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	Error response

			(correct PIN)	(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 (Lc)	Command Data	Expected Length (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	0x010:	*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:
`0000017B2D198AC40000000000000000050432D313030303030303031000000000000000000000000000000`

Request:
`8813040000009A0000017BE9B01AB40000000000000000050432D313030303030303031000000000000000`

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:
`0000017B2D198AC40000000000000000050432D313030303030303031000000000000000000000000000000`

Command Data CRC: `90F2BC39`

Request:
`88130102E0000017BE9B01AB40000000000000000050432D313030303030303031000000000000000000`

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)