This document is preliminary and is subject to change without prior notice.

As this version of the application is in final phase of the development, the current document could continue to evolve until September 2004.
However, no fundamental changes should be introduced in order to minimise the impact on parties using the document as a reference.

Written by: Sylvain Lhostis – Project Leader - Axalto
Frédéric Leprieur – Software Engineer - Axalto
Marc Stern – Senior Consultant - CSC

Approved by: Raphael Rosset – Program Manager - Axalto
Johan Rommelaere – Project Manager - Zetes

TABLE OF CONTENT

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 DOCUMENT OUTLINE

The purpose of this document is to define the functional specification of the BelPIC electronic identity card.

1.2 REFERENCES

1.2.1 External References [ER]

[ER1]	ISO CEI 7816-3	Integrated circuits cards with contacts – Electronic signals and transmission protocols – 1997
[ER2]	ISO CEI 7816-4	Integrated circuits cards with contacts – Inter-industry commands for interchange – 1995 + amendment 1- 1997
[ER3]	ISO CEI 7816-5	Integrated circuits cards with contacts – Inter-industry commands for interchange – 1996
[ER4]	ISO CEI 7816-8	Integrated circuits cards with contacts – Security related inter-industry commands – 1999
[ER5]	ISO CEI 7816-9	Integrated circuits cards with contacts – Additional inter-industry commands and security attributes – 2000

[ER6]	Java Card 2.1.1	Java Card 2.1 Application Programming Interface – Final revision 1.1– June 7, 1999

[ER7]	Open Platform	Card Specification – version 2.0.1 – April 7, 2000
[ER8]	PKCS#15 1.1	Cryptographic Token Information Standard, version 1.1, RSA laboratories, June 2000 URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-15/pkcs-15v1_1.pdf

[ER9]	Appel d'offres général pour la fabrication, la personnalisation, l'initialisation et la distribution des cartes d'identité digitales et pour la fourniture de services de certification	Cahier Spécial des Charges, ref. RRN/006/2001, version 2.5 dated from the 08/04/02
[ER10]	Technical specifications for the BelPIC electronic identity card chip, Annex 5	Appendix 5, 15/10/2001, version 2.1

[ER11]	PKCS#1 2.0	RSA Cryptography Standard, version 2.0, RSA laboratories, September 1998 URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-0a1.pdf
[ER12]	PKCS#1 2.1	RSA Cryptography Standard, version 2.1, RSA laboratories, June 2002 URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf

[ER13]	ISO CEI 9564-1	Banking – Personal Identification Number (PIN) management and security – 2002
[ER14]	ISO/IEC 9798-3	Information technology – Security techniques – Entity authentication Part3: Mechanisms suing digital signature techniques Second edition 1998-10-15

1.3 DEFINITIONS

Nibble	Half part of a byte (4 bits).
Digit	Nibble taking values between 0 and 9
Temporary mute state	Mute until next reset
Local key or PIN	Key or PIN belonging to the currently selected application. This key or PIN cannot be shared with other applications
Global key or PIN	Key or PIN that can be used by all applications. Note that global key or PIN are created during BelPIC application creation, other new applications cannot create global key or PIN objects.

1.4 ACRONYMS AND ABBREVIATIONS

For the purposes of this document, the following abbreviations apply:

AID	Application provider identifier
AODF	Authentication object directory file
APDU	Application protocol data unit
API	Application programming interface
BCD	Binary-coded decimal
CA	Certification Authority
CC	Cryptographic Checksum
CG	Cryptogram
CDF	Certificate directory file
DF	Dedicated File
DO	Data Object
DODF	Data object directory file
DST	Digital Signature Template
DTBS	Data to be signed
EID	Electronic Identity Device
EF	Elementary file
FCI	File Control Information
FCP	File Control Parameter
IFD	Interface device (e.g. reader)
MF	Master file
MSB	Main significant byte
ODF	Object directory file
OP	Open Platform
OS	Operating system
PIN	Personal identification number
PUK	PIN Unblocking Key
PuK	RSA public key
PrK	RSA private key
PrKDF	Private key directory file
PuKDF	Public key directory file
RID	Registered application provider identifier
SE	Security Environment
VOP	Visa Open Platform (refer to **[ER07]**)

1.5 METHODOLOGY AND CONVENTIONS

- All the values between quotations are in hexadecimal notation.
- The values are in MSB first.
- The symbol || represents a concatenation.

1.6 REMARKS ON STANDARDS USED

PKCS#1 version used is 2.1 (see [ER12]).

Two signature schemes are available:

- ***RSASSA-PKCS1-v1.5 using SHA1 or MD5***
- ***RSAPSS-PSS using SHA1*** defined in ***PKCS#1-v2.1 (refer to §2.3)***

Two deciphering algorithms are available***:***

- ***RSAES-PKCS1-v1.5-Decrypt***
- ***RSAES-OAEP using SHA1*** defined in ***PKCS#1-v2.1 (refer to §2.3)***

2. GENERAL DESCRIPTION OF THE APPLICATION

The BelPIC application is composed of two applications:

- The electronic signature application
- The electronic identification application

The MF is selected after a reset.

The BelPIC security environment has to be explicitly set after a reset.

PIN objects and key objects used by the application are local to the application (DF) unless specified otherwise.

2.1 ASSUMPTIONS

- Only application (DF) deletion is supported, file deletion is not.
- The PKCS#15 information contained by the BelPIC file system is not processed; this information is managed and used only by the external application (middleware).
- Only the transparent elementary files are supported.
- The format of the public keys stored in the chip is RSA 2048.
- The format of the private keys stored in the chip is RSA 1024 with CRT.

2.2 SECURITY CONDITIONS

The following table lists the different security conditions of the EID card:

	Type Meaning
→	The operation is not possible
NEV	The operation is never allowed
ALW	The operation is always allowed
CHV	The operation is only allowed during the operational phase after a successful PIN verification (refer to 4.1).
EXA	The operation is only allowed during the operational phase after a successful signature based external authentication (refer to 0).

Only one security condition can be fulfilled at a time in the active DF.

2.3 RSASSA-PSS AND RSASES-OAEP ALGORITHMS DESCRIPTION

Both algorithms are defined in **[ER12]**.

- Common to RSASSA-PSS and RSAES-OAEP schemes:

The MGF function is a Mask Generation Function based on a hash function.

The MGF function used for RSASSA-PSS-Encode, RSASSA-PSS-Verify and RSAES-OAEP-Decrypt is the one proposed in **[ER12]**: MGF1, based on SHA-1 hash function.

- The RSASSA-PSS algorithm is used in a signature scheme.

The input message for signature must be a hash of the complete message using SHA-1.

The RSASSA-PSS algorithm uses the EMSA-PSS-encode and EMSA-PSS-verify methods.
These methods are parameterised by the choice of hash function, mask generation function, and salt length.
The choices made are:

Parameter	Implementation	Data Size	Comment
HASH n°1 function	SHA-1	20 Bytes	-
SALT length	Random Generator	20 bytes	Strong random generation in the card.
MGF function	MGF1()	N/A	The internal HASH n°2 function is SHA-1, the initial message to hash for each round depends on the round number, as described in **[ER12]**.

The encoded message length is directly linked to the length of the RSA modulus.
- The RSASSA_PSS-encode applies for RSA keys with modulus length 1024 bits.
- The RSASSA_PSS-verify applies for RSA keys with modulus length 2048 bits.

- The RSAES-OAEP algorithm is used in an encoding scheme.

Only RSAES-OAEP-Decrypt is used inside the card.

RSAES-OAEP is parameterised by the choice of hash function and mask generation function.
The choices made are:

Parameter	Implementation	Data Size	Comment
HASH n°1 function	SHA-1	20 Bytes	-
SEED length	Random Generator	20 bytes	Strong random generation in the card.
MGF function	MGF1()	N/A	The internal HASH n°2 function is SHA-1, the initial message to hash for each round depends on the round number, as described in **[ER12]**.
Label	No Label used	0 Byte	The Label is a fixed zero length string.

The encoded message length is directly linked to the length of the RSA modulus.

The RSAES-OAEP-Decrypt applies for RSA keys with modulus length of 1024 bits.

The RSAES-OAEP-Decrypt returns the decrypted data without the padding.

3. KEYS AND PIN CODES DESCRIPTION

3.1 RSA PRIVATE KEY DESCRIPTION

All private RSA keys are 1024 bit-long.

Two types of private RSA keys are defined: Signature key and Decipher key:

RSA private key	Application
Signature	• Used to sign exclusively using PSO Compute Digital Signature (i.e. using a signature algorithm) • Can be used for Internal Authenticate command • Deciphering with this key is forbidden, i.e. using a non signature algorithm is forbidden
Decipher	• Used to decipher exclusively using PSO Decipher • Signing with this key is forbidden

Table 1 - Private RSA key description

RSA private key	Local / Global	Number of keys	Mandatory/Optional
Signature	Local	Any in BelPIC None in other DF	Optional
Decipher	Local	Any	Optional

Table 2 - Private RSA key properties

Note: the ***Get Card Data*** and ***Get PIN Status*** commands sign the status with the key '81'.

Command on RSA private key	*PSO: Compute Digital Signature*	*PSO: Decipher*
Signature	AC	✗
Decipher	✗	AC

Table 3 - Private RSA key access condition definition

AC: Access control to be defined during key creation

✗ : Not accessible

3.2 RSA PUBLIC KEY DESCRIPTION

All public RSA keys are 2048 bit-long.

	Application
RSA public key	• Used in External Authenticate command to get EXA security conditions

Table 4 - Public RSA key description

	Local / Global	Number of keys	Mandatory/Optional
RSA public key	Global or local	any	Mandatory if ***External Authenticate*** is used

Table 5 – Public RSA key properties

	External Authenticate
RSA public key	AC

Table 6 - Public key access condition definition

AC: Access control to be defined during key creation

❌: Disabled

3.3 PIN DESCRIPTION

Several PIN can be created, in any DF:

- Each PIN is totally independent from any other PIN in the same DF or another one.
- At creation time, each PIN may be linked to one **$PUK_{unblock}$** and one **PIN_{reset}** from the same DF. If some global **$PUK_{unblock}$** and **PIN_{reset}** exist, they may also be used to link to.

PIN length are always 8-byte long.

PIN	Application
$PIN_{permanent}$	• The PIN access right granted is permanent until current access condition specifically change (card reset, logoff, external authenticate…)
$PIN_{transient}$	• The PIN access right granted is available for the next command only.
$PUK_{unblock}$	• Used to unblock **$PIN_{permanent}$** and **$PIN_{transient}$**
PIN_{reset}	• Used to Set the **$PIN_{permanent}$** and **$PIN_{transient}$** to random code

Table 7 – PIN description

PIN	Local / Global	Number of PIN	Mandatory/Optional
$PIN_{permanent}$	Local	Any	Optional
$PIN_{transient}$	Local	Any	Optional
$PUK_{unblock}$	Local or Global	Any[1]	Optional
PIN_{reset}	Local or Global	Any[2]	Optional

Table 8 – PIN properties

	MVP: Change Reference Data
PIN	CHV + PIN_{reset} reference

Table 9 - PIN access condition definition

[1] or none
[2] or none

PIN format

The PIN are 8-bytes strings with the following format (by nibble) as defined in ER[07] and [ER13]:

C	L	P	P	P	P	P/'F'	P/'F'	P/'F'	P/'F'	P/'F'	P/'F'	P/'F'	P/'F'	'F'	'F'

Nibble	Signification
C	Control parameter, contains always '2'
L	Length of the PIN (in nibbles) ; from '4' to 'C'
P	Four mandatory digits
P/'F'	The rest of the PIN digits depending on the length, 'F' else
'F'	Padding contains always 'F'

Table 10 - PIN format

PIN	Minimum number of digits
$PIN_{permanent}$	4
$PIN_{transient}$	4
$PUK_{Unblock}$	12
PIN_{reset}	12

Table 11 - PIN length (in digits)

Note : The minimum PIN length is checked at key creation and when PIN is changed.

4. AUTHENTICATION PROCESSES

4.1 PIN VERIFICATION

The PIN verification consists in verifying a PIN code from an external application against the reference data stored into the EID card. If this verification process succeeds then the card grants the access right associated to the PIN reference.

The CHV process uses the ***MVP: Verify (ISO 7816-4)*** APDU command:

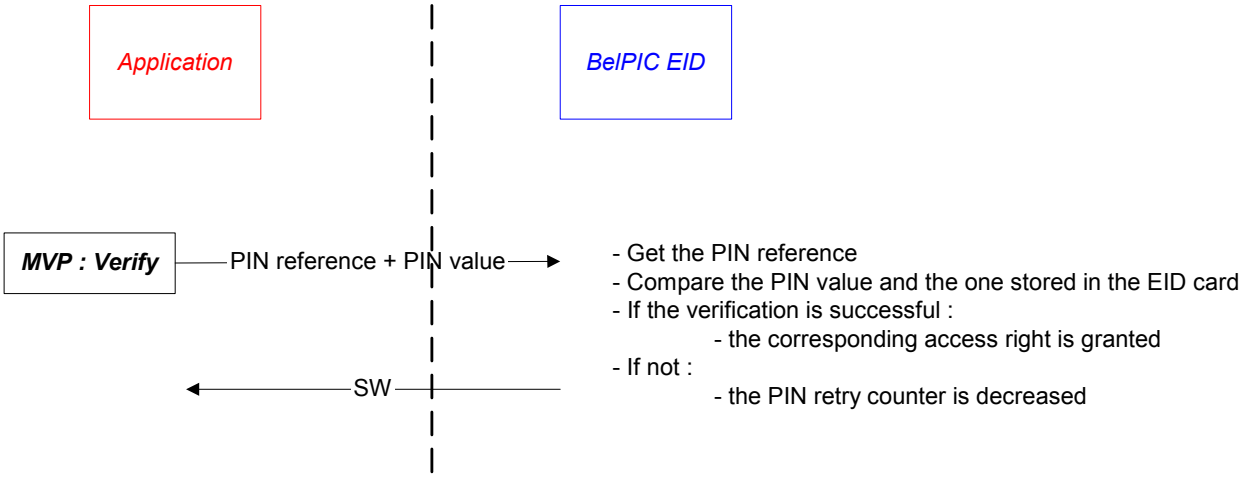

Figure 1 - PIN verification process

PIN verification sequence

Two types of PIN are defined: **$PIN_{permanent}$** and **$PIN_{transient}$**.

- Using the **$PIN_{permanent}$**, the PIN access right granted is permanent until current access condition specifically change (card reset, logoff, external authenticate…). This is the usual way of using a PIN.
- Using the **$PIN_{transient}$**, the PIN access right granted is available for the next command only.

Note:

Considering updating a file with **$PIN_{transient}$** access condition implies that the **$PIN_{transient}$** must be verified before each ***Update Binary*** command.

Examples of an EF (Elementary File) protected by a PIN for the command "*Read Binary*":

- OK: Select(EF) , Verify(**$PIN_{permanent}$**), Read(EF), Read(EF), Read(EF)
- OK: Verify(**$PIN_{permanent}$**), Select(EF), Read(EF), Read(EF), Read(EF)
- OK: Select(EF), Verify(**$PIN_{transient}$**), Read(EF), Verify(**$PIN_{transient}$**), Read(EF), Verify(**$PIN_{transient}$**), Read(EF)
- NOK: Verify(**$PIN_{transient}$**), Read(EF), Read(EF) , Read(EF)
- NOK: Verify(**$PIN_{transient}$**), Select(EF), Read(EF)

4.2 EXTERNAL AUTHENTICATION

The external authentication is the process whereby the card authenticates the external application by means of a signature based on challenge/response authentication scheme. If this verification process succeeds then the external card application gets access to the authorized data and functions in the EID card.

The application private key must be 2048 bits long.

The access right of the referenced key related to External Authenticate must be fulfilled prior using the command.

The external authentication process uses the ***Get Challenge (ISO 7816-4)*** and ***External Authenticate (ISO 7816-4)*** APDU commands

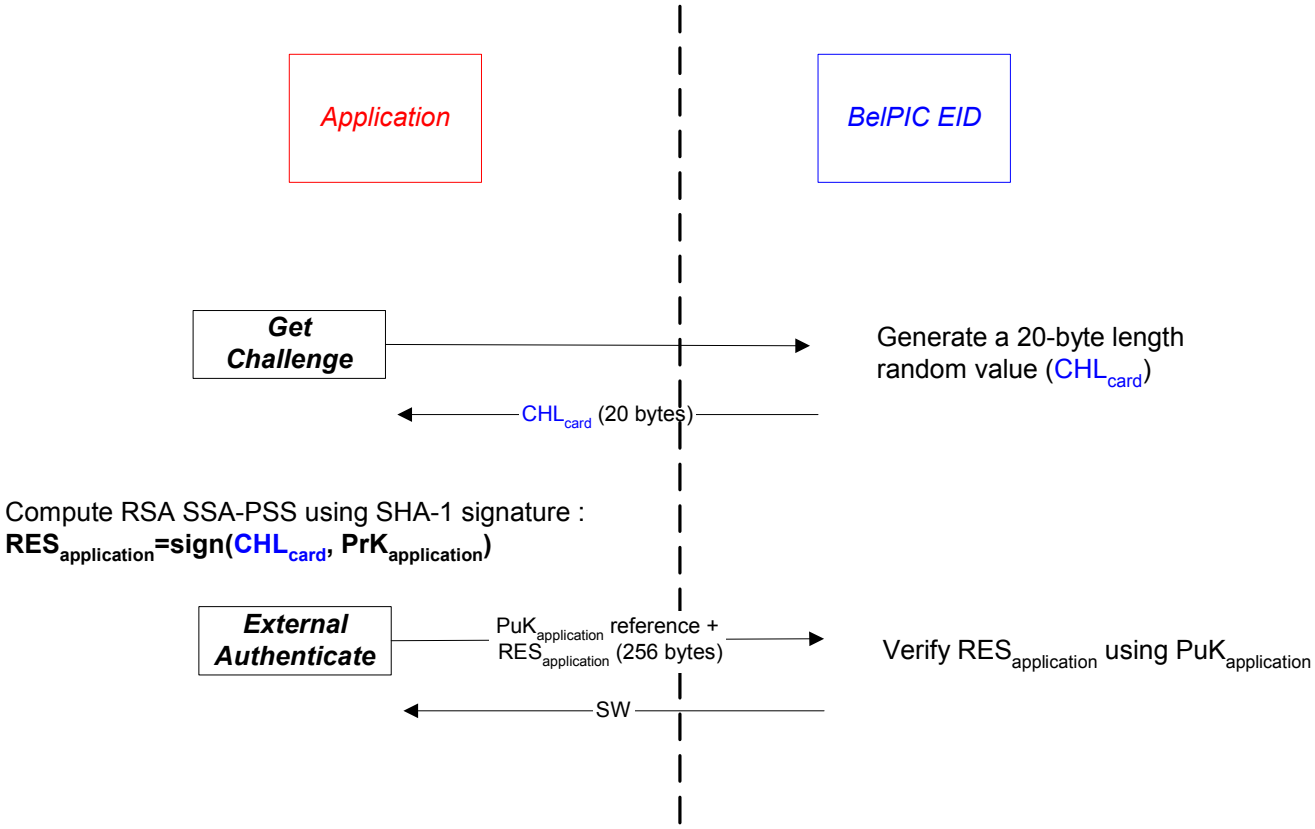

Figure 2 - External Authentication without certificate verification process

Compatibility note

In application V1:

- The application private key had to be 1024 bits long

5. COMMAND INTERFACE

The following table shows the APDU interface of the available commands according to the Application life cycle (operational phase).

The column "V1" indicates if the command was available in – or changed from – application V1.

Instruction	CLA	INS	P1	P2	Lc	Input Data field	Le	Output Data field	V1
GET RESPONSE	'00'	'C0'	'00'	'00'	-	-	Length	response from previous command	✔
SELECT FILE	'00'	'A4'	'02' or '04' or '08'	'0C'	Length	File Id or AID	-	-	✔
READ BINARY	'00'	'B0'	OFF_H	OFF_L	-	-	Length	Read data	✔
UPDATE BINARY	'00'	'D6'	OFF_H	OFF_L	Length	Data to update	-	-	✔
ERASE BINARY	'00'	'0E'	OFF_H	OFF_L	-	-	-	-	✔
MVP: VERIFY	'00'	'20'	'00'	Data Ref.	Length	Verification Data	-	-	✔
MVP: CHANGE REFERENCE DATA	'00'	'24'	'00'	Data Ref.	Length	Existing PIN || New PIN Or New PIN	-	-	✔
GET CHALLENGE	'00'	'84'	'00'	'00'	-	-	Length	Random value	✔
EXTERNAL AUTHENTICATE	'00'/'10'	'82'	Algo Ref	Data Ref.	Length	Signature (+ certificate) to verify	-	-	☑
MSE: SET	'00'	'22'	'41'	'B6'/'B8'	Length	Digital signature template	-	-	✔
PSO: COMPUTE DIGITAL SIGNATURE	'00'	'2A'	'9E'	'9A'	Length	Data to be signed	Length	Signature	☑
PSO: DECIPHER	'00'	'2A'	'80'	'86'	'80'	Data to be deciphered	Length	Deciphered data	✘
GET CARD DATA	'80'	'E4'	'00' / '02'	'00'	-	-	Length	Card information	☑
GET PIN STATUS	'80'	'EA'	'00' / '02'	Data Ref.	-	-	Length	PIN status	✘
LOG OFF	'80'	'E6'	'00	'00'	-	-	-	-	✔

Table 12 - APDU Commands (operational phase)

✔	**Available**
☑	**Available, but modified**
✘	**Not available**
NA	**Not applicable**

5.1 PROTOCOL FOR T=0 (ISO 7816-3)

The card currently only implements the protocol "***T=0***", which does not support input and output data in the same command (cf. ISO 7816-3). Such commands – referred as ***case 4*** commands – must be called without the ***Le*** parameter and return a Status Word '***61 xx***' where '***xx***' is the length of the output data to retrieve in an additional command. This protocol-level only command to use is ***Get Response*** (see 5.2).

The ***Get Response*** command may also be used in case of ***case 2*** and ***case 4*** commands where available output is greater than 256 bytes. In this case a following ***Get Response*** allows to get all available data.

Remarks:

- Take care that some readers hardly accept to receive 256 bytes from the card when Le=00, if not sure prefer using Le='FF' and perform an other following ***Get Response*** command.

5.2 GET RESPONSE COMMAND

5.2.1 Get Response Description

T=0 protocol

This command retrieves the data output by a ***case 4*** command.

T=0 and T=1 protocols

When a ***case 2*** or a ***case 4*** command has more than 256 bytes to output, it issues a '***61 xx***' status word – where '***xx***' is the number of bytes still available. A '***61 00***' status word means that at least 256 bytes are available). The '***xx***' bytes available must be retrieved using subsequent ***Get Response***.

It is possible to chain ***Get Response*** to get more than 256 bytes.

No security conditions are required to perform this command.

Note: ***Get Response*** command must follow immediately the command issuing the '***61 xx***' status word, otherwise the output information is lost.

Compatibility note

The ***Get Response*** command chaining was not available in application V1.

5.2.2 Command structure

Instruction	CLA	INS	P1	P2	Le	Case
GET RESPONSE	'00'	'C0'	'00'	'00'	Length	2

Command ADPU

Field	Value
CLA	'00'
INS	'C0'
P1	'00'
P2	'00'
Le	Length of the data to retrieve

Table 13 – GET RESPONSE Command APDU

Response APDU

Field	Value
Data	Data to retrieve
SW1-SW2	Status Bytes

Table 14 – GET RESPONSE Response APDU

Status bytes

Value	Meaning
’61 **xx**’	**xx** remaining bytes to retrieve
’6C **xx**’	***Le*** too long, only **xx** bytes available (hexadecimal value)
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 15 – GET RESPONSE Status bytes

5.3 COMMAND CHAINING

When the length of the data to be send to the card is greater than the maximum data field length (i.e. 255 bytes), the command chaining has to be used.

The only APDU command that supports the command chaining described here is:

- EXTERNAL AUTHENTICATE,

When the process has been initiated it must be completed before any commands; otherwise the process is discarded.

During the process, if the card sends an error status word, all the sequence must be redone from the beginning.

The command chaining is defined in **[ER4]**.

CLA coding (bit field)	**Meaning**
'xxx0xxxx'	For the last (or only) command involved
'xxx1xxxx'	For a command which is not the last command

Table 16 - Coding for command chaining

During command chaining each command must have the same value for the 'x' bits in the CLA byte.

If SW1-SW2 is set to '***90 00***' in a response to a command that is not the last command, then it means that the processing has been successful so far.

If the parameter P1 or parameter P2 differ from the previous command that is part of the same chaining, the chaining aborts, and an error is generated.

5.4 CODING OF THE ALGORITHM REFERENCES

In the BelPIC application, the algorithm references used by the command interface is coded as follow:

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Meaning
0	0	0	0	0	0	0	1	RSASSA-PSS using SHA1
0	0	0	0	0	0	1	0	RSASSA-PKCS1-v1.5 using SHA1
0	0	0	0	0	1	0	0	RSASSA-PKCS1-v1.5 using MD5
1	0	0	1	0	0	0	0	RSAES-PKCS1-v1.5 private key decryption (padding computed off-card)
1	0	1	0	0	0	0	0	RSAES-OAEP private key decryption
1	0	1	1	0	0	0	0	RSA-KEM private key– RFU

Table 17 - Algorithm references

Compatibility note

Only RSASSA-PKCS1-v1.5 algorithms were available in application V1.

5.5 CODING OF THE KEY AND PIN OBJECTS REFERENCE

The data objects are divided into two categories

- Key objects
- PIN objects

The following table show the references of the objects known by the application.

Number	Meaning	Usage
'01' or '81'	Reference PrK#1 (Basic key)	Card and PIN status signature
'07' or '87'	Reference PuK#7	Role CA key

Table 18 - Key and PIN object numbers

The following table describes the coding of local or global reference.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Meaning
0	0	0	0	0	0	0	0	No information is given (RFU)
0	-	-	-	-	-	-	-	Global reference data
1	-	-	-	-	-	-	-	Local reference data
-	X	X	-	-	-	-	-	'00' others are RFU
-	-	-	X	X	X	X	X	Data object number

Table 19 - Data object references (IS0 7816)

Important Note: A global object is the local object of BelPIC application.

For an application, a global object with a given data object number refers to the local object of BelPIC application with the same data object number.

In BelPIC application a local or global reference with the same data object number refer to the same object.

5.6 SELECT FILE COMMAND (ISO 7816-4)

5.6.1 Description

This command is used to select a file from the file system according to:

1. A file identifier, for EF selection
2. An application identifier (AID), for DF selection
   Remark: For compatibility with version 1, the short reference of the ID DF ('DF 01') is still supported.

No security conditions are required to perform this command.

If a select file is performed, current access conditions are unchanged.

If a select application is performed:

- The PIN access rights of the previously selected application are kept,
- The local EXA access rights of the previously selected application are lost

5.6.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
SELECT FILE	'00'	'A4'	'02' or '04'	'0C'	Length	-	3

Command ADPU

Field	Value
CLA	'00'
INS	'A4'
P1	1. '02' (the data field contains a File ID) 2. '04' (the data field contains an AID)
P2	'0C': (No FCI to be returned)
Lc	Length of the subsequent data
Data	1. File ID (2 bytes) 2. Full AID (between 5 and 16 bytes)
Le	Empty

Table 20 – SELECT FILE Command APDU

Remark: The BelPIC AID of the EID card is: A000000177504B43532D3135.

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 21 – SELECT FILE Response APDU

Status bytes

Value	Meaning
’62 83’	Selected file not activated
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
‘6A 82’	File not found
‘6A 86’	Wrong parameter P1-P2
‘6A 87’	Lc inconsistent with P1-P2
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 22 – SELECT FILE Status bytes

5.7 READ BINARY COMMAND (ISO 7816-4)

5.7.1 Description

This command is used to read the content of a transparent EF.

The transparent EF must be currently selected and activated.

The security conditions to fulfil before performing this command depend on the current selected file.

5.7.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
READ BINARY	'00'	'B0'	OFF_H	OFF_L	-	Length	2

Command ADPU

Field	Value
CLA	'00'
INS	'B0
P1	OFF_H: Higher byte of the offset (bit 8 =0)
P2	OFF_L: Lower byte of the offset
Lc	Empty
Data	Empty
Le	Length of the data to read

Table 23 – READ BINARY Command APDU

Note: If ***Le*** is equal to 0, then it is interpreted as 256 bytes.

Response APDU

Field	Value
Data	Read data
SW1-SW2	Status Bytes

Table 24 – READ BINARY Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’69 82’	Security status not satisfied
’69 85’	Condition of use not satisfied (File not activated)
’69 86’	Command not allowed (no current EF)
‘6B 00’	Wrong parameter P1-P2 (offset outside the EF)
‘6C’ XX	***Le*** incorrect, ***XX*** indicates the expected length (hexadecimal value)
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 25 – READ BINARY Status bytes

5.8 UPDATE BINARY COMMAND (ISO 7816-4)

5.8.1 Description

This command is used to update the content of a transparent EF.

The transparent EF must be currently selected and activated.

The security conditions to fulfil before performing this command depend on the current selected file.

5.8.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
UPDATE BINARY	'00'	'D6'	OFF_H	OFF_L	Length	-	3

Command ADPU

Field	Value
CLA	'00'
INS	'D6'
P1	OFF_H: Higher byte of the offset (bit 8 =0)
P2	OFF_L: Lower byte of the offset
Lc	Length of the subsequent data field
Data	Data to be updated
Le	Empty

Table 26 – UPDATE BINARY Command APDU

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 27 – UPDATE BINARY Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’67 00’	Wrong length (Lc=0, P1||P2 + Lc outside the EF)
’69 82’	Security status not satisfied
’69 85’	Condition of use not satisfied (File not activated)
’69 86’	Command not allowed (no current EF)
‘6B 00’	Wrong parameter P1-P2 (offset outside the EF)
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 28 – UPDATE BINARY Status bytes

5.9 ERASE BINARY COMMAND (ISO 7816-4)

5.9.1 Description

This command is used to erase the content of a transparent EF to '00' starting from a given offset up to the end of the file.

The transparent EF must be currently selected and activated.

The content of the transparent EF is physically overwritten to ensure that the data cannot be retrieved.

The security conditions to fulfil before performing this command depend on the current selected file.

5.9.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
ERASE BINARY	'00'	'0E'	OFF_H	OFF_L	-	-	1

Command ADPU

Field	Value
CLA	'00'
INS	'0E'
P1	OFF_H: Higher byte of the offset (bit 8 =0)
P2	OFF_L: Lower byte of the offset
Lc	Empty
Data	Empty
Le	Empty

Table 29 – ERASE BINARY Command APDU

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 30 – ERASE BINARY Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’69 82’	Security status not satisfied
’69 85’	Condition of use not satisfied (File not activated)
’69 86’	Command not allowed (no current EF)
‘6B 00’	Wrong parameter P1-P2 (offset outside the EF)
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 31 – ERASE BINARY Status bytes

5.10 MVP: VERIFY COMMAND (ISO 7816-4)

5.10.1 Description

This command is used to fulfil a PIN access right. This command is usually defined as a “PIN verification” procedure.

See Figure 1 - PIN verification process.

This command is performed atomically.

The previous access condition on this PIN[3] is discarded whatever the authentication result is.

The $PIN_{permanent}$ status is kept when selecting another application and coming back to the current one.

5.10.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
MVP: VERIFY	‘00’	‘20’	‘00’	Data Reference	‘08’	-	3

Command ADPU

Field	Value
CLA	‘00’
INS	‘20’
P1	‘00’
P2	Data reference
Lc	‘08’ : Length of verification data
Data	Verification data
Le	Empty

Table 32 – MVP: VERIFY Command APDU

[3] All other PIN still keep their status

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 33 – MVP: VERIFY Response APDU

Status bytes

Value	Meaning
’63 C**x**’	Verification failed, ‘**x**’ retries remaining
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’69 83’	Authentication method blocked (PIN counter null)
’69 85’	Condition of use not satisfied (e.g. Key reference does not point on a PIN)
‘6A 86’	Wrong parameter P1-P2
‘6A 88’	Referenced PIN not found
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 34 – MVP: VERIFY Status bytes

5.11 MVP: CHANGE REFERENCE DATA COMMAND (ISO 7816-8)

5.11.1 Description

This command is used to replace an existing PIN value with a new one

The new value is presented with the same format, as it exists within the card.

This command is used in two different ways:

1. The user changes his PIN value.
   In that case, the previous access condition is discarded; the current PIN is presented and compared with the one stored in the EID card.
   If the comparison fails, the PIN retry counter is decreased and the PIN value not changed.
   Otherwise if the verification is successful, the PIN value is modified with the new PIN value and the associated access right granted.

2. The administrator presets the PIN to a random value and resets the PIN try counter.
   The new PIN must have the format defined in §3.3.
   In that case an administrator access right has to be granted (i.e., the MVP: Verify (PIN_{reset}) APDU command has to be previously executed). The PIN access right is not granted.

This command is usually defined as a “PIN updating” procedure.

The PIN update is performed atomically.

5.11.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
MVP: CHANGE REFERENCE DATA	‘00’	‘24’	‘00’	Ref Data	’10’	-	3

Command ADPU

Field	Value
CLA	‘00’
INS	‘24’
P1	1. ‘00’ (User)
P2	Data reference (PIN reference)
Lc	Length of subsequent data field ‘10’
Data	1. Existing PIN || New PIN
Le	Empty

Table 35 – MVP: CHANGE REFERENCE DATA Command APDU

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 36 – MVP: CHANGE REFERENCE DATA Response APDU

Status bytes

Value	Meaning
’63 CX’	Verification failed, ‘***X***’ retries remaining
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’67 00’	Wrong length
’69 82’	Security status not satisfied (PIN_{reset} not granted)
’69 83’	Authentication method blocked (PIN blocked)
’69 85’	Condition of use not satisfied (e.g. Key reference does not point on a PIN)
‘6A 80’	Incorrect parameter in data field (e.g. wrong PIN format)
‘6A 86’	Wrong parameter P1-P2
‘6A 88’	Referenced data not found (PIN not found)
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 37 – MVP: CHANGE REFERENCE DATA Status bytes

5.12 GET CHALLENGE COMMAND (ISO 7816-4)

5.12.1 Description

This command is used to generate a random value from the card.

No security conditions are required to perform this command.

5.12.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
Get Challenge	'00'	'84'	'00'	'00	-	Length	2

Command ADPU

Field	Value
CLA	'00'
INS	'84'
P1	'00'
P2	'00
Lc	Empty
Data	Empty
Le	Length of the required random value

Table 38 – GET CHALLENGE Command APDU

Note: If ***Le*** is equal to 0, then it is interpreted as 256 bytes.

Response APDU

Field	Value
Data	Random value
SW1-SW2	Status Bytes

Table 39 – GET CHALLENGE Response APDU

Status bytes

Value	Meaning
'64 00'	No precise diagnostic
'65 81'	EEPROM corrupted (followed by a temporary mute state)
'6A 86'	Wrong parameter P1-P2
'6D 00'	Command not available within the current life cycle
'6E 00'	CLA not supported
'90 00'	Normal ending of the command

Table 40 – GET CHALLENGE Status bytes

5.13 EXTERNAL AUTHENTICATE COMMAND (ISO 7816-4)

5.13.1 Description

The EID card uses this command to authenticate the external application.

Two external authentication mechanisms can be performed:

1. External authentication without certificate (refer to 0) that grants the EXA access right.

This command uses the command chaining process.

All activated RSA 2048 public keys can be used by this command.

The access right of the referenced key related to this command must be fulfilled prior using the command.

The previous access condition is discarded whatever the authentication result is. (EXA access conditions are lost).

A ***Get Challenge*** APDU command must be executed just before performing the External Authenticate command.

5.13.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
EXTERNAL AUTHENTICATE	'00' or '10'	'82'	Algorithm	Key ref.	Length	-	3

Command ADPU

Field	Value
CLA	'00' or '10' (chained command)
INS	'82'
P1	Algorithm reference = '01' (refer to Table 17): Only RSASSA-PSS SHA-1 is supported
P2	Key reference 1. Public Key reference
Lc	Refer to command chaining paragraph
Data	1. '9E' || Signature length = ' 82 0100' || Signature ($RES_{application}$)
Le	Empty

Table 41 – EXTERNAL AUTHENTICATE Command APDU

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 42 – EXTERNAL AUTHENTICATE Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’67 00’	Wrong length
’69 82’	Security Status not satisfied (e.g. wrong certificate, wrong signature, PIN access right not granted)
’69 85’	Condition of use not satisfied (e.g. no previous Get Challenge, P2 doesn’t refer to a public key, key deactivated, key not initialised, etc.)
‘6A80’	Incorrect parameter in data field (e.g. wrong certificate format)
‘6A 86’	Wrong parameter P1-P2
‘6A 88’	Referenced key not found
‘6A 88’	Referenced key is not a RSA 2048 public key
‘6D 00’	Command not available within the current life cycle
‘6E 00’	CLA not supported
’90 00’	Normal ending of the command

Table 43 – EXTERNAL AUTHENTICATE Status bytes

Compatibility note

This command changed from application V1.

5.14 MSE: SET COMMAND (ISO 7816-8)

5.14.1 Description

The MSE: SET command is used to set attributes in the current Security Environment.

- Selecting the algorithm used to perform a signature/decryption via the algorithm reference (see 0 “
- ***Coding of the algorithm*** references”)
- Selecting the RSA Private Key (by using a key reference inside the **D**igital **S**ignature **T**emplate or **C**onfidentiality **T**emplate) that is used in the digital signature creation or decipher process.

No security conditions are required to perform this command.

5.14.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
MSE: SET	‘00’	‘22’	‘41’	‘B6’ / ‘B8’	‘05’	-	3

Command ADPU

Field	Value
CLA	‘00’
INS	‘22’: Manage Security Environment
P1	‘41’: Set the signature mode
P2	‘B6’ / ‘B8’: Value of the DST / CT in data field
Lc	‘05’ : Length of subsequent data field
Data	Length of following data =’04’ || Tag for Algorithm reference =’80’ || Algorithm reference = refer to Table 17 || Tag for private key reference =’84’ || Private key reference
Le	Empty

Table 44 – MSE: SET command APDU

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 45 – MSE: SET Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a temporary mute state)
’67 00’	Wrong length
’69 85’	Condition of use not satisfied (P2 doesn’t refer to a private key, P2 does not refer to a correct pair: key/algorithm, key deactivated, key not initialised, key reference incompatible with algorithm reference)
‘6A 80’	Incorrect parameter in the data field (e.g. Wrong tag, Wrong algorithm reference).
‘6B 00’	Wrong parameter P1-P2
‘6A 88’	Referenced key not found
‘6D 00’	Command not available within the current life cycle
’6E 00’	Wrong Class byte
’90 00’	Normal ending of the command
’94 84’	Algorithm not supported

Table 46 – MSE: SET Status bytes

Compatibility note

Only MSE SET for Digital Signature Template was supported in application V1.

5.15 PSO: COMPUTE DIGITAL SIGNATURE COMMAND (ISO 7816-8)

5.15.1 Description

The PSO: Compute Digital Signature command initiates the computation of a digital signature. The private key and the algorithm to be used has been previously specified by a MSE: SET command.

The access right of the referenced key related to this command must be fulfilled prior using the command (refer to Table 3).
"Signature keys" only apply to signature algorithms.

Only BelPIC application is allowed to use the PSO: Compute Digital Signature command.

5.15.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
PSO: COMPUTE DIGITAL SIGNATURE	'00'	'2A'	'9E'	'9A'	Length	Length	4

Command ADPU

Field	Value
CLA	'00'
INS	'2A': Perform Security Operation
P1	'9E': PSO: Compute Digital Signature
P2	'9A' (Data field contains data to be signed)
Lc	Length of the data to be signed
Data	Data to be signed (not padded)
Le	Length of the signature – always '80' (128 bytes)

Table 47 – PSO: COMPUTE DIGITAL SIGNATURE command APDU

Note: When PKCS#1 is used, the card pads the DTBS according to PKCS#1 version 2.1.

Algorithms to be used inside PSO: Compute Digital Signature:

See 0 "

Coding of the algorithm references":

- RSASSA-PKCS1-v1.5 using SHA1 algorithm selected:
  The length of the DTBS must be equal to 20 bytes.
- RSASSA-PKCS1-v1.5 using MD5 algorithm selected:
  The length of the DTBS must be equal to 16 bytes.
- RSASSA-PSS using SHA1 (defined in **[ER12]**) selected:
  The length of the DTBS must be equal to 20 bytes as a result of a SHA1.

.

Response APDU

Field	Value
Data	Signature
SW1-SW2	Status Bytes

Table 48 – PSO: COMPUTE DIGITAL SIGNATURE Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a mute state)
’67 00’	Wrong length
’69 82’	Security status not satisfied (e.g. PIN access right not granted)
’69 85’	Condition of use not satisfied (e.g. security environment not set or BelPIC application not selected, security environment incompatible with the command, key not activated, key not initialised)
‘6B 00’	Wrong parameter P1-P2
‘6D 00’	Command not available within the current life cycle
’6E 00’	Wrong Class byte
’90 00’	Normal ending of the command

Table 49 – PSO: COMPUTE DIGITAL SIGNATURE Status bytes

Compatibility note

RSASSA-PSS was not available in application V1.

5.16 PSO: DECIPHER COMMAND (ISO 7816-8)

5.16.1 Description

The PSO: Decipher command uses a private key to decipher and output the enciphered message. The private key and the algorithm to be used has been previously specified by a MSE: SET command.

The access right of the referenced key related to this command must be fulfilled prior using the command (refer to Table 3).
"Decipher" keys only apply to ciphering algorithms.

5.16.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
PSO: DECIPHER	'00'	'2A'	'80'	'86'	'81'	Length	4

Command ADPU

Field	Value
CLA	'00'
INS	'2A': Perform Security Operation
P1	'80': decrypted value is returned in response
P2	'86': data field contains padding indicator byte (00h according to ISO/IEC 7816-4) followed by the cryptogram
Lc	Length of the data to be deciphered – always '81' (129 bytes)
Data	'00' + Data to be deciphered
Le	Length of the message

Table 50 – PSO: DECIPHER command APDU

Algorithms to be used inside PSO: Decipher:

See 0 "

Coding of the algorithm references":

- RSAES-PKCS1-v1.5 algorithm selected (no padding):
  The length of the data to be deciphered must be equal to 128 bytes.
- RSAES-OAEP using SHA1 algorithm selected (no padding):
  The length of the data to be deciphered must be equal to 128 bytes.

Response APDU

Field	Value
Data	Signature
SW1-SW2	Status Bytes

Table 51 – PSO: DECIPHER Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic – Error in expected padding format after deciphering.
‘65 81’	EEPROM corrupted (followed by a mute state)
’67 00’	Wrong length
’69 82’	Security status not satisfied (e.g. PIN access right not granted)
’69 85’	Condition of use not satisfied (e.g. security environment not set, security environment incompatible with the command , key not activated, key not initialised)
’6A 80’	Incorrect parameter in data field (e.g. incorrect padding indicator)
‘6B 00’	Wrong parameter P1-P2
‘6D 00’	Command not available within the current life cycle
’6E 00’	Wrong Class byte
’90 00’	Normal ending of the command

Table 52 – PSO: DECIPHER Status bytes

Compatibility note

This command was not available in application V1.

5.17 GET CARD DATA COMMAND

5.17.1 Description

This command is used to retrieve the some useful information about the card and the current application.

No security conditions are required to perform this command.

According to P1 parameters, this command is able to sign the output result with the local application base key (reference '**81**') (algorithm RSASSA-PKCS1 v1.5 SHA-1).

5.17.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
GET CARD DATA	'80'	'E4'	'00' / '02'	'00'	-	Length	2

Command ADPU

Field	Value
CLA	'80'
INS	'E4'
P1	'00' result not signed '02' result signed
P2	'00'
Lc	Empty
Data	Empty
Le	Length of the response ('1C' not signed or '9C' signed)

Table 53 - GET CARD DATA Command APDU

Response APDU

Field	Value
Data	Serial Number (16 bytes) [4] || Component code (1 byte) || OS number (1 byte) || OS version (1 byte) Softmask number (1 byte) || Softmask version (1 byte) || Application version (1 byte) || Global OS version (2 byte) [5] || Application interface version (1 byte) || PKCS#1 support (1 byte) refer to Table 55|| Key exchange version (1 byte) || Application Life cycle (1 byte) refer to Table 56 || [Signature (RSASSA-PKCS1 v1.5 SHA-1)] optional : if signed result is required
SW1-SW2	Status Bytes

Table 54 – GET CARD DATA Response APDU

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Meaning
-	-	-	-	-	-	-	1	RSASSA-PKCS1 v1.5 supported (MD5 and SHA-1)
-	-	-	-	-	-	1	-	RSASSA-PSS supported (SHA-1)
-	-	-	-	-	1	-	-	RSAES-PKCS1 v1.5 supported
-	-	-	-	1	-	-	-	RSAES-OAEP supported
-	-	-	1	-	-	-	-	RSA-KEM supported
0	0	0	-	-	-	-	-	Other values are RFU

Table 55 – PKCS#1 support[6]

Application Life cycle value	Meaning
'0F'	DEACTIVATED state
'8A'	ACTIVATED state
'FF'	LOCKED state

Table 56 – Application Life cycle values

[4] The serial number is composed of 2 bytes reserved for axalto, 2 bytes identifying the chip manufacturer, and 12 bytes identifying uniquely the chip inside all chips from this manufacturer.
[5] This global number is unique for a given set composed of: Component code || OS number || OS version || Softmask number || Softmask version || Application version
[6] The PKCS#1 version supported is set before personalisation phase.

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a mute state)
’67 00’	Wrong length
’69 85’	Condition of use not satisfied (e.g. key not activated, key not initialised, key reference is not a signature key)
6A 86’	Wrong parameter P1-P2
‘6A 88’	Key Reference not found
‘6D 00’	Command not available within the current life cycle
’6E 00’	Wrong Class byte
’90 00’	Normal ending of the command

Table 57 – GET CARD DATA Status bytes

Compatibility note

The signature was not available in application V1.

5.18 GET PIN STATUS COMMAND

5.18.1 Description

This command is used to retrieve some information about the referenced PIN.

No security conditions are required to perform this command.

According to P1 parameters, this command is able to sign the output result with the local application base key (reference '**81**') (algorithm RSASSA-PKCS1 v1.5 SHA-1).

5.18.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
GET PIN STATUS	'80'	'EA'	'00' / '02'	PIN ref.	-	Length	2

Command ADPU

Field	Value
CLA	'80'
INS	'EA'
P1	'00' result not signed '02' result signed
P2	Data reference (PIN reference)
Lc	Empty
Data	Empty
Le	Length of the response ('01' not signed or '81' signed)

Table 58 - GET PIN STATUS Command APDU

Response APDU

Field	Value
Data	Number of tries remaining (1 byte)|| [Signature (RSASSA-PKCS1 v1.5 SHA-1)] optional : if signed result is required
SW1-SW2	Status Bytes

Table 59 – GET PIN STATUS Response APDU

If number of tries remaining reaches the zero value, then the referenced PIN is considered as blocked.

Status bytes

Value	Meaning
'64 00'	No precise diagnostic
'65 81'	EEPROM corrupted (followed by a mute state)
'67 00'	Wrong length
'69 85'	Condition of use not satisfied (e.g. key not activated, key not initialised, key reference is not a signature key)
'6A 88'	Referenced data not found (e.g. PIN not found, signature key not found)
'6B 00'	Wrong parameter P1-P2
'6D 00'	Command not available within the current life cycle
'6E 00'	Wrong Class byte
'90 00'	Normal ending of the command

Table 60 – GET PIN STATUS Status bytes

Compatibility note

This command was not available in application V1.

5.19 LOG OFF COMMAND

5.19.1 Description

This command is used to discard the current fulfilled access condition (PIN and EXA in the current DF. In BelPIC DF, all global access conditions are also discarded.

The access conditions in other DF are not modified.

No security conditions are required to perform this command.

5.19.2 Command structure

Instruction	CLA	INS	P1	P2	Lc	Le	Case
LOG OFF	'80'	'E6'	'00'	'00'	-	-	1

Command ADPU

Field	Value
CLA	'80'
INS	'E6'
P1	'00'
P2	'00'
Lc	Empty
Data	Empty
Le	Empty

Table 61 - LOG OFF Command APDU

Response APDU

Field	Value
Data	Empty
SW1-SW2	Status Bytes

Table 62 – LOG OFF Response APDU

Status bytes

Value	Meaning
’64 00’	No precise diagnostic
‘65 81’	EEPROM corrupted (followed by a mute state)
’67 00’	Wrong length
‘6A86’	Wrong parameter P1-P2
‘6D 00’	Command not available within the current life cycle
’6E 00’	Wrong Class byte
’90 00’	Normal ending of the command

Table 63 – LOG OFF Status bytes

MODIFICATION SHEET

Date	Version	Modifications
16-06-04	1.0	Initial user public version - extracted from MRDSTG023089 v2.0y “General Technical Specification BelPIC v2.0”

End of document