Interoperability Specification for ICCs and Personal Computer Systems

Part 10 IFDs with Secure PIN Entry Capabilities

Gemalto
HID Global
NXP Semiconductors N.V.
Oracle America
SCM Microsystems

Revision 2.02.08
April 2010

AMENDMENT 1
2011-06-03

AMENDMENT 1:

PIN-Verification with Smart Cards based on PACE

Scope

The scope of this amendment is:
PIN-Verification with Contactless Smart Cards based on PACE.

Amendment 1 to PC/SC part 10, revision 2.02.08 from April 2010 was prepared by the members of PC/SC Workgroup, Tech Meeting, June 3, 2011, on request of SCM Microsystems, HID Global and KOBIL.

Interoperability Specification for ICCs and Personal Computer Systems

Part 10:
IFDs with Secure PIN Entry Capabilities

AMENDMENT 1: PIN-Verification with Contactless Smart Cards based on PACE

Clause 2.3 Definition of Features, page 3

Add the following rows to the table:

allocated, PC/SC defined usage	0x14 – 0x1F
FEATURE_EXECUTE_PACE	0x20
allocated, PC/SC defined usage	0x21 – 0x7F
non-allocated tags, for vendor specific use	0x80 – 0xFE
future use	0xFF

Replace the object string with:

01 04 XX XX XX XX 02 04 XX XX XX 03 04 XX XX XX XX 04 04 XX XX XX XX 05 04 XX XX XX XX 06 04 XX XX XX XX 07 04 XX XX XX XX 08 04 XX XX XX XX 09 04 XX XX XX XX 0A 04 XX XX XX XX 0B 04 XX XX XX XX 0C 04 XX XX XX XX 0D 04 XX XX XX XX 0E 04 XX XX XX XX 0F 04 XX XX XX XX 10 04 XX XX XX XX 11 04 XX XX XX XX 12 04 XX XX XX XX 13 04 XX XX XX XX 20 04 XX XX XX XX

Clause 2.5 Structure list, page 12

Add the following chapter:

2.5.12 EXECUTE_PACE

The InBuffer structure is defined in the following table:

Byte offset	Field	Type	Description
0	idxFunction	BYTE	Index of PACE function 1 – GetReaderPACECapabilities 2 – EstablishPACEChannel 3 – DestroyPACEChannel
1	length_InputData	USHORT	Length of InputData
3	InputData	BYTE[]	Depends on function (Field idxFunction)

The OutBuffer structure is defined in the in the next table:

Byte offset	Field	Type	Description
0	Result	ULONG	See table below (result values)
4	length_OutputData	USHORT	Length of OutputData
3	OutputData	BYTE[]	Depends on function (see InBuffer)

Table for result values:

Result	***Description***
0x00000000	No error
Errors in input data	
0xD0000001	Inconsistent lengths in input
0xD0000002	Unexpected data in input
0xD0000003	Unexpected combination of data in input
Errors during protocol execution	
0xE0000001	Syntax error in TLV response
0xE0000002	Unexpected or missing object in TLV response
0xE0000003	Unknown PIN-ID
0xE0000006	Wrong Authentication Token
Response APDU of the card reports error (status code SW1SW2)	
0xF000SW1SW2	Select EF.CardAccess
0xF001SW1SW2	Read Binary EF.CardAccess
0xF002SW1SW2	MSE: Set AT
0xF003SW1SW2	General Authenticate Step 1-4
Others	
0xF0100001	Communication abort (e.g. card removed during protocol)
0xF0100002	No card
0xF0200001	Abort
0xF0200002	Timeout
Additional application dependent error codes may be used.	

Clause 2.6 Feature list, page 25

Add the following chapter:

2.6.16 FEATURE_EXECUTE_PACE

This new feature is defined for support of PACE. The PACE protocol is a secure channel between the IFD and the smart card. PACE is described in [TR-03110] and [ICAO].
This chapter describes only how an application can trigger the IFD PACE protocol.

GetReaderPACECapabilities

The command GetReaderPACECapabilities is used to query the PACE support of the IFD. The result is given as a bit mask.

- 0x80 denotes support of explicit DestroyPACEChannel [1]);
- 0x40 denotes generic PACE support;

Other bits denote the support of application specific extensions of the protocol, e.g.

- 0x20 denotes support for the eID application of the German eID-Card,
- 0x10 support for the qualified electronic signature function on contactless cards.

These extensions may comprise support for specialized coding of further parameters, using a display of the IFD or performing additional cryptographic functions.

[1]) Note: Not all IFD's support this feature, because these IFD's destroy the PACE channel implicitly after the transaction.

InputData: None.

OutputData:

Number	***Type***	***Name***	***Description***
1	BYTE	length_BitMap	Length of BitMap
2	BYTE[]	BitMap	0x40 – The IFD supports PACE 0x80 – The IFD supports DestroyPACEChannel other – Application specific extensions supported

EstablishPACEChannel

The command EstablishPACEChannel is used to trigger the execution of the PACE protocol between the chip and the IFD. The result of this command is the verification of the PIN and the establishment of a secure messaging channel between the chip and the IFD. Until secure messaging is stopped, the IFD must encrypt/decrypt APDUs received from the host/the chip before forwarding them to the chip/the host, respectively.

Note: If no PIN is given in InputData, the PIN must be entered on the secure PIN entry device of the IFD by the user. The key $K\pi$ is derived from the PIN for use in PACE.

Input data of EstablishPACEChannel is defined as follows:

Number	***Type***	***Name***	***Description***
1	BYTE	PinID	0x01: MRZ 0x02: CAN 0x03: PIN 0x04: PUK
The following elements are only present if the execution of PACE is to be followed by an execution of Terminal Authentication Version 2 as defined in [TR-03110]			
2	BYTE	length_CHAT	Length of CHAT
3	BYTE[]	CHAT	Role Identifier and Certificate Holder Authorization Template, see [TR-03110]
If the PIN to be used is not secret (e.g. printed on the card/stored in the host), it may be delivered by the host to the IFD in the following elements. A suitable command filter should be employed by the IFD to refuse delivery of secret PINs by the host.			
4 + Len	BYTE	length_PIN	Length of PIN
5 + Len	BYTE[]	PIN	Password given by the host
Further application dependent data may follow. Depending on the data additional actions may be performed by the IFD during the execution of EstablishPACEChannel.			

Output data of EstablishPACEChannel is defined as follows:

Number	***Type***	***Name***	***Description***
1	USHORT	Status	Status codes of MSE:Set AT
2	USHORT	length_CardAccess	Length of EF_CardAccess
3	BYTE[]	EF_CardAccess	Contents of EF.CardAccess as read from the chip
The following elements are only present if the execution of PACE is to be followed by an execution of Terminal Authentication Version 2 as defined in [TR-03110]. These data are needed to perform the Terminal Authentication.			
4	BYTE	length_CARcurr	Length of CARcurr
5	BYTE[]	CARcurr	Current Certificate Authority Reference
6	BYTE	length_CARprev	Length of CARprev
7	BYTE[]	CARprev	Previous Certificate Authority Reference
8	USHORT	length_IDicc	Length of IDicc
9	BYTE[]	IDicc	Ephemeral PACE public key of the IFD

DestroyPACEChannel

The command DestroyPACEChannel terminate the PACE channel.

Note: Not all IFD's support this feature, because these IFD's destroy the PACE channel implicitly after the transaction.

InputData: None.
OutputData: None.

Clause 3, Abbreviations, page 26

Add the following abbreviations in alphabetic order:

PACE	Password Authenticated Connection Establishment
MRZ	Machine Readable Zone
CAN	Card Access Number
PUK	PIN Unblock Key
EF	Elementary File
CAR	Certification Authority Reference

Clause 4, References, page 27

Add the following references:

[TR-03110] Technical Guideline TR-03110, Advanced Security Mechanisms for Machine Readable Travel Documents - Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE) and Restricted Identification (RI), https://www.bsi.bund.de/ElektronischeAusweiseTR

[ICAO] ICAO: Technical Report "Supplemental Access Control", http://mrtd.icao.int

