Secure Element Applet API

Communication with a Secure element Applet API is performed through standard APDU commands.

For a detailed description of APDU communication, APDU commands data structure and particular bytes meaning, please refer to ISO/IEC 7816-4 standard.

Commands are grouped into three categories based on the type of usage:

1. General
2. Fiscalization
3. Audit

Important Notes

1. All APDU commands are sent to the Smart Card using T1 communication protocol
2. All amounts or counter values are submitted to/received from the Secure element using Big-endian. Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address)
3. P1 and P2 values considered in the request processing when,
    1. Select Applet Command
    2. force using CRC for Data in APDU transimission
4. PIN is sent in ASCII hex format from SE applet version 3.2.2.
5. CRC is available from SE applet version 3.2.5, and it is optional to use.
6. Invoice DateTime must be within Certificate validity NotBefore and NotAfter from applet version 3.2.8.
7. PIN can be sent in both ASCII and decimal hex format from applet version 3.2.9. as backward comatiblity. ASCII hex format is considerd default behaviour.

Content

1. General Commands
   Secure Element Applet is installed as a non-default applet on a smart card. Before any APDU command is invoked, the applet is selected using the standard Select command.

2. Fiscalization
   PIN verification is a method that “unlocks” a card for invoice signing and other operations protected by PIN code. Depending on the SE applet version, PIN is sent in decimal or hex format with ASCII encoding, and it is sent as an array of byte digits.

3. Audit
   Returns 259 bytes data structure represents public card key (256 bytes modulus and 3 bytes exponent). This key is used to encrypt Audit packages.

4.

4. Secure Element Specific APDU Error Codes
   This table contains the expected error codes and descriptions that a caller may encounter while working with the Secure Element Applet.

General Commands

Secure Element Applet is installed as a non-default applet on a smart card. Before any APDU command is invoked, the applet is selected using the standard Select command.

NOTE:
The availability of specific commands, as well as their content, depends on the secure element (SE) version. You can use the *Get Secure Element Version* command (see below) to check the version of the SE you are using.

Select Applet

As previously mentioned, the Smart Card has two applets installed. This command selects the Secure Element Applet and routes subsequent APDU commands to it.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expecte Lengt (Le)
>= 2.0.0	Case3Sho	0x00	0xA4	0x040(	0x10	0xA000000748	0x00

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0	none	0x9000

Example:

Request: `00A4040010A000000748464A492D546178436F726500`

Response: `9000`

Get Secure Element Version

This command returns the version information about the current Api version. The response contains 12 bytes, where each 4 bytes represent unsigned integer of one version segment, making total of 3 version segments: major, minor and patch.

APDU Request

SE CAP Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0	`Case2Sho`	`0x88`	`0x08`	`0x000(`	none	none	`0x00`

APDU Response

SE CAP Version	Response Data	SW1SW2
>= 2.0.0	*12 bytes*	`0x9000`

Example 1:

Request: `8808040000`

Response: `000000020000000000000000 9000`

Example 2:

Request: `8808000000`

Response: `000000030000000100000001 9000`

Example 3:

Request: `8808000000`

Response: `000000030000000200000005 9000`

Forward Secure Element Directive

This command is used by E-SDC to forward instructions received from TaxCore.Api to Secure Element Applet via Secure Element APDU Command.

If APDU Command status (SW1SW2) is OK (`0x9000`), consider forward instructions operation is completed.

NOTE:
From the SE version 3.2.5, optionally, CRC can be calculated and used for data verification. If CRC is not used, the command is the same as in the previous applet version.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0 (no CRC)	`Case3Ext`	`0x88`	`0x40`	`0x040`	`0x000200`	*512 bytes received from TaxCore*	none
>= 3.2.5 (with CRC)	`Case3Ext`	`0x88`	`0x40`	`0x010`	`0x000204`	*512 bytes received from TaxCore + 4 bytes for CRC*	none

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0 (no CRC)	none	`0x9000`
>= 3.2.5 (with CRC)	none	`0x9000`

Example 1 (without CRC):

Command Data:
`5DBFC9CD04AF9DC76C50FA3FF54D32D1910B0D2E1EC5AF97EAE3E71A7423CCE066D6E264255838C1DBAD`

Request:
`884004000002005DBFC9CD04AF9DC76C50FA3FF54D32D1910B0D2E1EC5AF97EAE3E71A7423CCE066D6E2`

Response: `9000`

Example 2 (with CRC):

Command Data witout CRC:
`5DBFC9CD04AF9DC76C50FA3FF54D32D1910B0D2E1EC5AF97EAE3E71A7423CCE066D6E264255838C1DBAD`

Command Data CRC: `F50CFF4B`

Command Data:
`5DBFC9CD04AF9DC76C50FA3FF54D32D1910B0D2E1EC5AF97EAE3E71A7423CCE066D6E264255838C1DBAD`

Request:
`884004000002045DBFC9CD04AF9DC76C50FA3FF54D32D1910B0D2E1EC5AF97EAE3E71A7423CCE066D6E2`

Response: `9000`

Export Certificate

This command exports the taxpayer certificate in a DER format. This certificate contains location data that is present on the textual representation of an invoice.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0	`Case2Ext`	`0x88`	`0x04`	`0x040(`	none	none	`0x000000`

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0	*raw bytes random length*	`0x9000`

Example:

Request: `88040400000000`

Response: *raw bytes of x509 certificate public key* + `9000`

Get Last Signed Invoice

This command returns information about the last singed invoice. The structure of the data recived is the same as the response is in the Sign Invoice command.

NOTE:
From the SE version 3.2.5, optionally, CRC can be calculated and used for data verification. If CRC is not used, the command is the same as in the previous applet version.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 3.1.1 (no CRC)	`Case2Ext`	`0x88`	`0x15`	`0x040(`	none	none	`0x000000`
>= 3.2.5 (with CRC)	`Case2Ext`	`0x88`	`0x15`	`0x0102`	none	none	`0x000000`

APDU Response

SE Version	Response Data	SW1SW2
>= 3.1.1 (no CRC)	*577 or 833 bytes*	`0x9000`
>= 3.2.5 (with CRC)	*581 or 837 bytes*	`0x9000`

Example 1 (without CRC):

Request: `88150400000000`

Response: *577 or 833 bytes* + `9000`

Response Data

Start (byte)	Length (bytes)	Field	Description
0	8	Date/time	Same as data sent from E-SDC to SE
8	20	Taxpayer ID	Same as data sent from E-SDC to SE
28	20	Buyer ID	Same as data sent from E-SDC to SE
48	1	Invoice type	Same as data sent from E-SDC to SE
49	1	Transaction type	Same as data sent from E-SDC to SE
50	7	Invoice amount	Same as data sent from E-SDC to SE

57	4	Sale or refund counter value	Depends on request's Tax type field
61	4	Total counter value (sale+refund)	Unsigned int 32bit big endian,
65	256 or 512	Encrypted Internal Data	Encrypted Internal Data length depends on the number of available tax rates programmed during personalization. It may be 256 or 512 bytes long.
321 or 577	256	Digital signature	

Example 2 (with CRC):

Request: `8815010200`

Response: *581 or 837* + `9000`

Response Data

Start (byte)	Length (bytes)	Field	Description
0	8	Date/time	Same as data sent from E-SDC to SE
8	20	Taxpayer ID	Same as data sent from E-SDC to SE
28	20	Buyer ID	Same as data sent from E-SDC to SE
48	1	Invoice type	Same as data sent from E-SDC to SE
49	1	Transaction type	Same as data sent from E-SDC to SE
50	7	Invoice amount	Same as data sent from E-SDC to SE
57	4	Sale or refund counter value	Depends on request's Tax type field
61	4	Total counter value (sale+refund)	Unsigned int 32bit big endian,
65	256 or 512	Encrypted Internal Data	Encrypted Internal Data length depends on the number of available tax rates programmed during personalization. It may be 256 or 512 bytes long.
321 or 577	256	Digital signature	

577 or 833	4	CRC	CRC is calculated from 0 to 577 or 833 bytes.

Get PIN tries left from SE Applet

This command returns how many PIN tries are left before the card is locked

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 3.1.1	`Case2Sho`	`0x00`	`0x16`	`0x040(`	none	none	`0x00`

APDU Response

SE Version	Response Data	SW1SW2
>= 3.1.1	05 if 5 tries are left, 00 if the card is blocked	`0x9000`

Example:

Request: `8816040000`

Response: `05 9000`

Get CertParams

This command returns UID, SE Certificate NotBefore and SE Certificate NotAfter. NotBefore and NotAfter are in UTC time in Unix Timestamp format.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 3.2.8	`Case2Sho`	`0x00`	`0x33`	`0x000(`	none	none	`0x00`

APDU Response

SE Version	Response Data	SW1SW2
>= 3.2.8	*24 bytes*	`0x9000`

Example:

Request: `8833000000`

Response: `445337584C535245000001968743CA28000001AC9386D1E8 9000`

CertParam	Hex	Transform Value
UID	445337584C535245	DS7XLSRE
NotBefore	000001968743CA28	04/30/2025 15:14:49
NotAfter	000001AC9386D1E8	04/30/2028 15:24:49

Fiscalization

PIN Verify

PIN verification is a method that “unlocks” a card for invoice signing and other operations protected by PIN code. Depending on the SE applet version, PIN is sent in decimal or hex format with ASCII encoding, and it is sent as an array of byte digits.

For example, PIN 1234 can be represented in the following formats:

- decimal format - PIN is represented as 0x01, 0x02, 0x03, 0x04.
- ASCII hex format - PIN is represented as 0x31, 0x32, 0x33, 0x34.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)

2.0.0 □ SE version < 3.2.2 3.2.9 □ SE version	Case3Sho	0x88	0x11	0x000(	0x04	*4 bytes where each represents one PIN digit in decimal format*	none
3.2.2 □ SE version	Case3Sho	0x88	0x11	0x000(	0x04	*4 bytes where each represents one PIN digit in ASCII hex format*	none

Example:

This is an example for PIN 1234.

SE Version	Command Data	Request	Response (correct PIN)	Error response (wrong PIN)
2.0.0 □ SE version < 3.2.2	01020304	88110000040102030	9000	6302
>= 3.2.2	31323334	88110000043132333	9000	6302

Sign Invoice

Signs invoice and returns fiscalization data for a submitted invoice.

NOTE:
From the SE version 3.2.5: Optional - CRC can be calculated and used for data verification. If CRC is not used, the command is the same as in the previous applet version.

From applet version 3.2.8: Mandatory - Invoice Date/time must be greater then Certificate NotBefore and lower then Certificate NotAfter.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length	Command Data	Expected Length

					(Lc)		(Le)
>= 2.0.0 (no CRC)	Case4Ext	0x88	0x13	0x040(	*3 byte Command Data byte array length*	*Command Data byte array*	0x0000
>= 3.2.5 (with CRC)	Case4Ext	0x88	0x13	0x0102	*3 byte Command Data byte array length*	*Command Data byte array + 4 bytes for CRC*	0x0000

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0 (no CRC)	*byte array*	0x9000
>= 3.2.5 (with CRC)	*byte array + 4 byte CRC*	0x9000

Data structure without CRC:

Command data:

Start (byte)	Length (byte)	Field	Description
0	8	Date/time	E-SDC timestamp UTC time in Unix Timestamp. Example: 1495018011910 is 2017-05-17T10:46:51.910Z
8	20	Taxpayer ID	Hex encoded byte array, leading bytes filled with 0x00. Taxpayer ID value can consist only of ascii printable characters. **Zeros can be added only on the left side**. MSB are sent first Example: Taxpayer ID = 928615467, Byte array = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x39, 0x32, 0x38, 0x36, 0x31, 0x35, 0x34, 0x36, 0x37} (byte 0x37 is sent last to SE)
28	20	Buyer ID	If unknown, leave zeroes. Formatting is the same as for Taxpayer ID

48	1	Invoice type	Values 0, 1, 2, 3, 4 as explained in section Create Invoice.
49	1	Transaction Type	Sale=0, Refund=1
50	7	Invoice amount	Sale or refund total amount (including taxes) - depends on applied tax types
57	1	Number of tax categories	Defines the number of tax categories which appear on the invoice (value between 0 and 26). The following data structure **Tax Categories** must be repeated exactly this number of times.
58	8	Tax Category (1)	The first Tax Category (mandatory if **Number of tax categories > 0**)
66	8	Tax Category (2)	The second Tax Category (mandatory if **Number of tax categories > 1**)
74	...	Tax Category (n)	

Tax Categories:

Start (byte)	Length (byte)	Field	Description
58	[1]	[Tax category ID]	The first tax category's OrderID, as explained in Tax Rates section (mandatory if **Number of tax categories** > 0)
59	[7]	[Tax category amount]	The first total tax amount for the category specified in preceding field **Tax category ID** (mandatory if **Number of tax categories** > 0)
66	[1]	[Tax category ID]	The next tax category's OrderID (mandatory if **Number of tax categories** > 1)
67	[7]	[Tax category amount]	The next total tax amount for the category specified in preceding field **Tax category ID** (mandatory if **Number of tax categories** > 1)

Response data:

Start (byte)	Length (bytes)	Field	Description
0	8	Date/time	Same as data sent from E-SDC to SE

8	20	Taxpayer ID	Same as data sent from E-SDC to SE
28	20	Buyer ID	Same as data sent from E-SDC to SE
48	1	Invoice type	Same as data sent from E-SDC to SE
49	1	Transaction type	Same as data sent from E-SDC to SE
50	7	Invoice amount	Same as data sent from E-SDC to SE
57	4	Sale or refund counter value	Depends on request's Tax type field
61	4	Total counter value (sale+refund)	unsigned int 32bit big endian,
65	256 or 512	Encrypted Internal Data	Encrypted Internal Data length depends on the number of available tax rates programmed during personalization. It may be 256 or 512 bytes long.
321 or 577	256	Digital signature	

Example without CRC:

Command Data:
`0000017B2D198AC400000000000000000050432D313030303030303031000000000000000000000000000000`

Request:
`8813040000009A0000017BE9B01AB400000000000000000050432D31303030303030303100000000000000`

Response: *byte array* + `9000`

Data structure with CRC:

Command data:

Start (byte)	Length (byte)	Field	Description
0	8	Date/time	E-SDC timestamp UTC time in Unix Timestamp. Example: 1495018011910 is 2017-05-17T10:46:51.910Z
8	20	Taxpayer ID	Hex encoded byte array, leading bytes filled with 0x00. Taxpayer ID value can consist only of ascii printable characters. **Zeros can be added only on the left side**. MSB are sent first Example:

			Taxpayer ID = 928615467, Byte array = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x39, 0x32, 0x38, 0x36, 0x31, 0x35, 0x34, 0x36, 0x37} (byte 0x37 is sent last to SE)
28	20	Buyer ID	If unknown, leave zeroes. Formatting is the same as for Taxpayer ID
48	1	Invoice type	Values 0, 1, 2, 3, 4 as explained in section Create Invoice.
49	1	Transaction Type	Sale=0, Refund=1
50	7	Invoice amount	Sale or refund total amount (including taxes) - depends on applied tax types
57	1	Number of tax categories	Defines the number of tax categories which appear on the invoice (value between 0 and 26). The following data structure **Tax Categories** must be repeated exactly this number of times.
58	8	Tax Category (1)	The first Tax Category (mandatory if **Number of tax categories > 0**)
66	8	Tax Category (2)	The second Tax Category (mandatory if **Number of tax categories > 1**)
74	...	Tax Category (n)	
...	4	CRC	CRC is calculated from 0 to 74 bytes (or to last byte if data).

Tax Categories:

Start (byte)	Length (byte)	Field	Description
58	[1]	[Tax category ID]	The first tax category's OrderID, as explained in Tax Rates section (mandatory if **Number of tax categories** > 0)
59	[7]	[Tax category amount]	The first total tax amount for the category specified in preceding field **Tax category ID** (mandatory if **Number of tax categories** > 0)
66	[1]	[Tax category ID]	The next tax category's OrderID (mandatory if **Number of tax categories** > 1)

67	[7]	[Tax category amount]	The next total tax amount for the category specified in preceding field **Tax category ID** (mandatory if **Number of tax categories** > 1)

Response data:

Start (byte)	Length (bytes)	Field	Description
0	8	Date/time	Same as data sent from E-SDC to SE
8	20	Taxpayer ID	Same as data sent from E-SDC to SE
28	20	Buyer ID	Same as data sent from E-SDC to SE
48	1	Invoice type	Same as data sent from E-SDC to SE
49	1	Transaction type	Same as data sent from E-SDC to SE
50	7	Invoice amount	Same as data sent from E-SDC to SE
57	4	Sale or refund counter value	Depends on request's Tax type field
61	4	Total counter value (sale+refund)	unsigned int 32bit big endian,
65	256 or 512	Encrypted Internal Data	Encrypted Internal Data length depends on the number of available tax rates programmed during personalization. It may be 256 or 512 bytes long.
321 or 577	256	Digital signature	
577 or 833	4	CRC	CRC is calculated from 0 to 577 or 833 bytes.

Example with CRC:

Command Data:
`0000017B2D198AC400000000000000000050432D313030303030303031000000000000000000000000000000`

Command Data CRC: `90F2BC39`

Request:
`88130102E0000017BE9B01AB400000000000000000050432D313030303030303031000000000000000000`

Response: *byte array invoice + 4 byte CRC +* `9000`

Amount Status

Returns 14-bytes-long data structure (7 bytes for sum SALE and REFUND, and 7 bytes for Limit Amount)

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0	`Case2Sho`	`0x88`	`0x14`	`0x040(`	none	none	`0x00`

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0	*14 byte array*	`0x9000`

Example:

Request: `8814040000`

Response: `0000724AA18328038D7EA4C68000 9000` (SALE+REFUND=490878370600 , Limit Amount=1000000000000000)

Audit

Export TaxCore Public Key

Returns 259 bytes data structure represents public card key (256 bytes modulus and 3 bytes exponent). This key is used to encrypt Audit packages.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)

>= 2.0.0	Case2Ext	0x88	0x07	0x040(	none	none	0x000000

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0	*259 bytes data*	0x0900

Example:

Request: 88070400000000

Response: *256 bytes modulus + 3 bytes exponent* + 9000

Export Audit Data

Exports encrypted audit data.

NOTE:
From the SE version 3.2.5, optionally, CRC can be calculated and used for data verification. If CRC is not used, the command is the same as in the previous applet version.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0 (no CRC)	Case2Ext	0x88	0x12	0x040(	none	none	0x000000
>= 3.2.5 (with CRC)	Case2Ext	0x88	0x12	0x0102	none	none	0x000000

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0 (no CRC)	*565 or 821 bytes data*	0x9000
>= 3.2.5 (with CRC)	*569 or 825 bytes data*	0x9000

NOTE:
Depending on the Internal Data, the total length of the structure is 565 or 821 bytes. For versions **3.2.5 or later** if CRC is used, the total lenght can be 569 or 825 if CRC is added.

Exported audit data has the following structure, without CRC:

Offset	Length	Data	Note
0	4	TaxCore Key Version	
4	256	Crypted Internal Data	The length of Crypted Internal Data can be 256 or 512 bytes
260 or 516	20	Taxpayer Identification Number (TIN)	
280 or 536	20	Buyer ID	
300 or 556	1	Invoice type	
301 or 557	1	Transaction type	
302 or 558	7	Invoice amount	
309 or 565	256	Digital signature of the above structure	

Exported audit data has the following structure, with CRC:

Offset	Length	Data	Note
0	4	TaxCore Key Version	
4	256	Crypted Internal Data	The length of Crypted Internal Data can be 256 or 512 bytes
260 or 516	20	Taxpayer Identification Number (TIN)	
280 or 536	20	Buyer ID	
300 or 556	1	Invoice type	
301 or 557	1	Transaction type	
302 or 558	7	Invoice amount	

309 or 565	256	Digital signature of the above structure	
565 or 821	4	CRC	CRC is calculated from 0 to 565 or 821 bytes.

Example 1 (without CRC):

Request: `88120400000000`

Response: *565 or 821 bytes* + `9000`

Example 2 (with CRC):

Request: `88120102000000`

Response: *569 or 825 bytes* + `9000`

Start Audit

Notifies the Secure element that the audit process has been initialized by E-SDC.

Secure element returns an encrypted message that shall be submitted to TaxCore as the content of the field `auditRequestPayload` of audit-proof request.

NOTE:
From the SE version 3.2.5, optionally, CRC can be calculated and used for data verification. If CRC is not used, the command is the same as in the previous applet version.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0 (no CRC)	Case2Ext	0x88	0x21	0x040(	none	none	0x000000
>= 3.2.5 (with CRC)	Case2Ext	0x88	0x21	0x0102	none	none	0x000000

APDU Response

SE Version	Response Data	SW1SW2

>= 2.0.0 (no CRC)	*260 bytes data*	`0x9000`
>= 3.2.5 (with CRC)	*264 bytes data*	`0x9000`

Example 1 (without CRC):

Request: `88210400000000`

Response: *260 bytes data* + `9000`

Example 2 (with CRC):

Request: `88210102000000`

Response: *260 bytes data + 4 bytes CRC data* + `9000`

End Audit

Notifies the Secure element that the audit process has been finalized by TaxCore. If APDU Command status is OK (0x90 0x00) consider the audit operation is completed.

NOTE:
From the SE version 3.2.5, optionally, CRC can be calculated and used for data verification. If CRC is not used, the command is the same as in the previous applet version.

APDU Request

SE Version	IsoCase	Class	Instruction	P1-P2	Command Length (Lc)	Command Data	Expected Length (Le)
>= 2.0.0 (no CRC)	`Case3Ext`	`0x88`	`0x20`	`0x040`	`0x000100`	*256 bytes received from TaxCore*	none
>= 3.2.5 (with CRC)	`Case3Ext`	`0x88`	`0x20`	`0x010`	`0x000104`	*256 bytes received from TaxCore + 4 bytes for CRC*	none

APDU Response

SE Version	Response Data	SW1SW2
>= 2.0.0 (no CRC)	none	0x9000
>= 3.2.5 (with CRC)	none	0x9000

Example 1 (without CRC):

Command Data:
253AB91A21859A06813E8A880E10BA0C67A09DDBED0B7E001F638CA015D2E414744E0C5C2E0F5F827DFC

Request:
882004000001002 53AB91A21859A06813E8A880E10BA0C67A09DDBED0B7E001F638CA015D2E414744E0C
9000

Example 2 (with CRC):

Command Data:
253AB91A21859A06813E8A880E10BA0C67A09DDBED0B7E001F638CA015D2E414744E0C5C2E0F5F827DFC

Command Data CRC: CEE700A0

Request:
882001020001042 53AB91A21859A06813E8A880E10BA0C67A09DDBED0B7E001F638CA015D2E414744E0C
9000

Secure Element Specific APDU Error Codes

This table contains the expected error codes and descriptions that a caller may encounter while working with the Secure Element Applet.

Error Code	APDU Command	Description	Error Code to POS
0x6301	Sign Invoice	PIN verification required before executing a command	1500
0x6302	Verify PIN	PIN verification failed – wrong PIN code	2100
0x6303	Verify PIN	Wrong PIN size	2100
0x6304	Sign Invoice	Maximum number of tax categories exceeded	SDC related
0x6305	Sign Invoice	Secure Element amount has reached the defined limit. The Secure Element is locked and	2210

		no additional invoices can be signed before the audit is completed.	
0x6306	End Audit	End Audit is sent but there is no active Audit	SDC related
0x6307	Sign Invoice	Invoice fiscalization is disabled by system	2210
0x6308	Sign Invoice	Invoice DateTime must be within Certificate validity NotBefore and NotAfter	SDC related
0x6310	Verify PIN	The number of allowed PIN entries exceeded	2110
0x63FF	Sign Invoice	A Secure Element counter has reached its limit. The Secure Element must be replaced.	SDC related
0x6700	End Audit	Data must be 256 bytes long	SDC related
0x6A80	End Audit	Proof of Audit command payload provided as APDU Command Data does not match the latest Start Audit one which Secure Element expects. Probably a new Start Audit was initiated after this one was ended.	SDC related
0x6A80	Sign Invoice	The tax category order id exceeds the maximum allowed for the Secure Element.	2310
0x6F00	End Audit	APDU Command Data cannot be recognized as a valid Proof of Audit	SDC related