st, [192](#)  
     HSM\_KeyActUseFlagsType, [312](#)

- HSM\_KeyUsagesType, 320
- key\_act\_use\_flags\_t, 344
- req\_cb
  - ehsm\_cmd\_req, 86
- req\_cipher
  - ehsm\_cmd, 69
- req\_ctx
  - ehsm\_cmd\_req, 86
- req\_type
  - ehsm\_cmd\_req, 86
  - ehsm\_service\_info, 235
- reqhdl
  - ehsm\_service, 233
- request\_size
  - ehsm\_crypto\_randomgenerate\_param, 104
  - ehsm\_rng\_generate\_cmd, 220
- resered1
  - ehsm\_soc\_image\_verify\_cmd, 273
- resered2
  - ehsm\_soc\_image\_verify\_cmd, 273
- resered3
  - ehsm\_soc\_image\_verify\_cmd, 273
- reserved
  - eHSM\_If\_Evita\_Types\_lp.h, 712
  - ehsm\_change\_control\_field\_cmd, 65
  - ehsm\_get\_she\_id\_cmd, 160
  - ehsm\_low\_power\_cmd, 195
  - ehsm\_module\_status\_cmd, 208
  - ehsm\_rsa\_dh\_key\_size\_, 223
  - ehsm\_rsa\_key\_size\_, 224
  - ehsm\_se\_key\_, 227
  - ehsm\_she\_key\_st, 241
  - ehsm\_sym\_key\_size\_, 280
- reserved1
  - ehsm\_copy\_key\_cmd, 90
  - ehsm\_derive\_key\_cmd, 112
  - ehsm\_fw\_get\_random\_key\_cmd, 139
  - ehsm\_gen\_key\_cmd, 144
  - ehsm\_get\_challenge\_cmd, 152
  - ehsm\_get\_pub\_from\_priv\_cmd, 157
  - ehsm\_key\_remove\_cmd, 183
  - ehsm\_key\_status\_cmd, 189
  - ehsm\_sm9\_get\_tmp\_pubkey\_cmd, 258
  - ehsm\_sm9\_remove\_key\_cmd, 263
- reserved2
  - ehsm\_create\_dh\_key\_cmd, 92
  - ehsm\_create\_dh\_sm2\_ext\_param, 98
  - ehsm\_derive\_key\_cmd, 113
  - ehsm\_gen\_key\_cmd, 144
  - ehsm\_get\_pub\_from\_priv\_cmd, 158
  - ehsm\_key\_status\_cmd, 189
  - ehsm\_sm9\_get\_mast\_pubkey\_cmd, 257
  - ehsm\_sm9\_get\_tmp\_pubkey\_cmd, 258
- reserved3
  - ehsm\_create\_dh\_key\_cmd, 93
  - ehsm\_gen\_key\_cmd, 144
- Reservel
  - HSM\_BootCfgType, 289
- ResetDisable
  - HSM\_BootCfgType, 289
- resultLength
  - Crypto\_PrimitiveInfoType, 50
- ret\_code
  - ehsm\_mbox\_cancel\_channel\_rps, 205
- Rev
  - HSM\_SecureUpgradeType\_, 336
- rev
  - ehsm\_change\_control\_field\_st, 66
  - ehsm\_gen\_sm9\_key\_param, 147
  - ehsm\_mailbox\_req, 201
  - ehsm\_mbox\_cancel\_channel\_req, 204
  - ehsm\_mbox\_cancel\_channel\_rps, 205
  - soc\_image\_upgrade\_input, 374
- rev1
  - ehsm\_cmd\_cipher\_st, 72
  - ehsm\_debug\_authentication\_cmd, 109
  - ehsm\_fw\_encrypt\_key\_cmd, 138
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, 149
  - ehsm\_get\_emu\_cmd, 154
  - ehsm\_image\_upgrade\_cmd, 164
  - ehsm\_image\_verfiy\_cmd, 167
  - ehsm\_import\_key\_cmd, 172
  - ehsm\_rng\_generate\_cmd, 220
  - ehsm\_secure\_boot\_st, 230
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, 246
  - ehsm\_sm9\_exchg\_key\_cmd, 247
  - ehsm\_sm9\_export\_key\_cmd, 253
  - ehsm\_sm9\_import\_key\_cmd, 260
  - ehsm\_sm9\_unwrap\_key\_cmd, 265
  - ehsm\_sm9\_wrap\_key\_cmd, 269
  - ehsm\_soc\_image\_verify\_st, 277
- rev2
  - ehsm\_debug\_authentication\_cmd, 109
  - ehsm\_fw\_encrypt\_key\_cmd, 138
  - ehsm\_get\_emu\_cmd, 154
  - ehsm\_image\_upgrade\_cmd, 164
  - ehsm\_image\_verfiy\_cmd, 167
  - ehsm\_rng\_generate\_cmd, 220
  - ehsm\_sm9\_exchg\_key\_cmd, 247
  - ehsm\_sm9\_wrap\_key\_cmd, 269
- rev3
  - ehsm\_fw\_encrypt\_key\_cmd, 138
  - ehsm\_get\_emu\_cmd, 154
  - ehsm\_image\_upgrade\_cmd, 165
  - ehsm\_image\_verfiy\_cmd, 167
  - ehsm\_rng\_generate\_cmd, 221
  - ehsm\_sm9\_exchg\_key\_cmd, 248
  - ehsm\_sm9\_wrap\_key\_cmd, 269
- rev4
  - ehsm\_image\_upgrade\_cmd, 165
  - ehsm\_image\_verfiy\_cmd, 168
  - ehsm\_rng\_generate\_cmd, 221
  - ehsm\_sm9\_exchg\_key\_cmd, 248
  - ehsm\_sm9\_wrap\_key\_cmd, 269
- rev5
  - ehsm\_image\_verfiy\_cmd, 168
- rev\_key\_auth
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, 149
  - ehsm\_image\_upgrade\_cmd, 165
  - ehsm\_image\_verfiy\_cmd, 168
  - ehsm\_sm9\_export\_key\_cmd, 254
  - ehsm\_sm9\_import\_key\_cmd, 260

- ehsm\_sm9\_unwrap\_key\_cmd, 265
- rev\_key\_auth\_addr
  - ehsm\_get\_emu\_cmd, 155
- rev\_key\_auth\_size
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, 149
  - ehsm\_get\_emu\_cmd, 155
  - ehsm\_image\_upgrade\_cmd, 165
  - ehsm\_image\_verfiy\_cmd, 168
  - ehsm\_rng\_generate\_cmd, 221
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, 246
  - ehsm\_sm9\_export\_key\_cmd, 254
  - ehsm\_sm9\_import\_key\_cmd, 260
  - ehsm\_sm9\_unwrap\_key\_cmd, 265
  - ehsm\_sm9\_wrap\_key\_cmd, 269
- rev\_key\_auth\_value
  - ehsm\_rng\_generate\_cmd, 221
- rev\_key\_handle
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, 150
  - ehsm\_get\_emu\_cmd, 155
  - ehsm\_image\_upgrade\_cmd, 165
  - ehsm\_image\_verfiy\_cmd, 168
  - ehsm\_rng\_generate\_cmd, 221
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, 246
  - ehsm\_sm9\_import\_key\_cmd, 260
  - ehsm\_sm9\_wrap\_key\_cmd, 269
- role
  - ehsm\_exchange\_sm9\_key\_param, 127
  - ehsm\_sm9\_exchg\_key\_cmd, 248
- rps\_data
  - ehsm\_cmd\_req, 86
- Rsa
  - Hsm\_PubKeyDataType\_, 327
- rsa
  - ehsm\_pubkey\_data\_, 219
- rsa crt
  - ehsm\_prikey\_data\_, 217
- rsa crt mode
  - cipher\_session\_st, 18
  - ehsm\_cmd\_hdr\_pke\_st, 77
- rsa\_d
  - ehsm\_prikey\_data\_, 217
- rsa\_d\_size
  - ehsm\_rsa\_key\_size\_, 225
- rsa\_dh\_g\_size
  - ehsm\_rsa\_dh\_key\_size\_, 223
- rsa\_dh\_p\_size
  - ehsm\_rsa\_dh\_key\_size\_, 224
- rsa\_dh\_q\_size
  - ehsm\_rsa\_dh\_key\_size\_, 224
- rsa\_e\_bit\_size
  - ehsm\_gen\_key\_param\_st, 146
- rsa\_e\_bytes\_size
  - eHSM\_If\_Evita\_Types\_Ip.h, 712
  - ehsm\_export\_pub\_key\_, 132
  - key\_info\_st, 347
- rsa\_e\_size
  - ehsm\_rsa\_key\_size\_, 225
- rsa\_n\_size
  - ehsm\_rsa\_key\_size\_, 225
- RsaCtr
  - Hsm\_PriKeyDataType\_, 326
- rsacipher\_testvec, 356
  - c, 356
  - c\_size, 357
  - d, 357
  - d\_byte\_size, 357
  - dp, 357
  - dp\_byte\_size, 357
  - dq, 357
  - dq\_byte\_size, 358
  - e, 358
  - e\_byte\_size, 358
  - hash\_alg, 358
  - is crt mode, 358
  - m, 358
  - m\_size, 359
  - n, 359
  - n\_byte\_size, 359
  - p, 359
  - p\_byte\_size, 359
  - public\_key\_vec, 359
  - q, 360
  - q\_byte\_size, 360
  - salt, 360
  - salt\_byte\_size, 360
  - siggen\_sigver\_test, 360
  - u, 360
  - u\_byte\_size, 361
- RsaD
  - Hsm\_PriKeyDataType\_, 326
- rsphdl
  - ehsm\_service, 233
- s1\_s2
  - ehsm\_exchange\_sm9\_key\_param, 127
  - ehsm\_sm9\_exchg\_key\_cmd\_child, 250
- s1\_s2\_value
  - ehsm\_create\_dh\_sm2\_ext\_param, 98
  - sm2\_ext\_param, 363
- S2H\_SRV\_CMD\_CANCEL\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 857
- S2H\_SRV\_CMD\_CANCEL\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 857
- S2H\_SRV\_CMD\_CANCEL\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 857
- S2H\_SRV\_CMD\_JTAG\_END\_BIT
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 858
- S2H\_SRV\_CMD\_JTAG\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 858
- S2H\_SRV\_CMD\_JTAG\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 858
- S2H\_SRV\_GENERAL\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 858
- S2H\_SRV\_GENERAL\_WORD\_INDEX
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 858
- S2H\_SRV\_GENERAL\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 858
- S2H\_SRV\_JTAG\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 859
- S2H\_SRV\_MGR\_NOTE\_BIT
  - eHSM\_Mailbox\_Prtcl\_Ip.h, 859
- S2H\_SRV\_MGR\_WORD\_INDEX

- eHSM\_Mailbox\_Prtcl\_Ip.h, [859](#)
- S2H\_SRV\_MGR\_WORD\_SIZE
  - eHSM\_Mailbox\_Prtcl\_Ip.h, [859](#)
- s2h\_note\_bit
  - mailbox\_channel, [353](#)
- s2h\_start\_addr
  - mailbox\_channel, [353](#)
- s2h\_word\_size
  - mailbox\_channel, [353](#)
- SECURE\_BOOT\_TYPE\_IMAGE\_VERIFY
  - eHSM\_Com\_Struct\_Ip.h, [420](#)
- SECURE\_BOOT\_TYPE\_SECURE\_BOOT
  - eHSM\_Com\_Struct\_Ip.h, [420](#)
- SHE\_LOG\_DEBUG
  - eHSM\_Debug\_Ip.h, [494](#)
- SHE\_LOG\_ERROR
  - eHSM\_Debug\_Ip.h, [494](#)
- SHE\_LOG\_INFO
  - eHSM\_Debug\_Ip.h, [495](#)
- SHE\_LOG\_WARN
  - eHSM\_Debug\_Ip.h, [495](#)
- SHE\_PURE\_LOG\_DEBUG
  - eHSM\_Debug\_Ip.h, [495](#)
- SKE\_CTX\_BUF\_SIZE
  - eHSM\_If\_Evita\_Types\_Ip.h, [702](#)
- SM2\_PUBLIC\_KEY\_SIZE
  - eHSM\_If\_Ext\_Types\_Ip.h, [754](#)
- SM2\_S1\_S2\_SIZE
  - eHSM\_If\_Ext\_Types\_Ip.h, [754](#)
- SM2\_WITH\_SM4\_CBC\_UPGRADE
  - Hsm\_Hal.h, [959](#)
- SM2\_WITH\_SM4\_CBC\_VERIFY
  - Hsm\_Hal.h, [959](#)
- SM4\_CMACE\_WITH\_SM4\_CBC\_UPGRADE
  - Hsm\_Hal.h, [960](#)
- SM4\_CMACE\_WITH\_SM4\_CBC\_VERIFY
  - Hsm\_Hal.h, [960](#)
- SM4\_CTRDRBG
  - eHSM\_If\_She\_Ip.h, [772](#)
- SOC\_BOOT\_TYPE\_PARALLEL
  - eHSM\_Com\_Struct\_Ip.h, [420](#)
- SOC\_BOOT\_TYPE\_SEQUENTIAL
  - eHSM\_Com\_Struct\_Ip.h, [420](#)
- SOC\_CMD\_GET\_HSM\_FW\_VERSION
  - AC784xx\_Hsm\_Reg.h, [390](#)
- SOC\_CMD\_IMAGE\_UPGRADE\_UPGRADE
  - AC784xx\_Hsm\_Reg.h, [390](#)
- SOC\_CMD\_IMAGE\_VERIFY
  - AC784xx\_Hsm\_Reg.h, [390](#)
- SOC\_CODE\_VERIFY\_FALG
  - eHSM\_Com\_Struct\_Ip.h, [421](#)
- STATUS\_BASE
  - eHSM\_Mailbox\_Reg\_Ip.h, [876](#)
- sa\_sb
  - ehsm\_exchange\_sm9\_key\_param, [128](#)
  - ehsm\_sm9\_exchg\_key\_cmd\_child, [250](#)
- sa\_sb\_value
  - ehsm\_create\_dh\_sm2\_ext\_param, [98](#)
  - sm2\_ext\_param, [363](#)
- salt
  - kdf\_testvec, [342](#)
- rsacipher\_testvec, [360](#)
- salt\_byte\_size
  - rsacipher\_testvec, [360](#)
- salt\_data
  - ehsm\_derive\_key\_cmd, [113](#)
  - ehsm\_key\_derived\_param, [180](#)
- salt\_size
  - ehsm\_derive\_key\_cmd, [113](#)
  - ehsm\_key\_derived\_param, [180](#)
  - kdf\_testvec, [343](#)
- SaltData
  - HSM\_DeriveKeyCfgType, [295](#)
- SaltDataSize
  - HSM\_DeriveKeyCfgType, [295](#)
- sec\_input\_addr
  - ehsm\_cmd\_cipher\_st, [72](#)
- sec\_input\_size
  - ehsm\_cmd\_cipher\_st, [73](#)
- secondaryFamily
  - Crypto\_AlgorithmInfoType, [23](#)
- secondaryInputKeyElementId
  - Crypto\_JobRedirectionInfoType, [39](#)
- secondaryInputKeyId
  - Crypto\_JobRedirectionInfoType, [39](#)
- secondaryInputLength
  - Crypto\_JobPrimitiveInputOutputType, [36](#)
- secondaryInputPtr
  - Crypto\_JobPrimitiveInputOutputType, [36](#)
- secondaryOutputKeyElementId
  - Crypto\_JobRedirectionInfoType, [39](#)
- secondaryOutputKeyId
  - Crypto\_JobRedirectionInfoType, [40](#)
- secondaryOutputLengthPtr
  - Crypto\_JobPrimitiveInputOutputType, [37](#)
- secondaryOutputPtr
  - Crypto\_JobPrimitiveInputOutputType, [37](#)
- secret
  - kpp\_testvec, [351](#)
- secret\_size
  - kpp\_testvec, [351](#)
- SecretKeyBlob
  - HSM\_SecretKeyCfgType, [333](#)
- secure\_flag
  - ehsm\_she\_key\_st, [241](#)
- secureboot
  - ehsm\_key\_usages\_st, [192](#)
  - HSM\_KeyActUseFlagsType, [312](#)
  - HSM\_KeyUsagesType, [320](#)
  - key\_act\_use\_flags\_t, [344](#)
- securestorage
  - ehsm\_key\_usages\_st, [192](#)
  - HSM\_KeyActUseFlagsType, [312](#)
  - HSM\_KeyUsagesType, [320](#)
  - key\_act\_use\_flags\_t, [344](#)
- self\_id
  - ehsm\_exchange\_sm9\_key\_param, [128](#)
  - ehsm\_sm9\_exchg\_key\_cmd\_child, [250](#)
- self\_id\_size
  - ehsm\_exchange\_sm9\_key\_param, [128](#)
  - ehsm\_sm9\_exchg\_key\_cmd\_child, [250](#)
- self\_test\_cmd



- ehsm\_mbox\_mgr\_channel\_req, [207](#)
- sender\_tmp\_pri\_key
  - ecies\_testvec, [62](#)
- sender\_tmp\_pri\_key\_sz
  - ecies\_testvec, [62](#)
- sensor\_resp\_init\_cmd
  - ehsm\_mbox\_mgr\_channel\_req, [207](#)
- service
  - Crypto\_PrimitiveInfoType, [50](#)
- service\_ctx
  - ehsm\_service\_info, [235](#)
- service\_id
  - ehsm\_service, [234](#)
- service\_reqhdl
  - eHSM\_Srv\_Mgr\_lp.h, [915](#)
- service\_rsphdl
  - eHSM\_Srv\_Mgr\_lp.h, [915](#)
- service\_table
  - eHSM\_Srv\_Mgr\_lp.c, [907](#)
- session\_id
  - ehsm\_ctx\_session\_st, [106](#)
- session\_status\_e
  - eHSM\_If\_Evita\_Types\_lp.h, [707](#)
- Set\_UTC\_Time
  - eHSM\_If\_Evita\_lp.h, [659](#)
- set\_baudrate\_cmd
  - ehsm\_mbox\_mgr\_channel\_req, [207](#)
- SetKeyBlobSize
  - HSM\_SecretKeyCfgType, [333](#)
- SetKeyUseFlag
  - HSM\_SecretKeyCfgType, [334](#)
- setauthsize\_error
  - aead\_testvec, [10](#)
- setkey\_error
  - aead\_testvec, [10](#)
  - cipher\_testvec, [21](#)
  - hash\_testvec, [284](#)
- shared\_info
  - kdf\_testvec, [343](#)
- shared\_info1
  - ecies\_testvec, [63](#)
- shared\_info1\_bytes
  - ecies\_testvec, [63](#)
- shared\_info2
  - ecies\_testvec, [63](#)
- shared\_info2\_bytes
  - ecies\_testvec, [63](#)
- shared\_info\_size
  - kdf\_testvec, [343](#)
- she\_boot\_failure
  - eHSM\_If\_She\_lp.c, [759](#)
  - eHSM\_If\_She\_lp.h, [773](#)
- she\_boot\_ok
  - eHSM\_If\_She\_lp.c, [759](#)
  - eHSM\_If\_She\_lp.h, [773](#)
- she\_crypto\_cbc
  - eHSM\_If\_She\_lp.c, [760](#)
  - eHSM\_If\_She\_lp.h, [774](#)
- she\_crypto\_ecb
  - eHSM\_If\_She\_lp.c, [760](#)
  - eHSM\_If\_She\_lp.h, [775](#)
- she\_crypto\_ecb\_extend
  - eHSM\_If\_She\_lp.c, [761](#)
  - eHSM\_If\_She\_lp.h, [775](#)
- she\_debug
  - eHSM\_If\_She\_lp.c, [761](#)
  - eHSM\_If\_She\_lp.h, [776](#)
- she\_export\_ram\_key
  - eHSM\_If\_She\_lp.c, [762](#)
  - eHSM\_If\_She\_lp.h, [776](#)
- she\_ext\_flag
  - crypto\_she\_key, [52](#)
  - ehsm\_she\_key\_host\_param, [239](#)
  - ehsm\_she\_key\_param, [240](#)
  - ehsm\_she\_load\_export\_key\_cmd, [243](#)
- she\_extend\_seed
  - eHSM\_If\_She\_lp.c, [763](#)
  - eHSM\_If\_She\_lp.h, [777](#)
- she\_generate\_mac
  - eHSM\_If\_She\_lp.c, [763](#)
  - eHSM\_If\_She\_lp.h, [778](#)
- she\_get\_id
  - eHSM\_If\_She\_lp.c, [765](#)
  - eHSM\_If\_She\_lp.h, [778](#)
- she\_get\_status
  - eHSM\_If\_She\_lp.c, [766](#)
  - eHSM\_If\_She\_lp.h, [779](#)
- she\_init\_rng
  - eHSM\_If\_She\_lp.c, [766](#)
  - eHSM\_If\_She\_lp.h, [779](#)
- she\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, [577](#)
- she\_load\_export\_key
  - ehsm\_mailbox\_req, [201](#)
- she\_load\_key
  - eHSM\_If\_She\_lp.c, [767](#)
  - eHSM\_If\_She\_lp.h, [780](#)
- she\_load\_key\_extend
  - eHSM\_If\_She\_lp.c, [768](#)
  - eHSM\_If\_She\_lp.h, [781](#)
- she\_load\_plain\_key
  - eHSM\_If\_She\_lp.c, [768](#)
  - eHSM\_If\_She\_lp.h, [782](#)
  - ehsm\_mailbox\_req, [201](#)
- she\_plain\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, [577](#)
- she\_rnd
  - eHSM\_If\_She\_lp.c, [769](#)
  - eHSM\_If\_She\_lp.h, [782](#)
- she\_secure\_boot
  - eHSM\_If\_She\_lp.c, [770](#)
  - eHSM\_If\_She\_lp.h, [783](#)
- she\_verify\_mac
  - eHSM\_If\_She\_lp.c, [770](#)
  - eHSM\_If\_She\_lp.h, [784](#)
- Sig
  - ehsm\_cmd\_sm9\_sig\_vry\_output\_ptr\_st, [88](#)
- sig
  - sm9cipher\_testvec, [367](#)
- siggen\_sigver\_test
  - akcipher\_testvec, [13](#)
  - rsacipher\_testvec, [360](#)

- sm9cipher\_testvec, [367](#)
- sign
  - ehsm\_key\_usages\_st, [193](#)
  - ehsm\_module\_status\_st, [211](#)
  - HSM\_KeyActUseFlagsType, [312](#)
  - HSM\_KeyUsagesType, [320](#)
  - key\_act\_use\_flags\_t, [345](#)
- Sign\_Finish
  - eHSM\_If\_Evita\_AsymCper\_Ip.c, [620](#)
  - eHSM\_If\_Evita\_Ip.h, [660](#)
- Sign\_Init
  - eHSM\_If\_Evita\_AsymCper\_Ip.c, [620](#)
  - eHSM\_If\_Evita\_Ip.h, [660](#)
- Sign\_Update
  - eHSM\_If\_Evita\_AsymCper\_Ip.c, [621](#)
  - eHSM\_If\_Evita\_Ip.h, [661](#)
- sign\_addr
  - ehsm\_debug\_authentication\_cmd, [109](#)
  - ehsm\_secure\_boot\_st, [230](#)
  - ehsm\_soc\_image\_verify\_st, [277](#)
- sign\_id
  - ehsm\_key\_signature\_, [185](#)
- sign\_info
  - ehsm\_key\_signature\_, [185](#)
- sign\_key\_id
  - ehsm\_key\_signature\_, [185](#)
- sign\_size
  - ehsm\_debug\_authentication\_cmd, [110](#)
  - ehsm\_module\_status\_st, [211](#)
  - ehsm\_secure\_boot\_st, [230](#)
  - ehsm\_soc\_image\_verify\_st, [277](#)
- SignAddr
  - HSM\_BootCfgType, [289](#)
- SignDir
  - HSM\_AsymCfgType, [286](#)
- SignInBuf
  - HSM\_InOutSignType, [309](#)
- SignInBufLen
  - HSM\_InOutSignType, [309](#)
- signatrue
  - ehsm\_key\_signature\_, [185](#)
  - ehsm\_module\_status\_cmd, [209](#)
  - ehsm\_she\_get\_id\_param\_st, [237](#)
- signatrue\_addr
  - ehsm\_get\_she\_id\_cmd, [160](#)
- signatrue\_size
  - ehsm\_get\_she\_id\_cmd, [161](#)
  - ehsm\_module\_status\_cmd, [209](#)
  - ehsm\_she\_get\_id\_param\_st, [237](#)
- Signature
  - HSM\_DebugAuthConfigType, [293](#)
- signature
  - ehsm\_debug\_auth\_st, [108](#)
  - signature\_t, [361](#)
- signature\_addr
  - cipher\_session\_st, [18](#)
- signature\_key\_elements
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, [578](#)
- signature\_size
  - cipher\_session\_st, [18](#)
  - ehsm\_debug\_auth\_st, [108](#)
  - signature\_t, [361](#)
- signature\_st
  - eHSM\_If\_Evita\_Types\_Ip.h, [705](#)
- signature\_t, [361](#)
  - signature, [361](#)
  - signature\_size, [361](#)
  - utc\_time, [362](#)
- SignatureSize
  - HSM\_DebugAuthConfigType, [293](#)
- size
  - ehsm\_change\_control\_field\_cmd, [65](#)
  - ehsm\_change\_control\_field\_st, [67](#)
  - ehsm\_fast\_cmac\_st, [135](#)
  - ehsm\_get\_challenge\_st, [153](#)
  - ehsm\_otp\_read\_cmd, [212](#)
  - ehsm\_otp\_write\_cmd, [214](#)
  - ehsm\_sensor\_init\_param\_st, [232](#)
  - ehsm\_storage\_area\_param\_st, [279](#)
- size\_info
  - eHSM\_If\_Evita\_Types\_Ip.h, [712](#)
  - ehsm\_export\_pub\_key\_, [132](#)
- Ske\_CryptoPrimitives
  - eHSM\_If\_Asr\_KeyCfg\_Ip.c, [579](#)
- SlotIndex
  - HSM\_KeySlotInfoType, [317](#)
- SlotInfo
  - HSM\_FlashKeyType, [301](#)
- SlotValid
  - HSM\_KeySlotInfoType, [317](#)
- sm2\_ext\_para
  - ehsm\_create\_dh\_key\_param, [96](#)
- sm2\_ext\_param, [362](#)
  - ehsm\_create\_dh\_key\_cmd, [93](#)
  - local\_tmp\_key\_auth\_size, [362](#)
  - local\_tmp\_key\_auth\_value, [362](#)
  - local\_tmp\_key\_handle, [363](#)
  - peer\_pubkey, [363](#)
  - peer\_temp\_pubkey, [363](#)
  - s1\_s2\_value, [363](#)
  - sa\_sb\_value, [363](#)
  - sm2\_role, [363](#)
- sm2\_ext\_param\_st
  - eHSM\_Com\_Struct\_Ip.h, [425](#)
- sm2\_key\_exchange\_role\_e
  - eHSM\_Com\_Struct\_Ip.h, [432](#)
- sm2\_role
  - ehsm\_create\_dh\_sm2\_ext\_param, [98](#)
  - sm2\_ext\_param, [363](#)
- sm9\_exchg\_key
  - ehsm\_mailbox\_req, [202](#)
- sm9\_export\_key
  - ehsm\_mailbox\_req, [202](#)
- sm9\_import\_key
  - ehsm\_mailbox\_req, [202](#)
- sm9\_key\_type
  - ehsm\_gen\_sm9\_key\_param, [147](#)
- sm9\_remove\_key
  - ehsm\_mailbox\_req, [202](#)
- sm9\_unwrap\_key
  - ehsm\_mailbox\_req, [202](#)
- sm9\_wrap\_key



- ehsm\_mailbox\_req, 202
- sm9cipher\_testvec, 364
  - c, 364
  - c\_sz, 364
  - enc\_typed, 365
  - h, 365
  - hid, 365
  - id, 365
  - id\_sz, 365
  - K2len, 365
  - m, 366
  - m\_sz, 366
  - padding, 366
  - Ppub, 366
  - priv, 366
  - r, 366
  - r\_sz, 367
  - sig, 367
  - siggen\_sigver\_test, 367
- soc\_boot
  - ehsm\_mailbox\_req, 203
- soc\_image\_upgrade\_info, 367
  - check\_version\_flag, 368
  - cmd\_input, 368
  - header, 368
  - header\_size, 369
  - mac\_sign, 369
  - mac\_sign\_size, 369
  - process\_mode, 369
  - storage\_alg, 369
  - storage\_encryption\_flag, 369
  - storage\_image, 370
  - storage\_image\_size, 370
  - storage\_iv, 370
  - storage\_iv\_size, 370
  - upgrade\_alg, 370
  - upgrade\_decryption\_flag, 370
  - upgrade\_image, 371
  - upgrade\_image\_size, 371
  - upgrade\_iv, 371
  - upgrade\_iv\_size, 371
  - upgrade\_pubkey, 371
  - upgrade\_pubkey\_size, 371
  - upgrade\_sign, 372
  - upgrade\_sign\_size, 372
  - upgrade\_version, 372
  - upgrade\_version\_size, 372
- soc\_image\_upgrade\_input, 372
  - check\_version\_flag, 373
  - ctx\_addr, 373
  - ctx\_size, 373
  - header\_addr, 374
  - header\_size, 374
  - mac\_sign\_addr, 374
  - mac\_sign\_size, 374
  - process\_mode, 374
  - rev, 374
  - storage\_alg, 375
  - storage\_encryption\_flag, 375
  - storage\_image\_addr, 375
  - storage\_image\_size, 375
  - storage\_iv\_addr, 375
  - storage\_iv\_size, 375
  - upgrade\_alg, 376
  - upgrade\_decryption\_flag, 376
  - upgrade\_image\_addr, 376
  - upgrade\_image\_size, 376
  - upgrade\_iv\_addr, 376
  - upgrade\_iv\_size, 376
  - upgrade\_pubkey\_addr, 377
  - upgrade\_pubkey\_size, 377
  - upgrade\_sign\_addr, 377
  - upgrade\_sign\_size, 377
  - upgrade\_version\_addr, 377
  - upgrade\_version\_size, 377
- soc\_image\_verify
  - ehsm\_mailbox\_req, 203
- soc\_image\_verify\_info, 378
  - cmd\_input, 378
  - header, 378
  - header\_size, 379
  - pubkey, 379
  - pubkey\_size, 379
  - storage\_alg, 379
  - storage\_encryption\_flag, 379
  - storage\_image, 379
  - storage\_image\_size, 380
  - storage\_iv, 380
  - storage\_iv\_size, 380
  - storage\_sign, 380
  - storage\_sign\_size, 380
  - type, 380
  - update\_version\_flag, 381
  - version, 381
  - version\_size, 381
- soc\_image\_verify\_input, 381
  - header\_addr, 382
  - header\_size, 382
  - pubkey\_addr, 382
  - pubkey\_size, 382
  - storage\_alg, 382
  - storage\_encryption\_flag, 382
  - storage\_image\_addr, 383
  - storage\_image\_size, 383
  - storage\_iv\_addr, 383
  - storage\_iv\_size, 383
  - storage\_sign\_addr, 383
  - storage\_sign\_size, 383
  - type, 384
  - update\_version\_flag, 384
  - version\_addr, 384
  - version\_size, 384
- srv\_asr\_cancel\_cmd
  - eHSM\_Srv\_Ext\_Ip.c, 890
  - eHSM\_Srv\_Mgr\_Ip.c, 907
- srv\_bootloader\_cmd
  - eHSM\_Srv\_Ext\_Ip.c, 890
  - eHSM\_Srv\_Mgr\_Ip.c, 908
- srv\_certificate\_parse
  - eHSM\_Srv\_Key\_Ip.c, 899
  - eHSM\_Srv\_Mgr\_Ip.c, 908
- srv\_certificate\_verify

- eHSM\_Srv\_Key\_Ip.c, [899](#)
- eHSM\_Srv\_Mgr\_Ip.c, [908](#)
- srv\_change\_control\_field
  - eHSM\_Srv\_Ext\_Ip.c, [891](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [908](#)
- srv\_change\_lifecycle
  - eHSM\_Srv\_Ext\_Ip.c, [891](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [908](#)
- srv\_close\_debug
  - eHSM\_Srv\_Ext\_Ip.c, [891](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [908](#)
- srv\_create\_dh\_key
  - eHSM\_Srv\_Key\_Ip.c, [899](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [909](#)
- srv\_create\_random\_key
  - eHSM\_Srv\_Key\_Ip.c, [900](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [909](#)
- srv\_crypto\_hash
  - eHSM\_Srv\_Hash\_Ip.c, [897](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [909](#)
- srv\_crypto\_pke
  - eHSM\_Srv\_AsymCper\_Ip.c, [883](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [909](#)
- srv\_crypto\_randomgenerate
  - eHSM\_Srv\_Mgr\_Ip.c, [909](#)
  - eHSM\_Srv\_Rng\_Ip.c, [919](#)
- srv\_crypto\_randomseed
  - eHSM\_Srv\_Rng\_Ip.c, [919](#)
- srv\_crypto\_ske
  - eHSM\_Srv\_Ciper\_Ip.c, [884](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [909](#)
- srv\_debug\_auth
  - eHSM\_Srv\_Ext\_Ip.c, [892](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [910](#)
- srv\_derive\_key
  - eHSM\_Srv\_Key\_Ip.c, [900](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [910](#)
- srv\_export\_key
  - eHSM\_Srv\_Key\_Ip.c, [900](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [910](#)
- srv\_fw\_encrypt\_key
  - eHSM\_Srv\_Ext\_Ip.c, [892](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [910](#)
- srv\_fw\_get\_random\_key
  - eHSM\_Srv\_Ext\_Ip.c, [892](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [910](#)
- srv\_get\_challenge
  - eHSM\_Srv\_Ext\_Ip.c, [893](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [910](#)
- srv\_get\_module\_status
  - eHSM\_Srv\_Ext\_Ip.c, [893](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [911](#)
- srv\_get\_pub\_from\_priv
  - eHSM\_Srv\_Key\_Ip.c, [901](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [911](#)
- srv\_get\_she\_id
  - eHSM\_Srv\_Ext\_Ip.c, [893](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [911](#)
- srv\_get\_she\_status
  - eHSM\_Srv\_Ext\_Ip.c, [894](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [911](#)
- srv\_image\_upgrade
  - eHSM\_Srv\_Ext\_Ip.c, [894](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [911](#)
- srv\_image\_verify
  - eHSM\_Srv\_Ext\_Ip.c, [894](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [911](#)
- srv\_import\_key
  - eHSM\_Srv\_Key\_Ip.c, [901](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [912](#)
- srv\_key\_copy
  - eHSM\_Srv\_Key\_Ip.c, [901](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [912](#)
- srv\_key\_remove
  - eHSM\_Srv\_Key\_Ip.c, [902](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [912](#)
- srv\_key\_status
  - eHSM\_Srv\_Key\_Ip.c, [902](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [912](#)
- srv\_low\_power
  - eHSM\_Srv\_Ext\_Ip.c, [895](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [912](#)
- srv\_read\_otp\_data
  - eHSM\_Srv\_Ext\_Ip.c, [895](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [912](#)
- srv\_rng\_extend\_seed
  - eHSM\_Srv\_Rng\_Ip.c, [919](#)
- srv\_rng\_init
  - eHSM\_Srv\_Rng\_Ip.c, [920](#)
- srv\_self\_test
  - eHSM\_Srv\_Ext\_Ip.c, [895](#)
- srv\_set\_baudrate
  - eHSM\_Srv\_Ext\_Ip.c, [896](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [913](#)
- srv\_she\_candle\_cmd
  - eHSM\_Srv\_Ext\_Ip.c, [896](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [913](#)
- srv\_she\_load\_key
  - eHSM\_Srv\_Key\_Ip.c, [902](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [913](#)
- srv\_she\_load\_plain\_key
  - eHSM\_Srv\_Key\_Ip.c, [903](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [913](#)
- srv\_she\_ram\_key\_export
  - eHSM\_Srv\_Key\_Ip.c, [903](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [913](#)
- srv\_soc\_boot\_status
  - eHSM\_Srv\_Ext\_Ip.c, [896](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [913](#)
- srv\_write\_otp\_data
  - eHSM\_Srv\_Ext\_Ip.c, [897](#)
  - eHSM\_Srv\_Mgr\_Ip.c, [914](#)
- ss\_addr
  - ehsm\_create\_dh\_key\_cmd, [93](#)
  - ehsm\_create\_dh\_key\_param, [96](#)
- state
  - crypto\_object, [49](#)
- status
  - cipher\_session\_st, [18](#)
  - ehsm\_module\_status\_st, [211](#)
  - ehsm\_she\_get\_id\_param\_st, [237](#)
  - ehsm\_soc\_secure\_boot\_status\_st, [278](#)

- status\_addr
  - ehsm\_get\_she\_id\_cmd, 161
  - ehsm\_module\_status\_cmd, 209
- status\_size
  - ehsm\_get\_she\_id\_cmd, 161
  - ehsm\_module\_status\_cmd, 209
  - ehsm\_module\_status\_st, 211
  - ehsm\_she\_get\_id\_param\_st, 237
- StbWaitHSM
  - HSM\_BootCfgType, 289
- StbWaitVerify
  - HSM\_BootCfgType, 290
- Std\_HsmReturnTypes
  - Asr\_Standard\_Types.h, 402
- storage
  - ehsm\_image, 162
  - ehsm\_image\_verify\_st, 170
- storage\_addr
  - ehsm\_image\_upgrade\_cmd, 165
- storage\_alg
  - ehsm\_secure\_boot\_st, 230
  - ehsm\_soc\_image\_verify\_cmd, 273
  - ehsm\_soc\_image\_verify\_st, 277
  - soc\_image\_upgrade\_info, 369
  - soc\_image\_upgrade\_input, 375
  - soc\_image\_verify\_info, 379
  - soc\_image\_verify\_input, 382
- storage\_encryption\_flag
  - ehsm\_soc\_image\_verify\_cmd, 273
  - soc\_image\_upgrade\_info, 369
  - soc\_image\_upgrade\_input, 375
  - soc\_image\_verify\_info, 379
  - soc\_image\_verify\_input, 382
- storage\_image
  - soc\_image\_upgrade\_info, 370
  - soc\_image\_verify\_info, 379
- storage\_image\_addr
  - ehsm\_soc\_image\_verify\_cmd, 273
  - soc\_image\_upgrade\_input, 375
  - soc\_image\_verify\_input, 383
- storage\_image\_size
  - ehsm\_soc\_image\_verify\_cmd, 274
  - soc\_image\_upgrade\_info, 370
  - soc\_image\_upgrade\_input, 375
  - soc\_image\_verify\_info, 380
  - soc\_image\_verify\_input, 383
- storage\_info
  - key\_info\_st, 347
- storage\_iv
  - soc\_image\_upgrade\_info, 370
  - soc\_image\_verify\_info, 380
- storage\_iv\_addr
  - ehsm\_soc\_image\_verify\_cmd, 274
  - soc\_image\_upgrade\_input, 375
  - soc\_image\_verify\_input, 383
- storage\_iv\_size
  - ehsm\_soc\_image\_verify\_cmd, 274
  - soc\_image\_upgrade\_info, 370
  - soc\_image\_upgrade\_input, 375
  - soc\_image\_verify\_info, 380
  - soc\_image\_verify\_input, 383
- storage\_key\_type
  - key\_info\_st, 347
- storage\_key\_type\_e
  - eHSM\_If\_Evita\_Types\_lp.h, 707
- storage\_sign
  - soc\_image\_verify\_info, 380
- storage\_sign\_addr
  - ehsm\_soc\_image\_verify\_cmd, 274
  - soc\_image\_verify\_input, 383
- storage\_sign\_size
  - ehsm\_soc\_image\_verify\_cmd, 274
  - soc\_image\_verify\_info, 380
  - soc\_image\_verify\_input, 383
- storage\_size
  - ehsm\_image, 163
  - ehsm\_image\_verify\_st, 170
- StorageAlg
  - HSM\_ImageVerifyType\_, 305
  - HSM\_SecureUpgradeType\_, 336
- StorageEncryptionFlag
  - HSM\_ImageVerifyType\_, 305
  - HSM\_SecureUpgradeType\_, 336
- StorageImageAddr
  - HSM\_ImageVerifyType\_, 305
  - HSM\_SecureUpgradeType\_, 336
- StorageImageSize
  - HSM\_ImageVerifyType\_, 306
  - HSM\_SecureUpgradeType\_, 337
- StorageIvAddr
  - HSM\_ImageVerifyType\_, 306
  - HSM\_SecureUpgradeType\_, 337
- StorageIvSize
  - HSM\_ImageVerifyType\_, 306
  - HSM\_SecureUpgradeType\_, 337
- StorageSignAddr
  - HSM\_ImageVerifyType\_, 306
- StorageSignSize
  - HSM\_ImageVerifyType\_, 306
- surpport\_asyn
  - mailbox\_channel, 354
- SwitchClock
  - HSM\_BootCfgType, 290
- sym\_k
  - ehsm\_prikey\_data\_, 217
- SymAlgo
  - HSM\_CMacCfgType, 291
  - HSM\_SymCfgType, 341
- SymKey
  - Hsm\_PriKeyDataType\_, 326
- Sync
  - HSM\_AsymCfgType, 287
  - HSM\_CMacCfgType, 292
  - HSM\_HMacCfgType, 303
  - HSM\_SymCfgType, 341
- tag\_ptr
  - ehsm\_cmd\_aead\_ptr\_st, 70
- tag\_size
  - ehsm\_cmd\_hdr\_ske\_st, 81
- tap
  - hash\_testvec, 285

- target\_algorithm\_identifier
  - ehsm\_create\_dh\_key\_param, 96
  - ehsm\_create\_random\_key\_param, 102
- target\_key\_handle
  - ehsm\_key\_copy\_param, 177
- target\_key\_id
  - ehsm\_key\_signature\_, 185
- targetCrylffKeyld
  - Crypto\_JobPrimitiveInputOutputType, 37
- TargetKeyld
  - HSM\_KeyStatusType, 319
- tertiaryInputKeyElementld
  - Crypto\_JobRedirectionInfoType, 40
- tertiaryInputKeyld
  - Crypto\_JobRedirectionInfoType, 40
- tertiaryInputLength
  - Crypto\_JobPrimitiveInputOutputType, 37
- tertiaryInputPtr
  - Crypto\_JobPrimitiveInputOutputType, 37
- test\_int\_config.h, 994
  - CONFIG\_EHSM\_UNIT\_TEST\_EVITA\_SYM, 994
- test\_type
  - ehsm\_self\_test\_cmd, 231
- tick\_accuracy
  - ehsm\_tick\_value, 280
- tick\_length
  - ehsm\_tick\_value, 281
- time\_stamp
  - cipher\_session\_st, 18
  - ehsm\_cmd\_hdr\_pke\_st, 77
  - ehsm\_cmd\_hdr\_ske\_st, 82
- time\_stamp\_st
  - eHSM\_If\_Evita\_Types\_lp.h, 705
- timeout
  - ehsm\_cmd\_req, 87
  - ehsm\_service, 234
- timestamp
  - ehsm\_key\_usages\_st, 193
  - HSM\_KeyActUseFlagsType, 313
  - HSM\_KeyUsagesType, 321
  - key\_act\_use\_flags\_t, 345
- transport
  - ehsm\_key\_usages\_st, 193
  - HSM\_KeyActUseFlagsType, 313
  - HSM\_KeyUsagesType, 321
  - key\_act\_use\_flags\_t, 345
- transport\_key\_auth\_size
  - ehsm\_export\_key\_cmd, 131
  - ehsm\_import\_key\_cmd, 172
- transport\_key\_auth\_value
  - ehsm\_export\_key\_cmd, 131
  - ehsm\_import\_key\_cmd, 173
- transport\_key\_auth\_size
  - ehsm\_evita\_key\_export, 121
  - ehsm\_evita\_key\_import\_st, 124
- transport\_key\_auth\_value
  - ehsm\_evita\_key\_export, 121
  - ehsm\_evita\_key\_import\_st, 124
- transport\_key\_authorization
  - crypto\_import\_evita\_key\_info, 32
  - crypto\_key\_export\_info, 46
- transport\_key\_authorization\_size
  - crypto\_import\_evita\_key\_info, 32
  - crypto\_key\_export\_info, 46
- transport\_key\_handle
  - crypto\_import\_evita\_key\_info, 32
  - crypto\_key\_export\_info, 46
  - ehsm\_evita\_key\_export, 121
  - ehsm\_evita\_key\_import\_st, 124
  - ehsm\_export\_key\_cmd, 131
  - ehsm\_import\_key\_cmd, 173
- Trng\_CryptoPrimitives
  - eHSM\_If\_Asr\_KeyCfg\_lp.c, 579
- trnsp\_flags
  - ehsm\_key\_flags\_element\_st, 182
  - HSM\_KeyFlagsElementType, 314
- true
  - eHSM\_Types\_lp.h, 921
- Type
  - HSM\_DebugAuthConfigType, 293
  - HSM\_ImageVerifyType\_, 306
- type
  - crypto\_create\_evita\_key\_info, 27
  - crypto\_import\_evita\_key\_info, 32
  - crypto\_object, 49
  - ehsm\_change\_control\_field\_cmd, 66
  - ehsm\_change\_control\_field\_st, 67
  - ehsm\_change\_lifecycle\_cmd, 68
  - ehsm\_close\_debug\_cmd, 68
  - ehsm\_cmd\_hdr\_eccp\_keygen\_st, 74
  - ehsm\_cmd\_hdr\_rsa\_keygen\_st, 80
  - ehsm\_create\_dh\_key\_cmd, 93
  - ehsm\_create\_dh\_key\_param, 97
  - ehsm\_create\_evita\_key\_param, 100
  - ehsm\_create\_random\_key\_param, 102
  - ehsm\_debug\_auth\_st, 108
  - ehsm\_debug\_authentication\_cmd, 110
  - ehsm\_derive\_key\_cmd, 113
  - ehsm\_evita\_key\_import\_st, 124
  - ehsm\_exchange\_sm9\_key\_param, 128
  - ehsm\_gen\_key\_cmd, 144
  - ehsm\_gen\_sm9\_userpriv\_key\_cmd, 150
  - ehsm\_gen\_sm9\_userpriv\_key\_param, 151
  - ehsm\_get\_challenge\_cmd, 152
  - ehsm\_get\_challenge\_st, 153
  - ehsm\_image\_verfiy\_cmd, 168
  - ehsm\_image\_verify\_st, 170
  - ehsm\_key\_derived\_param, 180
  - ehsm\_module\_status\_cmd, 209
  - ehsm\_module\_status\_st, 211
  - ehsm\_secure\_boot\_st, 230
  - ehsm\_sm9\_exchg\_gen\_usertmp\_cmd, 246
  - ehsm\_sm9\_exchg\_key\_cmd, 248
  - ehsm\_sm9\_exckey\_gen\_tmpkey\_param, 252
  - ehsm\_sm9\_import\_key\_cmd, 260
  - ehsm\_sm9\_inexport\_key\_param, 262
  - ehsm\_soc\_image\_verify\_cmd, 274
  - ehsm\_soc\_image\_verify\_st, 277
  - mailbox\_channel, 354
  - soc\_image\_verify\_info, 380
  - soc\_image\_verify\_input, 384
- Typeld

- CryptoKey, 54
- u
  - ehsm\_rsa\_cert\_param\_, 223
  - Hsm\_RsaCertType\_, 331
  - rsacipher\_testvec, 360
- u\_byte\_size
  - rsacipher\_testvec, 361
- u\_hdr
  - ehsm\_cmd\_cipher\_st, 73
- UPGRADE\_VALID\_FLAG
  - eHSM\_Com\_Struct\_Ip.h, 421
- uart\_cmd
  - ehsm\_mailbox\_req, 203
- uart\_cmd\_buffer
  - ehsm\_uart\_cmd, 281
- update\_version\_flag
  - soc\_image\_verify\_info, 381
  - soc\_image\_verify\_input, 384
- UpdateVersionFlag
  - HSM\_ImageVerifyType\_, 307
- upgrade\_alg
  - soc\_image\_upgrade\_info, 370
  - soc\_image\_upgrade\_input, 376
- upgrade\_decryption\_flag
  - soc\_image\_upgrade\_info, 370
  - soc\_image\_upgrade\_input, 376
- upgrade\_image
  - soc\_image\_upgrade\_info, 371
- upgrade\_image\_addr
  - soc\_image\_upgrade\_input, 376
- upgrade\_image\_size
  - soc\_image\_upgrade\_info, 371
  - soc\_image\_upgrade\_input, 376
- upgrade\_iv
  - soc\_image\_upgrade\_info, 371
- upgrade\_iv\_addr
  - soc\_image\_upgrade\_input, 376
- upgrade\_iv\_size
  - soc\_image\_upgrade\_info, 371
  - soc\_image\_upgrade\_input, 376
- upgrade\_pubkey
  - soc\_image\_upgrade\_info, 371
- upgrade\_pubkey\_addr
  - soc\_image\_upgrade\_input, 377
- upgrade\_pubkey\_size
  - soc\_image\_upgrade\_info, 371
  - soc\_image\_upgrade\_input, 377
- upgrade\_sign
  - soc\_image\_upgrade\_info, 372
- upgrade\_sign\_addr
  - soc\_image\_upgrade\_input, 377
- upgrade\_sign\_size
  - soc\_image\_upgrade\_info, 372
  - soc\_image\_upgrade\_input, 377
- upgrade\_version
  - soc\_image\_upgrade\_info, 372
- upgrade\_version\_addr
  - soc\_image\_upgrade\_input, 377
- upgrade\_version\_size
  - soc\_image\_upgrade\_info, 372
- soc\_image\_upgrade\_input, 377
- UpgradeAlg
  - HSM\_SecureUpgradeType\_, 337
- UpgradeDecryptionFlag
  - HSM\_SecureUpgradeType\_, 337
- UpgradeImageAddr
  - HSM\_SecureUpgradeType\_, 337
- UpgradeImageSize
  - HSM\_SecureUpgradeType\_, 338
- UpgradeIvAddr
  - HSM\_SecureUpgradeType\_, 338
- UpgradeIvSize
  - HSM\_SecureUpgradeType\_, 338
- UpgradePubkeyAddr
  - HSM\_SecureUpgradeType\_, 338
- UpgradePubkeySize
  - HSM\_SecureUpgradeType\_, 338
- UpgradeSignAddr
  - HSM\_SecureUpgradeType\_, 338
- UpgradeSignSize
  - HSM\_SecureUpgradeType\_, 339
- UpgradeVersionAddr
  - HSM\_SecureUpgradeType\_, 339
- UpgradeVersionSize
  - HSM\_SecureUpgradeType\_, 339
- use\_flags
  - crypto\_key\_export\_info, 46
  - ehsm\_evita\_key\_export, 121
  - ehsm\_export\_key\_cmd, 131
  - ehsm\_key\_flags\_element\_st, 182
  - HSM\_KeyFlagsElementType, 315
- Used
  - HSM\_RamKeyInfoType, 330
- user\_id
  - ehsm\_sm9\_gen\_tmp\_pubkey\_param, 256
  - ehsm\_sm9\_get\_tmp\_pubkey\_cmd, 258
  - ehsm\_sm9\_wrap\_key\_param, 271
- user\_id\_size
  - ehsm\_gen\_sm9\_userpriv\_key\_param, 151
  - ehsm\_sm9\_get\_tmp\_pubkey\_cmd, 258
- user\_id\_value
  - ehsm\_gen\_sm9\_userpriv\_key\_param, 151
- user\_priv\_key\_handle
  - ehsm\_exchange\_sm9\_key\_param, 128
  - ehsm\_sm9\_exchg\_key\_cmd, 248
  - ehsm\_sm9\_unwrap\_key\_cmd, 265
  - ehsm\_sm9\_unwrap\_key\_param, 267
- user\_tmp\_key\_handle
  - ehsm\_exchange\_sm9\_key\_param, 128
  - ehsm\_sm9\_exchg\_key\_cmd, 248
- utc\_time
  - cipher\_session\_st, 18
  - hash\_hmac\_t, 282
  - mac\_t, 352
  - signature\_t, 362
- utcsync
  - ehsm\_key\_usages\_st, 193
  - HSM\_KeyActUseFlagsType, 313
  - HSM\_KeyUsagesType, 321
  - key\_act\_use\_flags\_t, 345
- utest\_vecs\_st.h, 994

- MAX\_TAP, [994](#)
- valid
  - ehsm\_crypto\_key, [103](#)
- valid\_until
  - crypto\_create\_evita\_key\_info, [27](#)
  - ehsm\_create\_dh\_key\_cmd, [93](#)
  - ehsm\_derive\_key\_cmd, [113](#)
  - ehsm\_gen\_key\_cmd, [145](#)
- valid\_util
  - eHSM\_If\_Evita\_Types\_Ip.h, [712](#)
  - ehsm\_key\_attr\_data\_, [176](#)
  - ehsm\_key\_status\_, [187](#)
- ValidUntil
  - HSM\_DeriveKeyCfgType, [295](#)
  - HSM\_GenKeyCfgType, [302](#)
- ValidUtil
  - HSM\_PlainKeyCfgType, [325](#)
- value
  - ehsm\_change\_control\_field\_st, [67](#)
- value\_addr
  - ehsm\_change\_control\_field\_cmd, [66](#)
- verify
  - ehsm\_certificate\_verify\_st, [65](#)
  - ehsm\_key\_usages\_st, [193](#)
  - HSM\_KeyActUseFlagsType, [313](#)
  - HSM\_KeyUsagesType, [321](#)
  - key\_act\_use\_flags\_t, [345](#)
- Verify\_Finish
  - eHSM\_If\_Evita\_AsymCper\_Ip.c, [622](#)
  - eHSM\_If\_Evita\_Ip.h, [662](#)
- Verify\_Init
  - eHSM\_If\_Evita\_AsymCper\_Ip.c, [622](#)
  - eHSM\_If\_Evita\_Ip.h, [662](#)
- Verify\_Update
  - eHSM\_If\_Evita\_AsymCper\_Ip.c, [623](#)
  - eHSM\_If\_Evita\_Ip.h, [663](#)
- verifyPtr
  - Crypto\_JobPrimitiveInputOutputType, [37](#)
- VerifySize
  - HSM\_BootCfgType, [290](#)
- version
  - soc\_image\_verify\_info, [381](#)
- version\_addr
  - soc\_image\_verify\_input, [384](#)
- version\_size
  - soc\_image\_verify\_info, [381](#)
  - soc\_image\_verify\_input, [384](#)
- VersionAddr
  - HSM\_BootCfgType, [290](#)
  - HSM\_ImageVerifyType\_\_, [307](#)
- VersionSize
  - HSM\_ImageVerifyType\_\_, [307](#)
- VersionUpdateEn
  - HSM\_BootCfgType, [290](#)
- Vry
  - HSM\_InOutMacType, [308](#)
  - HSM\_InOutSignType, [309](#)
- WdgTimeout
  - HSM\_BootCfgType, [290](#)
- with\_pubkey
  - ehsm\_gen\_sm9\_userpriv\_key\_param, [151](#)
- wk
  - aead\_testvec, [11](#)
  - cipher\_testvec, [21](#)
- write\_data\_size
  - ehsm\_otp\_write\_param\_st, [215](#)
- X
  - eHSM\_If\_Evita\_Hash\_Ip.c, [629](#)