CWSE APDU Command Specification

文件編號：CW-SPEC-0002

版別(0110)

合約號碼：N/A

出版日期：2015/11/16

交付日期：N/A

製文單位：銓安智慧科技股份有限公司

CWSE APDU Command Specification	文件編號： CW-SPEC-0002 版別(0110) 出版日期： 2015/11/16

銓安智慧科技股份有限公司			
擬訂： (Prepared by)	姓名/職稱	日期	姓名/職稱	日期
委託方			
接收： (Received by)	姓名/職稱	日期	
審查： (Reviewed by)	姓名/職稱	日期	姓名/職稱	日期
	姓名/職稱	日期	姓名/職稱	日期
核定： (Approval by)	姓名/職稱	日期	

修訂紀錄

版本編號	修訂日期	修訂人	修訂內容
0100	2014/5/7	Roy Lin	Initial draft
0101	2014/6/14	Roy Lin	● Add the following commands: ■ se_init_set_data ■ se_init_get_data_hash ■ se_init_confirm ■ se_init_vmk_chlng ■ se_init_back_init ■ se_init_change_vmk ■ se_perso_set_data ■ se_perso_get_data_hash ■ se_perso_confirm ■ se_perso_back_perso ■ se_puk_chlng ■ se_pin_unlock ■ se_set_currency ■ se_get_currency ■ se_qry_wallet_id ■ se_trx_begin ■ se_trx_verify_otp ■ se_trx_sign ■ se_trx_finish ■ se_trx_status

			● Remove the following commands: ■ se_puk_change ■ se_puk_version ● Wallet balance: 16 bytes -> 8 bytes ● se_actv_wallet command: ■ wallet profile: name, addr, prk
0102	2014/6/30	Roy Lin	● se_perso_back_perso: ■ Add new pin hash setting ● se_pin_unlock ■ Add new pin hash setting ● Responses for authentication: ■ Hash (32 bytes) -> Encryption result (16 bytes) ■ se_init_back_init ■ se_init_change_vmk ■ se_pin_auth ■ se_pin_unlock ● trx_sign commands: 6xh -> 7xh
0103	2014/8/9	Roy Lin	● Add commands: ■ se_trx_prepare ■ se_trx_get_ctxinfo ■ se_qry_wallet_info (Remove se_qry_wallet_balance) ■ se_qry_wallet_info_len ■ se_wallet_pkg_balnc ■ se_wallet_get_map ■ se_get_card_name

			■ se_set_card_name ● Modified commands: ■ se_find_empty_wallet: WAID 1 byte -> 2 bytes, RFU parameter removed ■ se_qry_wallet_id: WAID 1 byte -> 2 bytes, RFU parameter removed ■ se_qry_wallet_id: wallet name hash -> wallet name ■ se_actv_wallet: wallet name hash -> wallet name, WAID 1 byte -> 2 bytes ■ se_deactv_wallet: WAID 1 byte -> 2 bytes ■ se_trx_begin: WAID 1 byte -> 2 bytes ■ se_get_currency and se_set_currency: currency 1 byte -> 5 bytes ● Instruction IDs for trx signing commands are changed
0104	2014/8/30	Roy Lin	● Add commands: ■ se_actv_wallet_genkey ■ se_get_wapkg_name ■ se_set_wapkg_name ● Modify commands: ■ se_trx_get_ctxinfo: ◆ Add IN_ID parameter ◆ Modify INFOID ◆ num_in: 1 byte -> 2 bytes

			■ se_trx_prepare: Add parameter WAID, BALANCE and HASH ■ se_trx_begin: Remove parameter WAID and add parameter AMOUNT ● Trx signing input limit: 10 -> 256
0105	2014/10/15	Roy Lin	● Add commands: ■ Backup-restore commands ■ HDW commands ■ Host binding commands ● PINHASH is encrypted by BIND_CHAN in se_perso_back_perso and se_pin_unlock ● HASH value -> MAC value for the following commands: ■ se_pin_change ■ se_actv_wallet ■ se_qry_wallet_info ■ se_wallet_get_map ■ se_actv_wallet_genkey ■ se_trx_prepare ■ se_trx_sign ● WAPRK is encrypted by BIND_CHAN in se_actv_wallet ● WAADDR: 32 bytes -> 25 bytes in se_actv_wallet and se_qry_wallet_info (For consistency) ● Instruction ID change: ■ se_trx_status: 70h -> 80h

			■ se_bak_status: 80h -> 88h
0106	2014/11/18	Roy Lin	● se_pin_auth and se_pin_change: can be executed if PIN function is not enabled in security policy ● hdw_init_wallet_gen: Add parameter SEEDNUM, ACTVCODE, MAC ● hdw_qry_wa_info,hdw_qry_acc_info: don't need pin auth ● add hdw_init_wallet_gen_confirm ● hdw_prep_trx_sign: add parameter balance ● Add command: ■ se_hdw_qry_acc_keyinfo ■ se_set_secpo ● Remove HST_ID parameter in se_bind_logout
0107	2015/1/13	Roy Lin	● Remove wallet_mgmt commands ● Remove backup/restore commands ● Modified commands: ■ se_hdw_init_wallet ■ se_hdw_create_account ■ se_trx_get_ctxinfo: ◆ Output info_hash -> info ● se_bind_find_hst_id: Only supported in NOHOST mode ● New commands: ■ se_get_card_id ■ se_xchs_reg_status ■ se_xchs commands

			● OTP: 8 digits -> 6 digits ■ se_bind_reg_init ■ se_trx_begin ■ se_xchs_get_otp
0108	2015/6/29	Roy Lin	● Add command se_get_card_id ● Add card ID and XCHS SEMK in initialization data ● Add credential commands ● Add init data IDs: XCHS_SEMK, CARD_ID, XCHS_OTPK, XCHS_SMK ● Add command se_trx_outaddr ● Modify command se_trx_begin (Add enc_outaddr parameter) ● Binding host description: 128 bytes → 64 bytes ● Add error code list ● Fix some typos
0109	2015/8/14	Roy Lin	● Add se_get_basic_info command ● Add all_hdw_info in se_hdw_qry_wa_info command ● Add all_acc_info in se_hdw_qry_acc_info command
0110	2015/11/16	Roy Lin	● Add account public key parameter in se_hdw_qry_acc_keyinfo

目　錄

1. Introduction

SE (Secure Element) is cryptographic engine and secure storage of key and data for BCDC. It provides ISO and SPI communication interface. For SPI interface introduction, please refer to "*BCDC SE SPI Slave User Manual*". This documents list all supported APDU commands and specifications of them in ISO and SPI interface.

2. CWSE APDU Command List

Category: SE Info				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_get_mode_state	10	V	V	Get SE mode and state
se_get_fw_version	11	V	V	Get SE firmware version
se_get_unique_id	12	V	V	Get SE unique ID
se_get_mod_err	13	V	V	Get internal module error
se_get_basic_info	14	V	V	Get SE basic info
se_back_ikvldr	78	V	V	Back to IKV loader
Category: Init				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_init_set_data	A0	V	V	Set init data
se_init_get_data_hash	A1	V	V	Get init data hash
se_init_confirm	A2	V	V	Confirm init data
se_init_vmk_chlng	A3	V	V	Get VMK challenge
se_init_back_init	A4	V	V	Back to INIT state
se_init_change_vmk	A5	V	V	Change VMK
Category: Host Binding				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_bind_reg_init	D0	V	V	Init binding registeration
se_bind_reg_chlng	D1	V	V	Get registration challenge
se_bind_reg_finish	D2	V	V	Finish binding registration
se_bind_reg_info	D3	V	V	Get registered host info
se_bind_reg_approve	D4	V	V	Approve registered host
se_bind_reg_remove	D5	V	V	Remove registered host
se_bind_login_chlng	D6	V	V	Binding login challenge
se_bind_login	D7	V	V	Host binding login
se_bind_logout	D8	V	V	Host binding logout
se_bind_find_hstid	D9	V	V	Find host ID by credential
se_bind_back_nohost	DA	V	V	Back to no host mode
Category: Perso				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_perso_set_data	30	V	V	Set perso data

se_perso_get_data_hash	31	V	V	Get perso data hash
se_perso_confirm	32	V	V	Confirm perso data
se_perso_back_perso	33	V	V	Back to PERSO state
Category: Auth				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_pin_chlng	20	V	V	Get PIN auth challenge
se_pin_auth	21	V	V	PIN authentication
se_pin_change	22	V	V	Change PIN
se_pin_logout	23	V	V	PIN logout
se_puk_chlng	24	V	V	Get PUK challenge
se_pin_unlock	25	V	V	Unlock PIN by PUK
Category: BCDC Setting				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_set_currency	40	V	V	Set currency setting
se_get_currency	41	V	V	Get currency setting
se_get_card_name	42	V	V	Get SE card name
se_set_card_name	43	V	V	Set SE card name
se_get_secpo	44	V	V	Get security policy setting
se_set_secpo	45	V	V	Set security policy setting
se_get_card_id	46	V	V	Get SE card ID
Category: Trx Signing				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_trx_status	80	V	V	Get transaction signing status
se_trx_prepare	71	V	V	Prepare Trx signing
se_trx_begin	72	V	V	Transaction signing begins
se_trx_verify_otp	73	V	V	Verify OTP
se_trx_sign	74	V	V	Sign transaction
se_trx_get_ctxinfo	75	V	V	Get transaction signing context info
se_trx_finish	76	V	V	Finish transaction signing
se_trx_outaddr	79	V		Get trx signing output address
Category: HDW				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_hdw_init_wallet	B0	V	V	Initialize HDW
se_hdw_init_wallet_gen	B1	V	V	Initialize HDW (gen key)
se_hdw_qry_wa_info	B2	V	V	Query HDW info

se_hdw_set_wa_info	B3	V	V	Set HDW info
se_hdw_create_account	B4	V	V	Create HDW account
se_hdw_qry_acc_info	B5	V	V	Query HDW account info
se_hdw_set_acc_info	B6	V	V	Set HDW account info
se_hdw_next_trx_addr	B7	V	V	Get next trx address
se_hdw_prep_trx_sign	B8	V	V	Prepare HDW trx signing
se_hdw_init_wallet_gen_confirm	B9	V	V	Confirm HDW initialization (gen key)
se_hdw_qry_acc_keyinfo	BA	V	V	Query HDW account key info
Category: Mailbox				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
mbox_spi_get_msg	E0	V		Get mailbox message
mbox_spi_send_resp	E1	V		Send response to mailbox
mbox_iso_send_msg	E8		V	Send mailbox message
mbox_iso_get_resp	E9		V	Get response to mailbox
Category: Exchange Site				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_xchs_reg_status	F0	V	V	Get registration status
se_xchs_reg_init	F1	V	V	Registration init
se_xchs_reg_finish	F2	V	V	Registration finish
se_xchs_reg_clear	F3	V	V	Clear registration status
se_xchs_get_otp	F4	V	V	Get exchange site OTP
se_xchs_session_init	F5	V	V	Exchange site session init
se_xchs_session_estab	F6	V	V	Exchange site session establish
se_xchs_session_logout	F7	V	V	Logout established session
se_xchs_block_info	F8	V	V	Get blocking info
se_xchs_block_btc	F9	V	V	Block account Bitcoin
se_xchs_cancel_block	FA	V	V	Cancel Bitcoin blocking
se_xchs_trxsign_login	FB	V	V	Exchange site trx signing login
se_xchs_trxsign_prepare	FC	V	V	Exchange site trx signing prepare
se_xchs_trxsign_logout	FD	V	V	Exchange site trx signing logout
Category: Credential				
Command	**INS**	**SPI**	**ISO**	**Descriptions**
se_cred_get_mem	38	V	V	Get memory credential
se_cred_set_mem	39	V	V	Restore memory credential
se_cred_get_nvm	3A	V	V	Get NVM credential

se_cred_set_nvm	3B	V	V	Restore NVM credential

3. CWSE APDU Command Error Codes

Error ID (Hex)	Error Code	Description
01	ERR_CMD_NOT_SUPPORT	Command not supported
02	ERR_MODE_ID	Wrong mode ID
03	ERR_LC	Wrong APDU LC
04	ERR_TEST_FUNC_ID	Wrong test function ID
05	ERR_BCDC_TRX_STATE	Wrong trx signing state
06	ERR_TRX_VERIFY_OTP	Trx OTP verification fail
07	ERR_WALLET_INACTIVE	Wallet is not active
08	ERR_WALLET_ACTIVE	Wallet is active
09	ERR_WALLET_MISMATCH	Wallet id mismatch
0A	ERR_WRONG_OUTID	Wrong output id for trx_sign
0B	ERR_CDATA_TIMEOUT	Waiting for cdata timeout (7816 interface only)
0C	ERR_NO_RESP	No response data (7816 interface only)
0D	ERR_HASH_CHECK	Fail to pass hash check
0E	ERR_WAADDR_CHECK	Fail to pass wallet address check
0F	ERR_BCDC_INITSTATE	Wrong BCDC init state
10	ERR_BCDC_IDATAINFO	Wrong input init data information
11	ERR_BCDC_IDATASTATE	Wrong init data state
12	ERR_BCDC_PERSOSTATE	Wrong BCDC perso state
13	ERR_BCDC_PDATAINFO	Wrong input perso data information
14	ERR_BCDC_PDATASTATE	Wrong perso data state
15	ERR_DRNG_GEN_RAND	DRNG module failed to generate random bytes
16	ERR_BCDC_TRX_INID	Wrong input ID
17	ERR_NO_CHLNG	No auth challenge generated
18	ERR_LOCK	Auth locked
19	ERR_AUTHFAIL	Auth fail, not locked yet
1A	ERR_AUTHLOCK	Auth fail and locked
1B	ERR_NO_AUTH	Not authed yet
1C	ERR_NO_LOCK	Not locked

1D	ERR_TEST_SUBFUNC_ID	Wrong test sub-function ID
1E	ERR_NO_CARDNAME	No card name exists
1F	ERR_WALLET_ID	Wrong wallet ID
20	ERR_EWAINFO_ID	Wrong export wallet info ID
21	ERR_NO_CURRENCY	No currency data exists
22	ERR_TRX_INFOID	Wrong trx context INFO ID
23	ERR_WAPKG_ID	Wrong wallet package ID
24	ERR_INTER_MODULE	Internal module error
25	ERR_BAK_STATE	Wrong backup status
26	ERR_BAK_HANDLE	Wrong backup handle
27	ERR_WA_STATUS	Wrong wallet status
28	ERR_RES_STATE	Wrong restore status
29	ERR_RES_CHKSUM	Wrong restore checksum
2A	ERR_RES_HANDLE	Wrong restore handle
2B	ERR_RES_RSID	Wrong restore seed ID
2C	ERR_BIND_HSTID	Wrong binding host ID
2D	ERR_BIND_NOLOGIN	Not in binding login state
2E	ERR_BIND_HSTSTAT	Wrong host binding status
2F	ERR_BIND_LOGINSTAT	Wrong host login status
30	ERR_BIND_LOGIN	Binding login fail
31	ERR_HDW_STATUS	Wrong HD wallet status
32	ERR_HDW_NULEN	Wrong number set length
33	ERR_HDW_INFOID	Wrong HDW info ID
34	ERR_HDW_INFOLEN	Wrong HDW info length
35	ERR_HDW_ACCID	Wrong HDW account ID
36	ERR_HDW_ACCINFOID	Wrong HDW account info ID
37	ERR_HDW_KCID	Wrong key chain ID
38	ERR_HDW_KEYID	Wrong key ID
39	ERR_HDW_ACCINFOLEN	Wrong account info length
40	ERR_HDW_ACTVCODE	Wrong activation code
41	ERR_HDW_ACCPTR	Wrong account pointer value
42	ERR_HDW_OUTOFKEY	Out of keys
43	ERR_BIND_ALRDYNOHOST	Already no host
44	ERR_BIND_FIRST	Wrong first flag
45	ERR_BIND_HOSTFULL	Full of hosts

46	ERR_BIND_REGSTAT	Wrong host registration status
47	ERR_BIND_BRHANDLE	Wrong brhandle
48	ERR_BIND_REGRESP	Wrong registration response
49	ERR_LDR_AUTH	Back IKV loader auth fail
4A	ERR_LDR_BACK	Fail to back to IKV loader
4B	ERR_XCHS_REGST	Wrong XCHS registration status
4C	ERR_XCHS_SESSST	Wrong session status
4D	ERR_XCHS_SVRRESP	Wrong server response
4E	ERR_XCHS_OKTKN	Wrong OK token
4F	ERR_XCHS_MAC	Wrong XCHS MAC value
50	ERR_XCHS_BIFULL	XCHS block info full
51	ERR_XCHS_BLKAMNT	Wrong XCHS block amount
52	ERR_CRED_HANDLE	Wrong credential handle
53	ERR_HDW_NUCHKSUM	Wrong number set checksum
54	ERR_MAC	Wrong MAC value
55	ERR_INIT_PRID	Wrong pre-reg host ID
56	ERR_NVM_READ	Fail to perform NVM read
57	ERR_NVM_WRITE	Fail to perform NVM write
58	ERR_TRX_AMOUNT	Wrong trx amount
59	ERR_TRX_SIGTYPE	Wrong trx signature type

4. CWSE APDU Command Specification

4.1.SE Information

4.1.1. se_get_mode_state

Command	se_get_mode_state					
Description	Get SE mode and state					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 10 00 00	**LC**	00	**LE**	02
	CDATA	None				
	RESPONSE	[MODE] [STATE] [SW (2 bytes)]				
Parameter	MODE	Current MODE ID (1 byte) 00h: INIT 01h: PERSO 02h: NORMAL 03h: AUTH 04h: LOCK 05h: ERROR 06h: NOHOST 07h: DISCONN				
	STATE	Execution reuslt of last command (1 byte)				
Note	None					

4.1.2. se_get_fw_version

Command	se_get_fw_version					
Description	Get SE firmware version					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 11 00 00	**LC**	00	**LE**	10
	CDATA	None				
	RESPONSE	[VERINFO] [SW (2 bytes)]				
Parameter	VERINFO	Version info (16 bytes)				
Note	None					

4.1.3. se_get_unique_id

Command	se_get_unique_id					
Description	Get SE unique ID					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 12 00 00	**LC**	00	**LE**	08
	CDATA	None				
	RESPONSE	[UID] [SW (2 bytes)]				
Parameter	UID	Unique ID (8 bytes)				
Note	None					

4.1.4. se_get_mod_err

Command	se_get_mod_err					
Description	Get internal module error					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 13 00 00	**LC**	00	**LE**	02
	CDATA	None				
	RESPONSE	[MODID] [MODERR] [SW (2 bytes)]				
Parameter	MODID	Module ID (2 bytes, little-endian)				
	MODERR	Module error (6 bytes, little-endian)				
Note	None					

4.1.5. se_get_basic_info

Command	se_get_basic_info					
Description	Get SE basic info					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 14 00 00	**LC**	00	**LE**	22
	CDATA	None				
	RESPONSE	[MODE] [STATE] [VERINFO] [UID] [CARDID] [SW (2 bytes)]				
Parameter	MODE	Current MODE ID (1 byte) 00h: INIT 01h: PERSO 02h: NORMAL 03h: AUTH 04h: LOCK 05h: ERROR 06h: NOHOST 07h: DISCONN				
	STATE	Execution reuslt of last command (1 byte)				
	VERINFO	Version info (16 bytes)				
	UID	Unique ID (8 bytes)				
	CARDID	Card ID (8 bytes)				
Note	Card ID field is 00h’s if SE is not initialized yet					

4.1.6. se_back_ikvldr

Command	se_back_ikvldr					
Description	Back to IKV loader					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 78 00 00	**LC**	10	**LE**	00
	CDATA	[BLOTP]				
	RESPONSE	[SW (2 bytes)]				
Parameter	BLOTP	Back loader OTP (16 bytes)				
Note	None					

4.2.Initialization

4.2.1. se_init_set_data

Command	se_init_set_data						
Description	Set init data						
Supported Interfaces	ISO, SPI						
Supported Modes	00h: INIT						
Command Specification	**APDU**	80 A0 [IDID] [PRID]	**LC**	var.	**LE**	00	
	CDATA	[INITDATA] [IDHASH]					
	RESPONSE	[SW (2 bytes)]					
Parameter	IDID	Init data ID 0: Default User PIN hash (32 bytes) 1: PUK (32 bytes) 2: SEMK (32 bytes) 3: Card ID (8 bytes) 4: OTPK (32 bytes) 5: SMK (32 bytes) 6: Pre-reg host description (64 bytes) 7: Pre-reg host OTP key (32 bytes)					
	PRID	Pre-reg host ID (0 ~ 6) Only used for IDID 6 and 7					
	INITDATA	Init data (variable length)					
	IDHASH	SHA256 value of INITDATA (32 bytes)					
Note	None						

4.2.2. se_init_get_data_hash

Command	se_init_get_data_hash					
Description	Get init data hash					
Supported Interfaces	ISO, SPI					
Supported Modes	00h: INIT					
Command Specification	**APDU**	80 A1 [IDID] [PRID]	**LC**	00	**LE**	20
	CDATA	None				
	RESPONSE	[IDHASH] [SW (2 bytes)]				
Parameter	IDID	Init data ID 0: Default User PIN hash (32 bytes) 1: PUK (32 bytes) 2: SEMK (32 bytes) 3: Card ID (8 bytes) 4: OTPK (32 bytes) 5: SMK (32 bytes) 6: Pre-reg host description (64 bytes) 7: Pre-reg host OTP key (32 bytes)				
	PRID	Pre-reg host ID (0 ~ 6) Only used for IDID 6 and 7				
	IDHASH	SHA256 value of Init data (32 bytes)				
Note	None					

4.2.3. se_init_confirm

Command	se_init_confirm					
Description	Confirm init data					
Supported Interfaces	ISO, SPI					
Supported Modes	00h: INIT					
Command Specification	**APDU**	80 A2 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	None [SW (2 bytes)]				
Parameter	None					
Note	1. All init data should be set beform calling this command 2. SE will transist to PERSO state after init data confirmed					

4.2.4. se_init_vmk_chlng

Command	se_init_vmk_chlng					
Description	Get VMK challenge					
Supported Interfaces	ISO, SPI					
Supported Modes	00h: INIT 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 A3 00 00	**LC**	00	**LE**	10
	CDATA	None				
	RESPONSE	[CHLNG] [SW (2 bytes)]				
Parameter	CHLNG	Auth challenge (16 bytes)				
Note	None					

4.2.5. se_init_back_init

Command	se_init_back_init					
Description	Back to INIT state					
Supported Interfaces	ISO, SPI					
Supported Modes	06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 A4 00 00	**LC**	10	**LE**	00
	CDATA	[VMKRESP]				
	RESPONSE	[SW (2 bytes)]				
Parameter	VMKRESP	VMK response (16 bytes)				
Note	1. VMKRESP is calculated by AES_ENC_{VMK}(CHLNG) 2. CHLNG (challenge) is got from se_init_vmk_chlng command 3. VMK authentication should be passed to back INIT state successfully					

4.2.6. se_init_change_vmk

Command	se_init_change_vmk					
Description	Change VMK					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: INIT					
Command Specification	**APDU**	80 A5 00 00	**LC**	50	**LE**	00
	CDATA	[VMKRESP] [WRPVMK] [HASH]				
	RESPONSE	[SW (2 bytes)]				
Parameter	VMKRESP	VMK response (16 bytes)				
	WRPVMK	Wrapped VMK (32 bytes)				
	HASH	SHA256 of WRPVMK (32 bytes)				
Note	● VMKRESP is calculated by $AES\_ENC_{VMK}(CHLNG)$ ● CHLNG (challenge) is got from se_init_vmk_chlng command ● New VMK is wrapped by “old VMK” ● Encryption algorithm is AES-256 ECB mode					

4.3.Host Binding

4.3.1. se_bind_reg_init

Command	se_bind_reg_init					
Description	Init binding registeration					
Supported Interfaces	ISO, SPI					
Supported Modes	06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 D0 [FIRST] 00	**LC**	80	**LE**	0A
	CDATA	[HSTCRED] [HSTDESC] [HASH]				
	RESPONSE	[BRHANDLE] [OTP] [SW (2 bytes)]				
Parameter	FIRST	First registered host flag (1 byte)				
	HSTCRED	Host credential (32 bytes)				
	HSTDESC	Host description (64 bytes)				
	HASH	HASH of HSTCRED || HSTDESC (32 bytes)				
	BRHANDLE	Binding registration handle (4 bytes)				
	OTP	Composition of OTP key (6 bytes, ASCII format)				
Note	FIRST is for indicating this is the first registered host of BCSE. It should be set in the NOHOST mode and cleared in the DISCONN mode.					

4.3.2. se_bind_reg_chlng

Command	se_bind_reg_chlng					
Description	Get registration challenge					
Supported Interfaces	ISO, SPI					
Supported Modes	06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 D1 00 00	**LC**	04	**LE**	10
	CDATA	[BRHANDLE]				
	RESPONSE	[REGCHLNG] [SW (2 bytes)]				
Parameter	BRHANDLE	Binding registration handle (4 bytes)				
	REGCHLNG	Registration challenge (16 bytes)				
Note	None					

4.3.3. se_bind_reg_finish

Command	se_bind_reg_finish					
Description	Finish binding registration					
Supported Interfaces	ISO, SPI					
Supported Modes	06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 D2 00 00	**LC**	24	**LE**	02
	CDATA	[BRHANDLE] [REGRESP] [PINRESP]				
	RESPONSE	[HST_ID] [CONFIRM] [SW (2 bytes)]				
Parameter	BRHANDLE	Binding registration handle (4 bytes)				
	REGRESP	Registration response (16 bytes)				
	PINRESP	PIN response (16 bytes) *Irrelevant for the "add host" case*				
	HST_ID	Host ID (1 byte)				
	CONFIRM	Confirmation status (1 byte) 00h: Confirmed 01h: Not confirmed				
Note	● PINCHLNG should be got from se_pin_chlng command ● PINRESP is only necessary for "register 1st host" case (FIRST flag is set in the se_bind_reg_init command) ● REGRESP = AES256(DevKey, REGCHLNG), where DevKey = SHA256(HSTCRED || OTP)					

4.3.4. se_bind_reg_info

Command	se_bind_reg_info					
Description	Get registered host info					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO 02h: NORMAL 03h: AUTH 04h: LOCK 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 D3 [HST_ID] 00	**LC**	00	**LE**	81
	CDATA	None				
	RESPONSE	[BINDSTATE] [HSTDESC] [SW (2 bytes)]				
Parameter	HST_ID	Host ID (1 byte)				
	BINDSTATE	Binding status (1 byte) 00h: Empty 01h: Registered 02h: Confirmed				
	HSTDESC	Host description (64 bytes)				
Note	None					

4.3.5. se_bind_reg_approve

Command	se_bind_reg_approve					
Description	Approve unconfirmed registered host					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO 02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 D4 [HST_ID] 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	[SW (2 bytes)]				
Parameter	HST_ID	Host ID (1 byte)				
Note	None					

4.3.6. se_bind_reg_remove

Command	se_bind_reg_remove					
Description	Remove registered host					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO 02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 D5 [HST_ID] 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	[SW (2 bytes)]				
Parameter	HST_ID	Host ID (1 byte)				
Note	None					

4.3.7. se_bind_login_chlng

Command	se_bind_login_chlng					
Description	Get login challenge					
Supported Interfaces	ISO, SPI					
Supported Modes	07h: DISCONN					
Command Specification	**APDU**	80 D6 [HST_ID] 00	**LC**	00	**LE**	10
	CDATA	None				
	RESPONSE	[BINDCHLNG] [SW (2 bytes)]				
Parameter	HST_ID	Host ID (1 byte)				
	BINDCHLNG	Binding login challenge (16 bytes)				
Note	None					

4.3.8. se_bind_login

<table>
<tr><td>Command</td><td colspan="6">se_bind_login</td></tr>
<tr><td>Description</td><td colspan="6">Host binding login</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 D7 [HST_ID] 00</td><td>LC</td><td>10</td><td>LE</td><td>00</td></tr>
<tr><td>CDATA</td><td colspan="5">[BINDRESP]</td></tr>
<tr><td>RESPONSE</td><td colspan="5">[SW (2 bytes)]</td></tr>
<tr><td rowspan="2">Parameter</td><td>HST_ID</td><td colspan="5">Host ID (1 byte)</td></tr>
<tr><td>BINDRESP</td><td colspan="5">Binding login response (16 bytes)</td></tr>
<tr><td>Note</td><td colspan="6">● After login, the binding session (2 keys) is established. The established session keys are denoted by BIND_SENCK (Encryption key) and BIND_SMACK (MAC key)
● BIND_SENCK = SHA256 (BINDCHLNG || OTPKEY || BINDRESP || “ENC”)
● BIND_SMACK = SHA256 (BINDCHLNG || OTPKEY || BINDRESP || “MAC”)</td></tr>
</table>

4.3.9. se_bind_logout

Command	se_bind_logout					
Description	Host binding logout					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO 02h: NORMAL 04h: LOCK					
Command Specification	**APDU**	80 D8 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	[SW (2 bytes)]				
Parameter	None					
Note	None					

4.3.10. se_bind_find_hstid

<table>
<tr><td>Command</td><td colspan="6">se_bind_find_hst_id</td></tr>
<tr><td>Description</td><td colspan="6">Find registered host ID by host credential</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">01h: PERSO
02h: NORMAL
03h: AUTH
04h: LOCK
06h: NOHOST
07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 D9 00 00</td><td>LC</td><td>20</td><td>LE</td><td>02</td></tr>
<tr><td>CDATA</td><td colspan="5">[HSTCRED]</td></tr>
<tr><td>RESPONSE</td><td colspan="5">[HST_ID] [CONFIRM] [SW (2 bytes)]</td></tr>
<tr><td rowspan="3">Parameter</td><td>HSTCRED</td><td colspan="5">Host credential (32 bytes)</td></tr>
<tr><td>HST_ID</td><td colspan="5">Host ID (1 byte)
HST_ID is FFh means no matched host</td></tr>
<tr><td>CONFIRM</td><td colspan="5">Confirmation status (1 byte)
00h: Confirmed
01h: Not confirmed</td></tr>
<tr><td>Note</td><td colspan="6">None</td></tr>
</table>

4.3.11. se_bind_back_nohost

<table>
<tr><td>Command</td><td colspan="6">se_bind_back_nohost</td></tr>
<tr><td>Description</td><td colspan="6">Back to NOHOST mode</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">01h: PERSO
07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 DA 00 00</td><td>LC</td><td>30</td><td>LE</td><td>00</td></tr>
<tr><td>CDATA</td><td colspan="4">[PINRESP] [PINHASH]</td></tr>
<tr><td>RESPONSE</td><td colspan="4">[SW (2 bytes)]</td></tr>
<tr><td rowspan="2">Parameter</td><td>PINRESP</td><td colspan="4">PIN response (16 bytes)
For DISCONN mode only</td></tr>
<tr><td>PINHASH</td><td colspan="4">New PIN hash (32 bytes)
For PERSO mode only</td></tr>
<tr><td>Note</td><td colspan="6">● All wallet data and perso data will be cleared
● PINCHLNG should be got from se_pin_chlng command
● PINRESP is only necessary for “backing from DISCONN mode” case. If backing from PERSO mode, just set PINRESP as zero.
● PINHASH is encrypted by BIND_SENCK if backing from PERSO mode; it’s not necessary if backing from DISCONN mode, just set PINHASH as zero.</td></tr>
</table>

4.4.Personalization

4.4.1. se_perso_set_data

Command	se_perso_set_data					
Description	Set perso data					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO					
Command Specification	**APDU**	80 30 [PDID] 00	**LC**	var.	**LE**	00
	CDATA	[PERDATA] [PDMAC]				
	RESPONSE	[SW (2 bytes)]				
Parameter	PDID	Perso data ID 0: Security policy (4 bytes)				
	PERDATA	Perso data (variable length)				
	PDMAC	MAC of PERDATA (32 bytes) MAC key is BIND_SMACK				
Note	None					

4.4.2. se_perso_get_data_hash

Command	se_perso_get_data_hash					
Description	Get perso data hash					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO					
Command Specification	**APDU**	80 31 [PDID] 00	**LC**	00	**LE**	20
	CDATA	None				
	RESPONSE	[PDHASH] [SW (2 bytes)]				
Parameter	PDID	Perso data ID 0: Security policy (4 bytes)				
	PDHASH	SHA256 value of Perso data (32 bytes)				
Note	None					

4.4.3. se_perso_confirm

Command	se_perso_confirm					
Description	Confirm perso data					
Supported Interfaces	ISO, SPI					
Supported Modes	01h: PERSO					
Command Specification	**APDU**	80 32 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	None [SW (2 bytes)]				
Parameter	None					
Note	1. All perso data should be set before calling se_perso_confirm 2. SE state will transist to NORMAL after perso data confirmed					

4.4.4. se_perso_back_perso

Command	se_perso_back_perso					
Description	Back to PERSO state					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 04h: LOCK					
Command Specification	**APDU**	80 33 00 00	**LC**	20	**LE**	00
	CDATA	[PINHASH]				
	RESPONSE	[SW (2 bytes)]				
Parameter	PINHASH	New PIN hash (32 bytes) PINHASH is encrypted by BIND_SENCK				
Note	None					

4.5.Authentication

4.5.1. se_pin_chlng

Command	se_pin_chlng					
Description	Get PIN auth challenge					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 20 00 00	**LC**	00	**LE**	10
	CDATA	None				
	RESPONSE	[PINCHLNG] [SW (2 bytes)]				
Parameter	PINCHLNG	PIN challenge (16 bytes)				
Note	PINCHLNG can be used for PIN authentication (se_pin_auth), first host registration (se_bind_reg_finish) and backing to NOHOST mode from DISCONN mode (se_bind_back_nohost)					

4.5.2. se_pin_auth

Command	se_pin_auth					
Description	PIN authentication					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL					
Command Specification	**APDU**	80 21 00 00	**LC**	10	**LE**	00
	CDATA	[PINRESP]				
	RESPONSE	[SW (2 bytes)]				
Parameter	PINRESP	PIN response (16 bytes)				
Note	● This command can only be executed successfully when PIN function is enabled in security policy ● If auth passed, SE will enter AUTH mode ● RESP is calculated by $AES\_ENC_{PINHASH}(CHLNG)$					

4.5.3. se_pin_change

Command	se_pin_change					
Description	Change PIN					
Supported Interfaces	ISO, SPI					
Supported Modes	03h: AUTH					
Command Specification	**APDU**	80 22 00 00	**LC**	40	**LE**	00
	CDATA	[WRPINHASH] [MAC]				
	RESPONSE	[SW (2 bytes)]				
Parameter	WRPINHASH	Wrapped new PIN hash (32 bytes)				
	MAC	MAC of WRPINHASH (32 bytes) MAC key is BIND_SMACK				
Note	● New PIN hash is wrapped by “old PIN hash” ● Encryption algorithm is AES-256 ECB mode					

4.5.4. se_pin_logout

Command	se_pin_logout					
Description	PIN logout					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 23 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	[SW (2 bytes)]				
Parameter	None					
Note	● This command can only be executed successfully when PIN function is enabled in security policy ● After this command SE will enter UNAUTH mode					

4.5.5. se_puk_chlng

Command	se_puk_chlng					
Description	Get PUK verify challenge					
Supported Interfaces	ISO, SPI					
Supported Modes	04h: LOCK					
Command Specification	**APDU**	80 24 00 00	**LC**	00	**LE**	10
	CDATA	None				
	RESPONSE	[CHLNG] [SW (2 bytes)]				
Parameter	CHLNG	PUK auth challenge (16 bytes)				
Note	This command can only be executed successfully when PIN and LOCK functions are enabled in security policy					

4.5.6. se_pin_unlock

Command	se_pin_unlock					
Description	Unlock PIN by PUK					
Supported Interfaces	ISO, SPI					
Supported Modes	04h: LOCK					
Command Specification	**APDU**	80 25 00 00	**LC**	30	**LE**	00
	CDATA	[RESP] [PINHASH]				
	RESPONSE	[SW (2 bytes)]				
Parameter	RESP	Verification response (16 bytes)				
	PINHASH	New PIN hash (32 bytes) PINHASH is encrypted by BIND_SENCK				
Note	● This command can only be executed successfully when PIN and LOCK functions are enabled in security policy ● RESP is calculated by $AES\_ENC_{PUK}(CHLNG)$ ● CHLNG (challenge) is got from se_puk_chlng command					

4.6.BCDC Setting

4.6.1. se_set_currency

<table>
<tr><td>Command</td><td colspan="6">se_set_currency</td></tr>
<tr><td>Description</td><td colspan="6">Set currency setting</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">02h: NORMAL
03h: AUTH
06h: NOHOST
07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 40 00 00</td><td>LC</td><td>05</td><td>LE</td><td>00</td></tr>
<tr><td>CDATA</td><td colspan="5">[CURNCY]
● BYTE[0]: CurrType
● BYTE[1 ~ 4]: CurrRate, big-endian</td></tr>
<tr><td>RESPONSE</td><td colspan="5">[SW (2 bytes)]</td></tr>
<tr><td>Parameter</td><td>CURNCY</td><td colspan="5">Currency setting (5 bytes)</td></tr>
<tr><td>Note</td><td colspan="6">This command can be executed successfully only if SE is persoed</td></tr>
</table>

4.6.2. se_get_currency

<table>
<tr><td>Command</td><td colspan="6">se_get_currency</td></tr>
<tr><td>Description</td><td colspan="6">Get currency setting</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">02h: NORMAL
03h: AUTH
06h: NOHOST
07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 41 00 00</td><td>LC</td><td>00</td><td>LE</td><td>05</td></tr>
<tr><td>CDATA</td><td colspan="5">None</td></tr>
<tr><td>RESPONSE</td><td colspan="5">[CURNCY] [SW (2 bytes)]</td></tr>
<tr><td>Parameter</td><td>CURNCY</td><td colspan="5">Currency setting (5 bytes)
● BYTE[0]: CurrType
● BYTE[1 ~ 4]: CurrRate, big-endian</td></tr>
<tr><td>Note</td><td colspan="6">This command can be executed successfully only if SE is persoed</td></tr>
</table>

4.6.3. se_get_card_name

Command	se_get_card_name					
Description	Get SE card name					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 42 00 00	**LC**	00	**LE**	20
	CDATA	None				
	RESPONSE	[CARDNAME] [SW (2 bytes)]				
Parameter	CARDNAME	Card name (32 bytes)				
Note	This command can be executed successfully only if SE is persoed					

4.6.4. se_set_card_name

Command	se_set_card_name					
Description	Set SE card name					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 43 00 00	**LC**	20	**LE**	00
	CDATA	[CARDNAME]				
	RESPONSE	[SW (2 bytes)]				
Parameter	CARDNAME	Card name (32 bytes)				
Note	This command can be executed successfully only if SE is persoed					

4.6.5. se_get_secpo

<table>
<tr><td>Command</td><td colspan="6">se_get_secpo</td></tr>
<tr><td>Description</td><td colspan="6">Get security policy setting</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">02h: NORMAL
03h: AUTH
06h: NOHOST
07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 44 00 00</td><td>LC</td><td>00</td><td>LE</td><td>04</td></tr>
<tr><td>CDATA</td><td colspan="5">None</td></tr>
<tr><td>RESPONSE</td><td colspan="5">[SECPO] [SW (2 bytes)]</td></tr>
<tr><td>Parameter</td><td>SECPO</td><td colspan="5">Security policy setting (4 bytes)
● BYTE[0]:
■ BIT[0]: Trx signing OTP verification
■ BIT[1]: Trx signing button confirm
■ BIT[2]: PIN verification
■ BIT[3]: PIN LOCK mechanism
■ BIT[4]: WatchDog enable
■ BIT[5]: Display Address
■ BIT[6 ~ 7]: rfu
● BYTE[1 ~ 3]: rfu</td></tr>
<tr><td>Note</td><td colspan="6">None</td></tr>
</table>

4.6.6. se_set_secpo

<table>
<tr><td>Command</td><td colspan="6">se_set_secpo</td></tr>
<tr><td>Description</td><td colspan="6">Set security policy setting</td></tr>
<tr><td>Supported Interfaces</td><td colspan="6">ISO, SPI</td></tr>
<tr><td>Supported Modes</td><td colspan="6">02h: NORMAL
03h: AUTH
06h: NOHOST
07h: DISCONN</td></tr>
<tr><td rowspan="3">Command Specification</td><td>APDU</td><td>80 45 00 00</td><td>LC</td><td>04</td><td>LE</td><td>00</td></tr>
<tr><td>CDATA</td><td colspan="5">[SECPO]</td></tr>
<tr><td>RESPONSE</td><td colspan="5">[SW (2 bytes)]</td></tr>
<tr><td>Parameter</td><td>SECPO</td><td colspan="5">Security policy setting (4 bytes)
● BYTE[0]:
■ BIT[0]: Trx signing OTP verification
■ BIT[1]: Trx signing button confirm
■ BIT[2]: PIN verification
■ BIT[3]: PIN LOCK mechanism
■ BIT[4]: WatchDog enable
■ BIT[5]: Display Address
■ BIT[6 ~ 7]: rfu
● BYTE[1 ~ 3]: rfu</td></tr>
<tr><td>Note</td><td colspan="6">None</td></tr>
</table>

4.6.7. se_get_card_id

Command	se_get_card_id					
Description	Get SE card ID					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 46 00 00	**LC**	00	**LE**	08
	CDATA	None				
	RESPONSE	[CARDID] [SW (2 bytes)]				
Parameter	CARDID	Card ID (8 bytes)				
Note	None					

4.7. Transaction Signing

4.7.1. se_trx_status

Command	se_trx_status					
Description	Get transaction signing status					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 80 00 00	**LC**	00	**LE**	01
	CDATA	None				
	RESPONSE	[TRXSTAT] [SW (2 bytes)]				
Parameter	TRXSTAT	Transaction signing status (1 byte) 00h: Idle 01h: Transaction signing in preparing 02h: Transaction signing beginned 03h: Transaction signing OTP verified 04h: Transaction signing in progress				
Note	If PIN function is enabled, this command can only be executed successfully in AUTH mode					

4.7.2. se_trx_begin

Command	se_trx_begin					
Description	Transaction signing begins					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 72 00 00	**LC**	38	**LE**	06
	CDATA	[AMOUNT] [ENCOUTADDR]				
	RESPONSE	[OTP] [SW (2 bytes)]				
Parameter	AMOUNT	Transaction amount (8 bytes, big-endian)				
	ENCOUTADDR	Encrypted output address (48 bytes) The encryption key is BIND_SENCK				
	OTP	OTP digits (6 bytes, ASCII format)				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● If OTP function is disabled, the output OTP is all zero					

4.7.3. se_trx_verify_otp

Command	se_trx_verify_otp					
Description	Verify OTP					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 73 00 00	**LC**	06	**LE**	00
	CDATA	[OTP]				
	RESPONSE	[SW (2 bytes)]				
Parameter	OTP	OTP digits (6 bytes)				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● If OTP function is disabled, any OTP value can pass verification					

4.7.4. se_trx_sign

Command	se_trx_sign					
Description	Sign transaction					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 74 [IN_ID] 00	**LC**	00	**LE**	60
	CDATA	None				
	RESPONSE	[SIG] [SIGMAC][SW (2 bytes)]				
Parameter	IN_ID	Input ID (1 byte, 0 ~ 255)				
	SIG	Signature (64 bytes)				
	SIGMAC	MAC of signature (32 bytes) MAC Key is BIND_SMACK				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● Signature material is set in se_trx_prepare ● Input ID must be those set in se_trx_prepare					

4.7.5. se_trx_get_ctxinfo

Command	se_trx_get_ctxinfo					
Description	Get transaction signing context info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 75 [IN_ID] 00	**LC**	00	**LE**	2E
	CDATA	None				
	RESPONSE	[NUMIN] [SITYP] [WAID] [ACCID] [KCID] [KEYID] [SIGMTRL] [SW (2 bytes)]				
Parameter	IN_ID	Input ID (1 byte, 0 ~ 255)				
	NUMIN	Number of inputs (2 bytes, little-endian)				
	SITYP	Signing type (1 byte) 00h: HDW 01h: XCHS (Exchange site) 02h: Old-wallet				
	WAID	Signing wallet ID (2 bytes, little-endian) *Only effective in Old-wallet type*				
	ACCID	Signing account ID (4 bytes, little-endian) *Only effective in HDW or XCHS type*				
	KCID	Key chain ID (1 byte) *Only effective in HDW or XCHS type*				
	KEYID	Key ID (4 bytes, little-endian) *Only effective in HDW or XCHS type*				
	SIGMTRL	Signature material (32 bytes)				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode					

4.7.6. se_trx_finish

Command	se_trx_finish					
Description	Finish transaction signing					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 76 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	[SW (2 bytes)]				
Parameter	None					
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● Signing status is set to be IDLE ● All context info are cleared					

4.7.7. se_trx_outaddr

Command	se_trx_outaddr					
Description	Get trx signing output address					
Supported Interfaces	SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 79 00 00	**LC**	00	**LE**	22
	CDATA	None				
	RESPONSE	[OUTADDR] [SW (2 bytes)]				
Parameter	OUTADDR	Output address (34 bytes)				
Parameter	None					
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● Signing status must be in BEGIN or VERIFIED ● This output address is only used in HDW signing type, other type will just output zeros					

4.9.HD Wallet

4.9.1. se_hdw_init_wallet

Command	se_hdw_init_wallet					
Description	Initialize HDW					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B0 00 00	**LC**	80	**LE**	00
	CDATA	[HDWNAME] [EMKSEED] [MAC]				
	RESPONSE	[SW (2 bytes)]				
Parameter	HDWNAME	HD wallet name (32 bytes)				
	EMKSEED	Encrypted HDW master key seed (64 bytes)				
	MAC	MAC value of EMKSEED MAC key is BIND_SMACK				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● HDW must be in INACTIVE state ● EMKSEED is encrypted by BIND_SENCK					

4.9.2. se_hdw_init_wallet_gen

Command	se_hdw_init_wallet_gen					
Description	Initialize HDW (gen key)					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B1 00 00	**LC**	var.	**LE**	var.
	CDATA	[HDWNAME] [NULEN] [PALEN] [PASSPHR]				
	RESPONSE	[NUMSET] [ACTVCODE] [MAC] [SW (2 bytes)]				
Parameter	HDWNAME	HD wallet name (32 bytes)				
	NULEN	Number set length (1 byte, 24, 36 or 48)				
	PALEN	Pass phrase length (1 byte, 0 ~ 16)				
	PASSPHR	Pass phrase (var. length)				
	NUMSET	Number set (var. length, BCD format)				
	ACTVCODE	Activation code (4 bytes)				
	MAC	MAC of NUMSET || ACTVCODE MAC key is BIND_SMACK				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● HDW must be in INACTIVE state					

4.9.3. se_hdw_qry_wa_info

Command	se_hdw_qry_wa_info					
Description	Query HDW info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 B2 [INFOID] 00	**LC**	00	**LE**	var.
	CDATA	None				
	RESPONSE	[HDWINFO] [SW (2 bytes)]				
Parameter	INFOID	HDW info ID 00h: HDW status (1 byte) 01h: HDWname (32 bytes) 02h: HDW account pointer (4 bytes) 03h: All HDW info (37 bytes)				
	HDWINFO	HDW info (variable length)				
Note	● HDW status: ■ 00h: INACTIVE ■ 01h: WAITACTV ■ 02h: ACTIVE ● All HDW info: [HDW status] [HDWname] [HDW account pointer] (37 bytes)					

4.9.4. se_hdw_set_wa_info

Command	se_hdw_set_wa_info					
Description	Set HDW info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B3 [INFOID] 00	**LC**	var.	**LE**	00
	CDATA	[HDWINFO]				
	RESPONSE	[SW (2 bytes)]				
Parameter	INFOID	HDW info ID (32 bytes) 01h: HDWname (32 bytes) 02h: HDW account pointer (4 bytes)				
	HDWINFO	HDW info (variable length)				
Note	If PIN function is enabled, this command can only be executed successfully in AUTH mode					

4.9.5. se_hdw_create_account

Command	se_hdw_create_account					
Description	Create HDW account					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B4 00 00	**LC**	24	**LE**	00
	CDATA	[ACCID] [ACCNAME]				
	RESPONSE	[SW (2 bytes)]				
Parameter	ACCID	Account ID (4 bytes, little-endian)				
	ACCNAME	Account name (32 bytes)				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● Account ID must be equal to current account pointer ● Key pointers of the previous account cannot be 0					

4.9.6. se_hdw_qry_acc_info

Command	se_hdw_qry_acc_info					
Description	Query HDW account info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 B5 [INFOID] 00	**LC**	04	**LE**	var.
	CDATA	[ACCID]				
	RESPONSE	[ACCINFO] [SW (2 bytes)]				
Parameter	INFOID	Account info ID (1 byte) 00h: Account name (32 bytes) 01h: Balance (8 bytes, big-endian) 02h: External key pointer (4 bytes, little-endian) 03h: Internal key pointer (4 bytes, little-endian) 04h: Exchange site blocked balance (8 bytes, big-endian) 05h: All account info (56 bytes)				
	ACCID	Account ID (4 bytes, little-endian)				
	ACCINFO	Output account info (variable length)				
Note	All account info (56 bytes): [Account name] [Balance] [External key pointer] [Internal key pointer] [Ex-site blocked balance]					

4.9.7. se_hdw_set_acc _info

Command	se_hdw_set_acc_info					
Description	Set HDW account info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B6 [INFOID] 00	**LC**	var.	**LE**	00
	CDATA	[ACCID] [ACCINFO] [MAC]				
	RESPONSE	[SW (2 bytes)]				
Parameter	INFOID	Account info ID (1 byte) 00h: Account name (32 bytes) 01h: Balance (8 bytes) 02h: External key pointer (4 bytes) 03h: Internal key pointer (4 bytes)				
	ACCID	Account ID (4 bytes, little-endian)				
	ACCINFO	Output account info (variable length)				
	MAC	MAC value for ACCINFO (32 bytes) MAC key is BIND_SMACK				
Note	If PIN function is enabled, this command can only be executed successfully in AUTH mode					

4.9.8. se_hdw_next_trx_addr

Command	se_hdw_next_trx_addr					
Description	Get next trx address					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B7 [KCID] 00	**LC**	04	**LE**	3D
	CDATA	[ACCID]				
	RESPONSE	[KID] [ADDR] [MAC] [SW (2 bytes)]				
Parameter	KCID	Key Chain ID (1 byte) 00h: External chain 01h: Internal chain				
	ACCID	Account ID (4 bytes, little-endian)				
	KID	Key ID (4 bytes, little-endian)				
	ADDR	Output trx address (25 bytes)				
	MAC	MAC value for (KID || ADDR) (32 bytes) MAC key is BIND_SMACK				
Note	If PIN function is enabled, this command can only be executed successfully in AUTH mode					

4.9.9. se_hdw_prep_trx_sign

Command	se_hdw_prep_trx_sign					
Description	Prepare HDW trx signing					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B8 [IN_ID] [KCID]	**LC**	50	**LE**	00
	CDATA	[ACCID] [KID] [BALNC] [SIGMTRL] [MAC]				
	RESPONSE	[SW (2 bytes)]				
Parameter	IN_ID	Input ID (1 byte, 0 ~ 255)				
	KCID	Key Chain ID (1 byte) 00h: External chain 01h: Internal chain				
	ACCID	Account ID (4 bytes, little-endian)				
	KID	Key ID (4 bytes, little-endian)				
	BALNC	Transaction amount for this input (8 bytes)				
	SIGMTRL	Signature material (32 bytes)				
	MAC	MAC value of signature input data (ACCID || KID || BALNC || SIGMTRL) MAC key is BIND_SMACK				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● IN_ID has to start from 0, and must be input in serial ● Key (decided by ACCID and KID) cannot repeat ● BALANCE will be substracted from this account					

4.9.10. se_hdw_init_wallet_gen_confirm

Command	se_hdw_init_wallet_gen_confirm					
Description	Confirm HDW initialization (gen key)					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 B9 00 00	**LC**	0A	**LE**	00
	CDATA	[ACTVCODE] [NUCHKSUM]				
	RESPONSE	[SW (2 bytes)]				
Parameter	ACTVCODE	Activation code (4 bytes)				
	NUCHKSUM	Number set checksum (6 bytes, ASCII format)				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● ACTVCODE is got from se_hdw_init_wallet_gen ● HDW must be in INACTIVE state ● NUCHKSUM = Sum of NumSet ASCII String (6 digits)					

4.9.11. se_hdw_qry_acc_keyinfo

Command	se_hdw_qry_acc_keyinfo					
Description	Query HDW account key info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 BA [KINFOID] [KCID]	**LC**	08	**LE**	var.
	CDATA	[ACCID] [KID]				
	RESPONSE	[ACCINFO] [MAC] [SW (2 bytes)]				
Parameter	KINFOID	Account key info ID (1 byte) 00h: Address (25 bytes) 01h: Public key (64 bytes) 02h: Key chain public key and chain code (96 bytes)				
	ACCID	Account ID (4 bytes, little-endian)				
	KCID	Key Chain ID (1 byte) 00h: External chain 01h: Internal chain				
	KID	Key ID (4 bytes, little-endian)				
	ACCINFO	Output account info (variable length)				
	MAC	MAC value of ACCINFO (32 bytes) MAC key is BIND_SMACK				
Note	● For KINFOID 02h (Key chain public key), KID is irrelevant. ● Key chain public key and chain code includes: ■ Key chain public key (former 64 bytes) ■ Key chain chain code (latter 32 bytes)					

4.10. Mailbox

4.10.1. mbox_spi_get_msg

Command	mbox_spi_get_msg					
Description	Get mailbox message					
Supported Interfaces	SPI					
Supported Modes	All modes except ERROR					
Command Specification	**APDU**	80 E0 00 00	**LC**	00	**LE**	81
	CDATA	None				
	RESPONSE	[MSGLEN] [MSG] [SW (2 bytes)]				
Parameter	MSGLEN	Message length (2 bytes, little-endian)				
	MSG	Message (128 bytes)				
Note	● After mailbox message is got from mbox_spi_get_msg, the mailbox will be empty ● Message is of fixed length, effective length is indicated by MSGLEN					

4.10.2. mbox_spi_send_resp

Command	mbox_spi_send_resp					
Description	Send response to mailbox					
Supported Interfaces	SPI					
Supported Modes	All modes except ERROR					
Command Specification	**APDU**	80 E1 00 00	**LC**	var.	**LE**	00
	CDATA	None				
	RESPONSE	[RESP] [SW (2 bytes)]				
Parameter	RESP	Response (1 ~ 128 bytes)				
Note	The sent response will be stored in the buffer of SE, it will stay in the buffer until it is received by the other entity via ISO7816 interface, or replaced by newly sent response					

4.10.3. mbox_iso_send_msg

Command	mbox_iso_send_msg					
Description	Send mailbox message					
Supported Interfaces	ISO					
Supported Modes	All modes except ERROR					
Command Specification	**APDU**	80 E8 00 00	**LC**	var.	**LE**	00
	CDATA	[MSG]				
	RESPONSE	[SW (2 bytes)]				
Parameter	MSG	Message (1 ~ 128 bytes)				
Note	The message can be read by mbox_spi_get_msg via SPI interface					

4.10.4. mbox_iso_get_resp

Command	mbox_iso_get_resp					
Description	Get response to mailbox					
Supported Interfaces	ISO					
Supported Modes	All modes except ERROR					
Command Specification	**APDU**	80 E9 00 00	**LC**	00	**LE**	81
	CDATA	None				
	RESPONSE	[RESPLEN] [RESP] [SW (2 bytes)]				
Parameter	RESPLEN	Response length (2 bytes, little-endian)				
	RESP	Response (128 bytes)				
Note	Response is of fixed length 128 bytes, the effective length is indicated by RESPLEN					

4.11. Exchange Site

4.11.1. se_xchs_reg_status

Command	se_xchs_reg_status					
Description	Get registration status					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 F0 00 00	**LC**	00	**LE**	01
	CDATA	None				
	RESPONSE	[REGSTAT] [SW (2 bytes)]				
Parameter	REGSTAT	Exchange site registration status (1 byte) 00h: NOT_REGISTERED 01h: REG_INIT 02h: REGISTERED				
Note	None					

4.11.2. se_xchs_reg_init

Command	se_xchs_reg_init					
Description	Registration init					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F1 [QRY] 00	**LC**	20	**LE**	30
	CDATA	[USREMAIL]				
	RESPONSE	[NONCE] [MAC] [SW (2 bytes)]				
Parameter	QRY	Query flag (1 byte) *If query flag is cleared (zero), nonce will be generated and MAC value of user email will be calculated. The registration state must be NOT_REGISTERED.* *If query flag is set (nonzero value), the registration state must be REG_INIT. Nonce and MAC generated/calculated before will be output directly instead of generated again. User mail field will be ignored.*				
	USREMAIL	User email (32 bytes)				
	NONCE	Random material for MAC (16 bytes)				
	MAC	MAC value of (USREMAIL || NONCE) (32 bytes) *MAC key is XCHS_SEMK*				
Note	If PIN function is enabled, this command can only be executed successfully in AUTH mode					

4.11.3. se_xchs_reg_finish

Command	se_xchs_reg_finish					
Description	Registration finish					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F2 00 00	**LC**	70	**LE**	30
	CDATA	[EOTPK] [ESMK] [MAC] [NONCE]				
	RESPONSE	[REGTKN] [NONCESE] [SW (2 bytes)]				
Parameter	EOTPK	Exchange site login OTP key (Encrypted by XCHS_SEMK, 32 bytes)				
	ESMK	Exchange site session master key (Encrypted by XCHS_SEMK, 32 bytes)				
	MAC	MAC value of (EOTPK || ESMK) (32 bytes) *MAC key is XCHS_SEMK*				
	NONCE	Nonce for REGTOKEN (16 bytes)				
	REGTKN	Registration token (32 bytes)				
	NONCESE	Nonce for REGTOKEN generated by SE (16 bytes)				
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● REGTOKEN is MAC value of the following data (MAC key is XCHS_SEMK): ■ CARD_ID ■ UID ■ UserEmail ■ LOGIN_OTPKEY ■ SESSION_MK ■ nonce_Svr ■ nonce_SE					

4.11.4. se_xchs_reg_clear

Command	se_xchs_reg_clear					
Description	Clear registration status					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F3 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	None				
Parameter	None					
Note	● If PIN function is enabled, this command can only be executed successfully in AUTH mode ● The registration status must be REG_INIT or REGISTERED					

4.11.5. se_xchs_get_otp

Command	se_xchs_get_otp					
Description	Get exchange site OTP					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH 06h: NOHOST 07h: DISCONN					
Command Specification	**APDU**	80 F4 00 00	**LC**	00	**LE**	06
	CDATA	None				
	RESPONSE	[OTP] [SW (2 bytes)]				
Parameter	OTP	Exchange site login OTP digits (6 bytes) *ASCII format*				
Note	The registration status must be REGISTERED					

4.11.6. se_xchs_session_init

Command	se_xchs_session_init					
Description	Exchange site session init					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F5 00 00	**LC**	10	**LE**	20
	CDATA	[SVRCHLNG]				
	RESPONSE	[SERESP] [SECHLNG] [SW (2 bytes)]				
Parameter	SVRCHLNG	Server challenge (16 bytes)				
	SERESP	SE response (16 bytes)				
	SECHLNG	SE challenge (16 bytes)				
Note	The registration status must be REGISTERED					

4.11.7. se_xchs_session_estab

Command	se_xchs_session_estab					
Description	Exchange site session establish					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F6 00 00	**LC**	10	**LE**	00
	CDATA	[SVRRESP]				
	RESPONSE	[SW (2 bytes)]				
Parameter	SVRRESP	Server response (16 bytes)				
Note	● The registration status must be REGISTERED ● The established session key is called XCHS_SK, which is SHA256(SMK || SVR_CHLNG || SE_CHLNG)					

4.11.8. se_xchs_session_logout

Command	se_xchs_session_logout					
Description	Logout established session					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F7 00 00	**LC**	00	**LE**	00
	CDATA	None				
	RESPONSE	[SW (2 bytes)]				
Parameter	None					
Note	● The registration status must be REGISTERED ● XCHS session status will be reset to IDLE					

4.11.9. se_xchs_block_info

Command	se_xchs_block_info					
Description	Get blocking info					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F8 00 00	**LC**	04	**LE**	0D
	CDATA	[OKTKN]				
	RESPONSE	[STATE] [ACCID] [AMOUNT] [SW (2 bytes)]				
Parameter	OKTKN	OK token (4 bytes)				
	STATE	Block info status (1 byte)				
	ACCID	Account ID (4 bytes, little-endian)				
	AMOUNT	Amount (8 bytes, big-endian)				
Note	● The registration status must be REGISTERED ● XCHS session status must be established					

4.11.10. se_xchs_block_btc

Command	se_xchs_block_btc					
Description	Block account Bitcoin					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 F9 00 00	**LC**	40	**LE**	64
	CDATA	[TRXID] [ACCID] [AMOUNT] [MAC1] [NONCE]				
	RESPONSE	[BLKSIG] [OKTKN] [ENC_UBLKTKN] [MAC2] [NONCESE] [SW (2 bytes)]				
Parameter	TRXID	Transaction ID (4 bytes)				
	ACCID	Account ID (4 bytes, little-endian)				
	AMOUNT	Block amount (8 bytes, big-endian)				
	MAC1	MAC value of (TRXID || ACCID || AMOUNT) (32 bytes) *MAC key is XCHS_SK*				
	NONCE	Nonce for block signature (16 bytes)				
	BLKSIG	Block signature (32 bytes)				
	OKTKN	OK token (4 bytes)				
	ENC_UBLKTKN	Encrypted unblock token (16 bytes)				
	MAC2	MAC value of (BLKSIG || OKTKN || UBLKTKN) (32 bytes) *MAC key is XCHS_SK*				
	NONCESE	Nonce for block signature generated by SE (16 bytes)				
Note	● The registration status must be REGISTERED ● XCHS session must be established ● BLKSIG is MAC value of the following data (MAC key is XCHS_SMK): ■ CARD_ID ■ UID ■ TRX_ID					

	■ ACC ■ AMOUNT ■ nonce ■ nonce_SE ● Unblock token (16 bytes) is composed of: ■ BYTE [0 ~ 7]: Prefix ■ BYTE [8 ~ 15]: Amount

4.11.11. se_xchs_cancel_block

Command	se_xchs_cancel_block					
Description	Cancel Bitcoin blocking					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 FA 00 00	**LC**	48	**LE**	50
	CDATA	[TRXID] [OKTKN] [ENC_UBLKTKN] [MAC1] [NONCE]				
	RESPONSE	[UBLKSIG] [MAC2] [NONCESE] [SW (2 bytes)]				
Parameter	TRXID	Transaction ID (4 bytes)				
	OKTKN	OK token (4 bytes)				
	ENC_UBLKTKN	Encrypted unblock token (16 bytes)				
	MAC1	MAC value of (TRXID || OKTKN || UBLKTKN) (32 bytes) *MAC key is XCHS_SK*				
	NONCE	Nonce for unblock signature (16 bytes)				
	UBLKSIG	Unblock signature (32 bytes)				
	MAC2	MAC value of UBLKSIG (32 bytes) *MAC key is XCHS_SK*				
	NONCESE	Nonce for unblock signature generated by SE (16 bytes)				
Note	● The registration status must be REGISTERED ● XCHS session must be established ● UBLKSIG is MAC value of the following data (MAC key is XCHS_SMK): ■ CARD_ID ■ UID ■ TRX_ID ■ ACC ■ AMOUNT ■ nonce					

	■ nonce_SE ● Unblock token (16 bytes) is composed of: ■ BYTE [0 ~ 7]: Prefix ■ BYTE [8 ~ 15]: Amount

4.11.12. se_xchs_trxsign_login

Command	se_xchs_trxsign_login					
Description	Exchange site trx signing login					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 FB 00 00	**LC**	44	**LE**	04
	CDATA	[TRXID] [OKTKN] [ENC_UBLKTKN] [ACCID] [DEALAMNT] [MAC]				
	RESPONSE	[TRXHANDLE] [SW (2 bytes)]				
Parameter	TRXID	Transaction ID (4 bytes)				
	OKTKN	OK token (4 bytes)				
	ENC_UBLKTKN	Encrypted unblock token (16 bytes)				
	ACCID	Account ID (4 bytes, little-endian)				
	DEALAMNT	Deal amount (8 bytes, big-endian)				
	MAC	MAC value of (TRXID || OKTKN || UBLKTKN || ACCID || DEALAMNT) (32 bytes) *MAC key is XCHS_SK*				
	TRXHANDLE	Transaction signing handle (4 bytes)				
Note	● The registration status must be REGISTERED ● XCHS session must be established ● Unblock token (16 bytes) is composed of: ■ BYTE [0 ~ 7]: Prefix ■ BYTE [8 ~ 15]: Amount					

4.11.13. se_xchs_trxsign_prepare

Command	se_xchs_trxsign_prepare					
Description	Exchange site trx signing prepare					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 FC [IN_ID] 00	**LC**	7F	**LE**	00
	CDATA	[TRXHANDLE] [ACCID] [KCID] [KID] [OUT1ADDR] [OUT2ADDR] [SIGMTRL] [MAC]				
	RESPONSE	[SW (2 bytes)]				
Parameter	TRXHANDLE	Transaction signing handle (4 bytes)				
	IN_ID	Input ID (0 ~ 255)				
	ACCID	Account ID (4 bytes, little-endian)				
	KCID	Key chain ID (1 byte) 00h: External key chain 01h: Internal key chain				
	KID	Key ID (4 bytes, little-endian)				
	OUT1ADDR	Output 1 address (25 bytes)				
	OUT2ADDR	Output 2 address (25 bytes)				
	SIGMTRL	Signature material (32 bytes)				
	MAC	MAC value of (ACCID || KCID || KID || OUT1ADDR || OUT2ADDR || SIGMTRL) (32 bytes) *MAC key is XCHS_SK*				
Note	● The registration status must be REGISTERED ● XCHS session must be established ● XCHS signing session must be login ● IN_ID has to start from 0, and must be input in serial					

4.11.14. se_xchs_trxsign_logout

Command	se_xchs_trxsign_logout					
Description	Exchange site trx signing logout					
Supported Interfaces	ISO, SPI					
Supported Modes	02h: NORMAL 03h: AUTH					
Command Specification	**APDU**	80 FD 00 00	**LC**	14	**LE**	50
	CDATA	[TRXHANDLE] [NONCE]				
	RESPONSE	[SIGRCPT] [MAC] [NONCESE] [SW (2 bytes)]				
Parameter	TRXHANDLE	Transaction signing handle (4 bytes)				
	NONCE	Nonce for signature receipt (16 bytes)				
	SIGRCPT	Signature receipt (32 bytes)				
	MAC	MAC value of SIGRCPT (32 bytes) *MAC key is XCHS_SK*				
	NONCESE	Nonce for signature receipt generated by SE (16 bytes)				
Note	● The registration status must be REGISTERED ● XCHS session must be established ● SIGRCPT is MAC value of the following data (MAC key is XCHS_SMK): ■ CARD_ID ■ UID ■ TRX_ID ■ ACC_ID ■ DEAL_AMOUNT ■ NUM_INPUTS ■ OUT1ADDR ■ OUT2ADDR ■ nonce ■ nonce_SE					

4.12. Credential

4.12.1. se_cred_get_mem

Command	se_cred_get_mem					
Description	Get memory credential					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 38 00 00	**LC**	00	**LE**	25
	CDATA	None				
	RESPONSE	[HANDLE] [MEMCRED] [SW (2 bytes)]				
Parameter	HANDLE	Memory credential handle (4 bytes)				
	MEMCRED	Memory credential (33 bytes)				
Note	None					

4.12.2. se_cred_set_mem

Command	se_cred_set_mem					
Description	Restore memory credential					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 39 00 00	**LC**	25	**LE**	00
	CDATA	[HANDLE] [MEMCRED]				
	RESPONSE	[SW (2 bytes)]				
Parameter	HANDLE	Memory credential handle (4 bytes)				
	MEMCRED	Memory credential (33 bytes)				
Note	None					

4.12.3. se_cred_get_nvm

Command	se_cred_get_nvm					
Description	Get NVM credential					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 3A 00 00	**LC**	00	**LE**	04
	CDATA	None				
	RESPONSE	[HANDLE] [SW (2 bytes)]				
Parameter	HANDLE	NVM credential handle (4 bytes)				
Note	None					

4.12.4. se_cred_set_nvm

Command	se_cred_set_nvm					
Description	Restore NVM credential					
Supported Interfaces	ISO, SPI					
Supported Modes	All modes					
Command Specification	**APDU**	80 39 00 00	**LC**	04	**LE**	00
	CDATA	[HANDLE]				
	RESPONSE	[SW (2 bytes)]				
Parameter	HANDLE	NVM credential handle (4 bytes)				
Note	None					

5. Supported Commands in Each Mode

5.1. APDU in Each Mode – ISO Interface

The following table lists supported APDU commands for each mode in ISO interface.

Mode	Supported commands	
COMMON (All modes)	se_get_mode_state	mbox_iso_get_resp
	se_get_fw_version	se_cred_get_mem
	se_get_unique_id	se_cred_set_mem
	se_get_mod_err	se_cred_get_nvm
	mbox_iso_send_msg	se_cred_set_nvm
INIT	se_init_set_data	se_init_vmk_chlng
	se_init_get_data_hash	se_init_change_vmk
	se_init_confirm	
PERSO	se_bind_reg_info	se_bind_back_nohost
	se_bind_reg_approve	se_perso_set_data
	se_bind_reg_remove	se_perso_get_data_hash
	se_bind_logout	se_perso_confirm
NORMAL	se_bind_reg_info	se_hdw_set_wa_info
	se_bind_reg_approve	se_hdw_create_account
	se_bind_reg_remove	se_hdw_qry_acc_info
	se_bind_logout	se_hdw_set_acc_info
	se_perso_back_perso	se_hdw_next_trx_addr
	se_pin_chlng	se_hdw_prep_trx_sign
	se_pin_auth	se_hdw_init_wallet_gen_confirm
	se_pin_logout	se_hdw_qry_acc_keyinfo
	se_set_currency	se_set_secpo
	se_get_currency	se_xchs_reg_status
	se_get_card_name	se_xchs_reg_init
	se_set_card_name	se_xchs_reg_finish
	se_get_secpo	se_xchs_reg_clear
	se_trx_status	se_xchs_get_otp

	se_trx_prepare	se_xchs_session_init
	se_trx_begin	se_xchs_session_estab
	se_trx_verify_otp	se_xchs_session_logout
	se_trx_sign	se_xchs_block_info
	se_trx_get_ctxinfo	se_xchs_block_btc
	se_trx_finish	se_xchs_cancel_block
	se_hdw_init_wallet	se_xchs_trxsign_login
	se_hdw_init_wallet_gen	se_xchs_trxsign_prepare
	se_hdw_qry_wa_info	se_xchs_trxsign_logout
AUTH	se_bind_reg_info	se_hdw_create_account
	se_bind_reg_approve	se_hdw_qry_acc_info
	se_bind_reg_remove	se_hdw_set_acc_info
	se_pin_change	se_hdw_next_trx_addr
	se_pin_logout	se_hdw_prep_trx_sign
	se_set_currency	se_hdw_init_wallet_gen_confirm
	se_get_currency	se_hdw_qry_acc_keyinfo
	se_get_card_name	se_xchs_reg_status
	se_set_card_name	se_xchs_reg_init
	se_get_secpo	se_xchs_reg_finish
	se_set_secpo	se_xchs_reg_clear
	se_trx_status	se_xchs_get_otp
	se_trx_prepare	se_xchs_session_init
	se_trx_begin	se_xchs_session_estab
	se_trx_verify_otp	se_xchs_session_logout
	se_trx_sign	se_xchs_block_info
	se_trx_get_ctxinfo	se_xchs_block_btc
	se_trx_finish	se_xchs_cancel_block
	se_hdw_init_wallet	se_xchs_trxsign_login
	se_hdw_init_wallet_gen	se_xchs_trxsign_prepare
	se_hdw_qry_wa_info	se_xchs_trxsign_logout
	se_hdw_set_wa_info	
LOCK	se_bind_reg_info	se_puk_chlng
	se_bind_logout	se_pin_unlock
	se_perso_back_perso	
ERROR		

NOHOST	se_init_vmk_chlng	se_get_card_name
	se_init_back_init	se_set_card_name
	se_bind_reg_init	se_get_secpo
	se_bind_reg_chlng	se_hdw_qry_wa_info
	se_bind_reg_finish	se_hdw_qry_acc_info
	se_bind_reg_info	se_set_secpo
	se_pin_chlng	se_xchs_reg_status
	se_set_currency	se_xchs_get_otp
	se_get_currency	
DISCONN	se_init_vmk_chlng	se_pin_chlng
	se_init_back_init	se_set_currency
	se_bind_reg_init	se_get_currency
	se_bind_reg_chlng	se_get_card_name
	se_bind_reg_finish	se_set_card_name
	se_bind_reg_info	se_get_secpo
	se_bind_login_chlng	se_hdw_qry_wa_info
	se_bind_login	se_hdw_qry_acc_info
	se_bind_find_hst_id	se_xchs_reg_status
	se_set_secpo	se_xchs_get_otp
	se_bind_back_nohost	

5.2. APDU in Each Mode – SPI Interface

The following table lists supported APDU commands for each mode in SPI interface.

Mode	Supported commands	
COMMON (All modes)	se_get_mode_state	mbox_spi_send_resp
	se_get_fw_version	se_cred_get_mem
	se_get_unique_id	se_cred_set_mem
	se_get_mod_err	se_cred_get_nvm
	mbox_spi_get_msg	se_cred_set_nvm
INIT	se_init_set_data	se_init_vmk_chlng
	se_init_get_data_hash	se_init_change_vmk
	se_init_confirm	
PERSO	se_bind_reg_info	se_bind_back_nohost
	se_bind_reg_approve	se_perso_set_data
	se_bind_reg_remove	se_perso_get_data_hash
	se_bind_logout	se_perso_confirm
NORMAL	se_bind_reg_info	se_hdw_set_wa_info
	se_bind_reg_approve	se_hdw_create_account
	se_bind_reg_remove	se_hdw_qry_acc_info
	se_bind_logout	se_hdw_set_acc_info
	se_perso_back_perso	se_hdw_next_trx_addr
	se_pin_chlng	se_hdw_prep_trx_sign
	se_pin_auth	se_hdw_init_wallet_gen_confirm
	se_pin_logout	se_hdw_qry_acc_keyinfo
	se_set_currency	se_set_secpo
	se_get_currency	se_xchs_reg_status
	se_get_card_name	se_xchs_reg_init
	se_set_card_name	se_xchs_reg_finish
	se_get_secpo	se_xchs_reg_clear
	se_trx_status	se_xchs_get_otp
	se_trx_prepare	se_xchs_session_init
	se_trx_begin	se_xchs_session_estab

	se_trx_verify_otp	se_xchs_session_logout
	se_trx_sign	se_xchs_block_info
	se_trx_get_ctxinfo	se_xchs_block_btc
	se_trx_finish	se_xchs_cancel_block
	se_trx_outaddr	se_xchs_trxsign_login
	se_hdw_init_wallet	se_xchs_trxsign_prepare
	se_hdw_init_wallet_gen	se_xchs_trxsign_logout
	se_hdw_qry_wa_info	
AUTH	se_bind_reg_info	se_hdw_set_wa_info
	se_bind_reg_approve	se_hdw_create_account
	se_bind_reg_remove	se_hdw_qry_acc_info
	se_pin_change	se_hdw_set_acc_info
	se_pin_logout	se_hdw_next_trx_addr
	se_set_currency	se_hdw_prep_trx_sign
	se_get_currency	se_hdw_init_wallet_gen_confirm
	se_get_card_name	se_hdw_qry_acc_keyinfo
	se_set_card_name	se_xchs_reg_status
	se_get_secpo	se_xchs_reg_init
	se_set_secpo	se_xchs_reg_finish
	se_trx_status	se_xchs_reg_clear
	se_trx_prepare	se_xchs_get_otp
	se_trx_begin	se_xchs_session_init
	se_trx_verify_otp	se_xchs_session_estab
	se_trx_sign	se_xchs_session_logout
	se_trx_get_ctxinfo	se_xchs_block_info
	se_trx_finish	se_xchs_block_btc
	se_trx_outaddr	se_xchs_cancel_block
	se_hdw_init_wallet	se_xchs_trxsign_login
	se_hdw_init_wallet_gen	se_xchs_trxsign_prepare
	se_hdw_qry_wa_info	se_xchs_trxsign_logout
LOCK	se_bind_reg_info	se_puk_chlng
	se_bind_logout	se_pin_unlock
	se_perso_back_perso	
ERROR		
NOHOST	se_init_vmk_chlng	se_get_card_name

	se_init_back_init	se_set_card_name
	se_bind_reg_init	se_get_secpo
	se_bind_reg_chlng	se_hdw_qry_wa_info
	se_bind_reg_finish	se_hdw_qry_acc_info
	se_bind_reg_info	se_set_secpo
	se_pin_chlng	se_xchs_reg_status
	se_set_currency	se_xchs_get_otp
	se_get_currency	
DISCONN	se_init_vmk_chlng	se_pin_chlng
	se_init_back_init	se_set_currency
	se_bind_reg_init	se_get_currency
	se_bind_reg_chlng	se_get_card_name
	se_bind_reg_finish	se_set_card_name
	se_bind_reg_info	se_get_secpo
	se_bind_login_chlng	se_hdw_qry_wa_info
	se_bind_login	se_hdw_qry_acc_info
	se_bind_find_hst_id	se_xchs_reg_status
	se_set_secpo	se_xchs_get_otp
	se_bind_back_nohost	