GLOBALPLATFORM®

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

GlobalPlatform Card Technology

Card Specification – ISO Framework

Version 1.0

Public Release

March 2014

Document Reference: GPC_SPE_055

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Figures

Tables

1 Introduction

This document proposes a framework based on the GlobalPlatform Card Specification [GPCS] that is compliant with ISO specifications ISO/IEC 7816-13 ([7816-13]), ISO/IEC 7816-4 ([7816-4]), and ISO/IEC 24727-2 ([24727-2]). In this specification, card content operations are protected by the Secure Channel Protocol '03' as defined in GPCS Amendment D [Amd D]. Further precisions on the usage of this protocol are given in this document.

This specification describes the ISO Security Domain (ISO-SD), which is a specific implementation of Security Domain that is compatible with the latest ISO specifications. In particular, an ISO-SD shall support an alternative set of commands, compatible with [7816-13], giving access to card content management, Secure Channel initiation, and personalization functions.

1.1 Audience

This document is intended primarily for card manufacturers and application developers developing GlobalPlatform card implementations.

It is assumed that the reader is familiar with smart cards and smart card production, and in particular familiar with [GPCS].

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of IPR held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://www.globalplatform.org/specificationsipdisclaimers.asp. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

Table 1-1: Normative References

Standard / Specification	Description	Ref
GlobalPlatform Card Specification	GlobalPlatform Card Specification v2.2.1, January 2011	[GPCS]
GPCS Amendment D	Secure Channel Protocol 03 – GlobalPlatform Card Specification v 2.2 – Amendment D, v1.1, September 2009	[Amd D]
ISO/IEC 7816-3: 2006	Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols	[7816-3]
ISO/IEC 7816-4: 2013	Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange	[7816-4]
ISO/IEC 7816-6: 2004	Identification cards – Integrated circuit cards – Part 6: Interindustry data elements for interchange	[7816-6]
ISO/IEC 7816-13: 2007	Identification cards – Integrated circuit cards – Part 13: Commands for application management in a multi-application environment	[7816-13]

Standard / Specification	Description	Ref
ISO/IEC 8825-1: 2008	Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)	[8825-1]
ISO/IEC 24727-2: 2008	Identification cards – Integrated circuit card programming interfaces – Part 2: Generic card interface	[24727-2]
GICS part 1 v0.90 and part 2 v0.23	Generic Identity Command Set (Specification from INCITS B10.12 part 1 & part 2)	[GICS]
FIPS PUB 186-3	Digital Signature Standard (DSS)	[FIPS]
NIST SP 800-38B	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication	[SP800-38b]
RFC 5639	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation	[5639]

1.4 Terminology and Definitions

Selected technical terms used in this document are included in Table 1-2. Additional technical terms are defined in [GPCS].

Table 1-2: Terminology and Definitions

Term	Definition
Application Dedicated File (ADF)	A structure hosting a Control Parameter Template and an application. The application could implement the Generic Identity Command Set specification [GICS].
Reference Data Qualifier	Term used in [7816-4] for the Object Identifier of a Security Object (that is, a Key Identifier). In this specification, the Reference Data Qualifier is the second byte of the Security Object Number.

1.5 Abbreviations and Notations

Hexadecimal values are enclosed in straight single quotation marks (example: '0F').

Selected abbreviations used in this document are included in Table 1-3. Additional abbreviations are defined in [GPCS].

Table 1-3: Abbreviations and Notations

Abbreviation / Notation	Meaning
ACD	Application Capability Descriptor
ADF	Application Dedicated File
AES	Advanced Encryption Standard
AID	Application Identifier *AID* is used in Java Card specifications and most GlobalPlatform specifications. *Application Dedicated File Name*, used in this specification and ISO specifications, is equivalent.
AM	Authorized Management (GlobalPlatform privilege)
AMB	Access Mode Byte
AMF	Access Mode Field
AMR	APPLICATION MANAGEMENT REQUEST command
APDU	Application Protocol Data Unit
BER-TLV	Basic Encoding Rules – Tag Length Value
C_DECRYPTION	The Secure Channel’s Security Level indicator value specifying that the data field of each APDU command is required to be encrypted.
C_MAC	The Secure Channel’s Security Level indicator value specifying that a MAC is required for each APDU command.
CBC	Cipher Block Chaining
CCD	Card Capability Descriptor
CIN	Card Identification Number
CLA	CLAss Byte
CLF	ContactLess Frontend
CP	Control Parameters
CRT	Control Reference Template
DF	Dedicated File
DO	Data Object
ECC	Elliptic Curve Cryptography
EF	Elementary File
ELF	Elementary Load File

Abbreviation / Notation	Meaning
FCI	File Control Information
FMD	File Management Data
GICS	Generic Identity Command Set, as defined in GICS part 1 and part 2 ([GICS])
IIN	Issuer Identification Number
INCITS	InterNational Committee for Information Technology Standards
INS	INStruction Byte
ISD	Issuer Security Domain
ISO	International Organization for Standardization
ISO-SD	ISO Security Domain
KDF	Key Derivation Function
Lc	Exact length of command data in a case 3 or case 4 command
Le	Maximum length of data expected in response to a case 2 or case 4 command
Len	Length
MAC	Message Authentication Code
MF	Master File
N/A	Not Applicable
OID	Object IDentifier
R_ENCRYPTION	Response Encryption
R_MAC	Response MAC
SCB	Security Condition Byte
SCP	Secure Channel Protocol
SE	Security Environment
SEID	Security Environment ID
SIM	Subscriber Identity Module
SM	Secure Messaging
SPT	Security Parameter Template
SSD	Supplementary Security Domain
SWP/HCI	Single Wire Protocol/Host Controller Interface
S-ENC	Secure Channel command and response encryption key
S-MAC	Secure Channel C-MAC session key
S-RMAC	Secure Channel R-MAC session key
T=0	Character-oriented asynchronous half duplex transmission protocol
T=1	Block-oriented asynchronous half duplex transmission protocol

Abbreviation / Notation	Meaning
T=CL	Transmission protocol defined in ISO 14443-4 for contactless cards
TLV	Tag Length Value
USB	Universal Serial Bus
Var	Variable

1.6 Revision History

Table 1-4: Revision History

Date	Version	Description
March 2014	1.0	Initial release

2 General Definitions

2.1 Class Byte (CLA)

In order to support the ISO command chaining mechanism, bit 5 of the CLASS byte (CLA) is defined according to Table 2-1.

Table 2-1: CLA Byte Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	–	0	–	–	x(*)	x(*)	Command is last or only command of a chain
0	0	–	1	–	–	x(*)	x(*)	Command is not the last command of a chain
0	0	–	–	0	0	x(*)	x(*)	No secure messaging
0	0	–	–	1	1	x(*)	x(*)	Secure messaging with authenticated command header (See [7816-4] § 5.4.1.)

(*) Bit 2 and bit 1 define the logical channel used to transmit the command to an application.

2.2 Instruction Byte (INS)

As defined in [7816-4] § 5.5, in the inter-industry class, bit 1 of INS indicates a data field format as follows:

- If bit 1 is set to 0 (even INS code), then no indication is provided.
- If bit 1 is set to 1 (odd INS code), then BER-TLV encoding shall apply.

2.3 Cryptographic Mechanism Reference Values

The reference values listed in Table 2-2 are used in the Security Object Number, the GENERAL AUTHENTICATE command, and Control Parameters of keys and ADFs.

All cryptographic mechanisms are not mandatory and some may be not supported by the card.

Table 2-2: Cryptographic Mechanism Reference

Reference Value	Cryptographic Algorithm Mode
'00'	3 Key Triple DES – ECB
'01'	2 Key Triple DES – ECB
'03'	3 Key Triple DES – ECB
'04'	3 Key Triple DES – CBC
'06'	RSA 1024 bit modulus, 65,537 ≤ exponent ≤ 2^{864} – 1
'07'	RSA 2048 bit modulus, 65,537≤ exponent ≤ 2^{1824} – 1
'08' or '09'	AES-128
'0A' or '0B'	AES-192
'0C' or '0D'	AES-256
'0E'	ECC: Curve P-224 as defined in [FIPS]
'11'	ECC: Curve P-256 as defined in [FIPS]
'14'	ECC: Curve P-384 as defined in [FIPS]
'15'	ECC: Curve P-521 as defined in [FIPS]
'20'	Key Establishment using Symmetric Key (internal authenticate)
'21'	Key Establishment using Symmetric Key (mutual authenticate)
'22'	Key Establishment using Symmetric Key (mutual authenticate and data integrity)
'23'	Key Establishment using Symmetric Key based on SCP '03'
'24'	RSA Key Transport
'25'	Key Establishment using RSA Key Pair
'26'	Key Establishment using an ECC Key Pair, Diffie-Hellman
'27'	Opacity with Zero Key Management
'28'	Key Establishment using Opacity with Full Secrecy
'29'	Opacity with Zero Key Management – Fast
'2A'	Opacity with Full Secrecy – Strong Key Transport
'2B'	Opacity with Full Secrecy – Strong Security
'2C'	Opacity with Full Secrecy – Very Strong Key Transport
'2D'	Opacity with Full Secrecy – Very Strong Security

2.4 Security Object Number

A Security Object Number is used to reference a Security Object. In this specification, Security Objects are intended to represent a key.

A Security Object Number is composed of two bytes:

- The first byte includes the reference value (from Table 2-2) that identifies the Key Algorithm.
- The second byte corresponds to the Reference Data Qualifier that identifies the Security Object itself.

The Security Object Number is contained in tag '83' that belongs to the Security Parameter Template (tag 'AD').

2.5 Dynamic Authentication Data Objects (Tag '7C')

The Dynamic Authentication Data Objects template is used in the GENERAL AUTHENTICATE command data field.

Table 2-3: Dynamic Authentication Data Objects (Tag '7C')

Tag	Length	Value
'7C'	Var	Template that includes information such as key input information, host challenge, card challenge, response cryptogram, etc.

2.6 Security Level Encoding

The Security Level Encoding is used in the template Key Input Information in the GENERAL AUTHENTICATE APDU command. It is used to define the security level requested by the host during establishment of the secure messaging session.

The security level for the secure channel protocol shall be coded as a bitmap on one byte, as defined in Table 2-4.

Table 2-4: Security Level Encoding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	ANY AUTHENTICATED (no secure messaging expected)
1								AUTHENTICATED (no secure messaging expected)
0	0	0	0	0	0	0	1	C_MAC
0	0	0	0	0	0	1	1	C_DECRYPTION & C_MAC
0	0	0	1	0	0	0	1	R_MAC & C_MAC
0	0	0	1	0	0	1	1	C_DECRYPTION & C_MAC & R_MAC
0	0	1	1	0	0	1	1	R_ENCRYPTION & R_MAC & C_MAC & C_DECRYPTION

2.7 Secure Channel Protocol Option

The Secure Channel Protocol Option is used in the template Key Output Information in the GENERAL AUTHENTICATE APDU response. It is used to indicate to the host, during establishment of the secure messaging session, the features supported by the card.

In Secure Channel Protocol '03' the “i” parameter shall be coded as a bitmap on one byte, as defined in Table 2-5.

Table 2-5: Secure Channel Protocol Option

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	–	–	0	0	0	0	0	Random card challenge
0	–	–	1	0	0	0	0	Pseudo-random card challenge
0	0	0	–	0	0	0	0	No R_MAC/R_ENCRYPTION support
0	0	1	–	0	0	0	0	R_MAC support / no R_ENCRYPTION support
0	1	1	–	0	0	0	0	R_MAC/R_ENCRYPTION support

The Secure Channel Protocol Option (SCP Option) supported by the ISO-SD is defined during installation of the ISO-SD.

2.8 Minimum Security Level Encoding

The Minimum Security Level is used in the template Key Output Information in the GENERAL AUTHENTICATE APDU command. It is used to define the minimum security level requested by the card during establishment of the secure messaging session.

This parameter shall be coded as a bitmap on one byte, as defined in Table 2-6.

Table 2-6: Minimum Security Level Encoding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	–	–	0	0	–	1	C_MAC is required
0	0	–	–	0	0	1	–	C_DECRYPTION is required
0	0	–	1	0	0	–	–	R_MAC is required
0	0	1	–	0	0	–	–	R_ENCRYPTION is required
0	0	0	0	0	0	0	0	No Secure Messaging required

2.9 Card/Host Challenge (Tag '81')

The Card/Host Challenge is used in the template Dynamic Authentication Data Objects in the GENERAL AUTHENTICATE APDU command.

A Card/Host Challenge is one or more random or pseudo random numbers or byte sequences to be used in the GENERAL AUTHENTICATE APDU command during establishment of the secure messaging session.

The Card/Host Challenge data shall be formatted as defined in Table 2-7.

Table 2-7: Card/Host Challenge TLV (Tag '81')

Tag	Length	Value
'81'	Var	Card/Host Challenge

2.10 Response Cryptogram (Tag '82')

A response cryptogram is a sequence of bytes encoding a response step in an authentication protocol.

A response cryptogram is used in the GENERAL AUTHENTICATE APDU command during establishment of the secure messaging session.

The response cryptogram is formatted as defined in Table 2-8.

Table 2-8: Response Cryptogram TLV (Tag '82')

Tag	Length	Value
'82'	Var	Response cryptogram

2.11 Key Input Information (Tag '88')

The Key Input Information is used in the GENERAL AUTHENTICATE APDU command during establishment of the secure messaging session. It defines the Security Objects and the security level that shall be used for establishment of the secure messaging session.

Key Input Information is formatted as defined in Table 2-9.

Table 2-9: Key Input Information TLV (Tag '88')

Tag	Length	Value
'88'	3	Bytes 1-2: Key File ID that corresponds to the Static MAC as defined in P1/P2 of the INITIALIZE UPDATE APDU command for SCP '03' Byte 3: The security level requested by the host, encoded as described in Table 2-4

2.12 Key Output Information (Tag '88')

The Key Output Information is used in the GENERAL AUTHENTICATE APDU command during establishment of the secure messaging session. It defines the Secure Channel Protocol and the minimum security level required by the card.

Key Output Information is formatted as defined in Table 2-10.

Table 2-10: Key Output Information TLV (Tag '88')

Tag	Length	Value	Description
'88'	2 or 3	'03'	Byte 1: Secure Channel Protocol
			Byte 2: SCP Option, as defined in Table 2-5
			Byte 3: Optional: The minimum security level required by the card, as defined in Table 2-6 If not defined, this value is assumed to be '00', meaning that no secure messaging is expected.

2.13 Sequence Counter (Tag '89')

The Sequence Counter (if present) is used in the template Dynamic Authentication Data Objects in the GENERAL AUTHENTICATE APDU command.

If defines the Sequence Counter used for Pseudo Random computation.

The Sequence Counter for Secure Channel Protocol '03' is formatted as defined in Table 2-11.

Table 2-11: Sequence Counter TLV (Tag '89')

Tag	Length	Value
'89'	3	Sequence Counter

2.14 Physical Interface Encoding (Tag '91')

The Physical Interface Encoding is used in the Security Attribute Template.

It defines the physical interface to which the access conditions shall apply.

The physical interface shall be coded as a bitmap on one byte, as defined in Table 2-12.

Table 2-12: Physical Interface Encoding (Tag '91')

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
X	X	X	X	–	–	–	–	RFU
–	–	–	–	–	1	–	–	USB Interface
–	–	–	–	–	–	1	–	Contactless Interface[1]
–	–	–	–	–	–	–	1	Contact Interface

2.15 Security Parameter Template (Tag 'AD') for Application Dedicated File (ADF)

The Security Parameter Template (tag 'AD') is used inside the Application Dedicated File (ADF) Control Parameter Data Object (tag '62') to define key storage.

When tag '8C' is used in Security Attribute Template, tag 'AD' is not used and shall not be present.

Table 2-13: Security Parameter Template (Tag 'AD') for ADF

<table>
<tr><th>Tag</th><th>Length</th><th colspan="3">Description</th></tr>
<tr><td rowspan="4">'AD'</td><td rowspan="4">7</td><td colspan="3">Security Parameter Template</td></tr>
<tr><th>Tag</th><th>Len</th><th>Description</th></tr>
<tr><td>'80'</td><td>1</td><td>Sequence Number. This value will be used in Security Condition Byte (SCB).</td></tr>
<tr><td>'83'</td><td>2</td><td>Security Object Number for key, as described in section 2.4</td></tr>
</table>

2.16 Card Diversification Data (Tag '85')

The Card Diversification Data is formatted as defined in Table 2-14.

Table 2-14: Card Diversification Data TLV (Tag '85')

Tag	Length	Value
'85'	10	Card Diversification Data

The Card Diversification Data is data typically used by a backend system to derive the card static keys.

[1] Contactless Interface corresponds to communication with a Contactless integrated circuit card compliant with ISO/IEC 14443.

2.17 Card Data Template (Tag '66')

The Card Data Template is a collection of Data Objects identifying the card, its state, and its applications for the purpose of managing or synchronizing the card from a Card Management System. (See [7816-6] table 10.)

Table 2-15: Card Data Template (Tag '66')

Tag	Length	Description	Presence
'66'	Var	Card Data Template: the content of this template is issuer dependent	Mandatory

2.18 Security Attribute (Tag 'A3')

2.18.1 Description

The Security Attributes are defined inside the Control Parameters (CP). They define access rules applying to the actions (APDU commands) supported by the object for one or more physical interfaces.

The Security Attribute Template (see Table 2-16) shall be used to specify Security Attributes for one or all physical interfaces. To specify different Security Attributes for different physical interfaces, the Security Attribute Template may contain several Physical Interface Type tags, each with an associated Security Attribute tag.

Template 'A3' shall contain at most one data object '91'.

- If template 'A3' contains a data object '91':
  - It shall be the first DO in the template.
  - All occurrences of data object '9C' or '8C' shall apply to interfaces identified in the data object '91'.
- If data object '91' is absent, then the security attributes contained in data object 'A3' apply to all interfaces.

Table 2-16: Security Attribute Template (Tag 'A3')

Tag	Length	Description			Presence
'A3'	Var	Security Attribute Template			Optional
		Tag	**Len**	**Description**	**Presence**
		'91'	1	Physical Interface Type	Optional
		'9C'	Var	Security Attribute in Compact Format, SPT oriented (see section 2.18.2)	Conditional – required for each occurrence of tag '91'. Tag '9C' and tag '8C' are mutually exclusive.
		'8C'	Var	Security Attribute in Compact Format, SE oriented	
		'5C'	0	This empty tag (L=0) indicates that the security rules shall apply to all Data Objects	Mandatory
'A3'	Var	Security Attribute Template			Optional
		'91'	1	Physical Interface Type	Optional
		'9C'	Var	Security Attribute in Compact Format, SPT oriented (see section 2.18.2)	Conditional – required for each occurrence of tag '91'. Tag '9C' and tag '8C' are mutually exclusive.
		'8C'	Var	Security Attribute in Compact Format, SE oriented	
		'5C'	0	Security rules shall apply to all Data Objects	Mandatory

2.18.2 Security Attribute in Compact Format (Tag '9C')

The Security Attribute in Compact Format allows specification of one or more access rules. Each access rule is composed of an Access Mode Field (AMF) that specifies the commands to which the access rule applies, followed by one or more Security Condition Bytes (SCB). The SCB provides information on the conditions under which a command can be executed. Exactly one SCB shall be present for each command specified in the AMF and SCBs shall appear in the same order as the command bits in the AMF (i.e. from most significant bit to least significant bit).

Several access rules (hence several SCBs) may be specified for the same command.

- If at least one of the SCBs specified for a specific command contains the “All Conditions” indicator, then the command shall be executed only if all the conditions specified for that command (by the different SCBs) have been satisfied (i.e. logical AND).
- Otherwise (i.e. the “All Conditions” indicator is never present for that command), then the command shall be executed if at least one of the conditions specified for that command (by the different SCBs) has been satisfied (i.e. logical OR).

Table 2-17: Security Attribute in Compact Format (Tag '9C')

Tag	Length	Description			Presence
'9C'	Var	Security Attribute in Compact Format			
		Access Rule #1	AMF#1	Access Mode Field #1 (see section 2.18.3)	Mandatory
			SCB	Concatenation of Security Condition Bytes (one per command specified in the AMF) (see section 2.18.4)	
			…		
			SCB		
		…	…	…	
		Access Rule #n	AMF#n	Access Mode Field #n (see section 2.18.3)	Optional
			SCB	Concatenation of Security Condition Bytes (one per command specified in the AMF) (see section 2.18.4)	
			…		
			SCB		

Note: Access conditions defined by SCBs shall be enforced as soon as the ADF enters the OPERATIONAL life cycle state.

2.18.3 Access Mode Field

The Access Mode Field (AMF) specifies one or more Access Mode Bytes (AMB). In this version of the specification, the AMF shall contain three bytes. The first byte of the AMF shall always be '00', indicating the presence of several AMBs in the AMF (i.e. second and third bytes). The encoding of the second and third bytes is described in the following tables.

Table 2-18: Access Mode Field (Second Byte)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	–	–	–	–	–	–	–	Another byte follows in the AMF
–	0	–	–	–	–	–	–	Bits b6 to b1 according to [7816-4] Table 22
–	–	0	–	–	0	0	–	RFU
–	–	–	1	–	–	–	–	ACTIVATE ADF
–	–	–	–	1	–	–	–	DEACTIVATE ADF
–	–	–	–	–	–	–	1	DELETE FILE

Table 2-19: Access Mode Field (Third Byte)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	–	–	–	–	–	–	–	Last byte of the AMF
	1							Bit 6 to bit 1 are proprietary[2]
		1						CREATE DATA
			1					DELETE DATA
				1				APPLICATION MANAGEMENT REQUEST
					1			REMOVE APPLICATION
						1		PUT DATA
							1	GET DATA

[2] This table is proprietary because of the CREATE DATA and DELETE DATA commands. These commands are under validation for a new version of ISO 7816-9.

2.18.4 Security Condition Byte

The Security Condition Byte (SCB) format is common to all files and is described in Table 2-20 ([7816-4] Table 30).

Table 2-20: Security Condition Byte

<table>
<tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th colspan="2">Meaning</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">No specific condition</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">The command shall be rejected</td></tr>
<tr><td>–</td><td>–</td><td>–</td><td>–</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">No reference to a Security Environment</td></tr>
<tr><td rowspan="2">–</td><td rowspan="2">–</td><td rowspan="2">–</td><td rowspan="2">–</td><td colspan="4" rowspan="2">Not all equal</td><td>Under tag '8C'</td><td>SEID</td></tr>
<tr><td>Under tag '9C'</td><td>Sequence Number of the Security Parameter Template 'AD' from 1 to 14</td></tr>
<tr><td>–</td><td>–</td><td>–</td><td>–</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">RFU</td></tr>
<tr><td>0</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td colspan="2">At least one condition</td></tr>
<tr><td>1</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td colspan="2">All conditions</td></tr>
<tr><td>–</td><td>1</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td colspan="2">Secure Messaging is required (at least)</td></tr>
<tr><td>–</td><td>–</td><td>1</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td colspan="2">External Authentication is required (at least)</td></tr>
<tr><td>–</td><td>–</td><td>–</td><td>1</td><td>–</td><td>–</td><td>–</td><td>–</td><td colspan="2">User authentication (i.e. Cardholder Verification) is required</td></tr>
</table>

Bits 8 to 5 indicate the required security conditions. If not all equal, bits 4 to 1 identify a Security Environment and the mechanisms defined in the Security Environment shall be used according to the indications in bits 7 to 5 for command protection and / or external authentication and / or user authentication.

- If bit 8 is set to 1, then all the conditions set in bits 7 to 5 shall be satisfied.
- If bit 8 is set to 0, then at least one of the conditions set in bits 7 to 5 shall be satisfied.

2.19 Security Environment (Tag '7B')

A Security Environment is an object created within the ISO-SD.

This object is used to reference one or more Security Object(s).

Table 2-21: Security Environment Template (Tag '7B')

Tag	Length	Description			Presence
'7B'	Var	Security Environment Template			Mandatory
		Tag	**Len**	**Description**	**Presence**
		'80'	1	Security Environment Identifier (from '01' to '0E')	Mandatory
		'A4'	Var	Authentication	Optional
		'A6'	Var	Key Agreement	Optional
		'AA'	Var	Hashing	Optional
		'B4'	Var	Cryptographic Checksum	Optional
		'B6'	Var	Digital signature	Optional
		'B8'	Var	Confidentiality	Optional

Note: Access conditions shall be enforced as soon as the ADF enters the OPERATIONAL life cycle state.

2.20 Security Objects

Security Objects can be created and maintained as autonomous objects within the ISO-SD, and they can be directly referenced through the card communication interfaces.

Security Objects define their own access rules and do not inherit access control rules of an object they may be nested in. These Security Objects could be placed anywhere in an application. Each of these objects shall have a Control Parameter Template (tag '62') which defines its characteristics and use.

A Security Object is an object that is used to store a cryptographic key. In addition, a Security Object can store a reference to another Security Object (see section 3.7). The purpose of Security Objects is to support access control and authentication. It shall not be possible for off-card entities to retrieve the content of Security Objects. In this version of the specification, Security Objects are used for Secure Channel Protocol '03', but their usage may be extended in a future version.

There shall be exactly one Security Attribute associated with each Security Object and this attribute shall be described using the Security Parameter Template (tag 'AD') within the Control Parameters.

Each Security Object shall be identified by a unique 2-byte Security Object Number (tag '83' within tag 'AD'). Moreover, the assigned value of each Security Object Number shall not conflict with any File Identifier value since Security Objects and files share the same namespace.

The Security Object is referenced in the GET DATA, PUT DATA, and GENERAL AUTHENTICATE commands.

The Security Object Security Parameter Template does not change when a key value is updated. However, if the new key requires different properties, the Security Object must first be deleted and a new key Security Object created with the correct attributes.

Table 2-22: Control Parameter Template for Security Objects (Tag '62')

Tag	Len							Description	Presence
'62'	Var							Control Parameter Template	Mandatory
		Tag	Len					Description	
		'A3'	Var					Security Attributes for DO (to set access control rule for PUT DATA and GET DATA at OPERATIONAL ACTIVATED STATE) – see Table 2-16.	Mandatory
		'AD'	Var					Security Parameter Template for Security Objects	Mandatory
				Tag	Len			Description	
				'82'	Var			Security Object (password/reference data type) number	Optionally supported
				'86'	Var			Key usage constraints indicator	Optionally supported
				'91'	Var			Maximum number of tries of an authentication procedure	Optionally supported
				'93'	Var			Remaining number of tries of an authentication procedure	Optionally supported
				'7B'	Var			A list of CRTs to further define the usage properties for the Security Object (See section 2.23.)	Optional
				'80'	1			Sequence Number between '01' and '0E' inclusive	Mandatory
				'83'	2			Security Object Number (Key Algorithm||Reference Data Qualifier) (See section 2.4)	Conditional
				'8A'	1			ISO Life Cycle State byte of the specific object (read only)	Conditional
				'95'	2			Maximum usage counter (binary coding)	Conditional
				'97'	2			Remaining usage counter (binary coding)	Conditional
				'A0'	Var			Security Attribute Extension for authentication objects	Conditional – mutually exclusive with Tags 'A1', 'A4', and 'A5'
						Tag	Len	Description	
						'A3'	Var	Security Attribute Template	Mandatory
				Tag	Len			Description	
				'A1'	Var			Security Attribute Extension for private keys	Conditional – mutually exclusive with Tags 'A0', 'A4', and 'A5'
						Tag	Len	Description	
						'A3'	Var	Security Attribute Template	Mandatory

Tag	Len	Description							Presence
				Tag	Len	Description			
				'A4'	Var	Security Attribute Extension for secret keys			Conditional – mutually exclusive with Tags 'A0', 'A1', and 'A5'
						Tag	Len	Description	
						'A3'	Var	Security Attribute Template	Mandatory
				Tag	Len	Description			
				'A5'	Var	Security Attribute Extension for private Diffie-Hellman keys			Conditional – mutually exclusive with Tags 'A0', 'A1', and 'A4'
						Tag	Len	Description	
						'A3'	Var	Security Attribute Template	Mandatory
				Tag	Len	Description			
				'AF'	Var	Proprietary Information			Optional
						Tag	Len	Description	
						'5C'	3	5F || Reference Data Qualifier || Cryptographic Mechanism Value	Mandatory

2.21 Card Management Capabilities

The Card Management Capabilities are defined in a template '80'. It describes the allowed transitions for card content management.

This value is defined inside the Control Parameters (CP) of the ADF.

Template '80' is composed of two bytes, each of which shall be coded as a bitmap, as defined in the following tables.

Table 2-23: Card Management Capabilities Byte 1

b8	b7	b6	b5	b4	b3	b2	b1	Supported Life Cycle State Transition
						1		Creation to Initialization
					1			Initialization to Operational Activated
				1				Creation to Operational Activated
		1						Operational Activated to Operational Deactivated
	1							Operational Deactivated to Operational Activated

Table 2-24: Card Management Capabilities Byte 2

b8	b7	b6	b5	b4	b3	b2	b1	Supported Life Cycle State Transition
					1			Initialization to Creation
				1				Operational Activated to Creation or Operational Deactivated to Creation

2.22 Reserved File Identifier Values

File Identifiers are used to reference files. The File Identifier values listed in Table 2-25 are reserved by [7816-4] and shall not be used by any files within an ISO-SD hierarchy:

Table 2-25: Reserved File Identifier Values

Reserved File Identifier Value	Meaning
'00 00'	Current file
'00 4D'	Extended header data list
'2F 00'	EF.DIR
'2F 01'	EF.ATR/INFO
'3F 00'	Master file
'3F FF'	File from the current context
'FF FF'	Current template

2.23 Control Reference Template

In [7816-4], the security architecture is shared across secure messaging, authentication (EXTERNAL / INTERNAL / GENERAL AUTHENTICATE, USER AUTHENTICATION), and general cryptographic services (PERFORM SECURITY OPERATION). The central concept is the Control Reference Template (CRT).

A CRT is a structure that holds an association of:

- A function (defined by the CRT tag)
- A cryptographic mechanism (reference into a list at DF level)
- A key reference (secret or public reference to unspecified storage)
- Management data such as initialization vectors, or counters

The structure of a CRT is defined in [7816-4], Table 54. The ISO Framework shall contain only templates with tags listed in Table 2-26.

Table 2-26: Templates Permitted in ISO-SD CRT

Tag	Description
'A4'	Authentication (AT)
'A6'	Key Agreement (KAT)
'AA'	Hashing (HT)
'B4'	Cryptographic Checksum (CCT)
'B6'	Digital Signature (DST)
'B8'	Confidentiality (CT)

3 ISO Security Domain (ISO-SD)

3.1 Introduction

The ISO Framework will be supported by a specific variant of Security Domains referred to as ISO Security Domains (ISO-SDs). Installation of ISO-SDs is out of the scope of this document and may be described in a specific configuration. ISO-SDs shall support ISO commands as defined in this specification.

An ISO-SD may store:

- Global Objects that may be retrieved by the ISO-SD itself and its associated Applications
- Local Objects that may only be accessed by the ISO-SD itself
- Security Objects (local or global) such as cryptographic keys that may be used to open a Secure Channel session
- Global Data Objects (BER-TLV) that may be used to store global information
- Local Data Objects (BER-TLV) that may be used to store specific information

Each ISO-SD shall support one or more Application Dedicated Files (ADFs).

An ADF is a structure hosting a Control Parameter Template and the application itself. The application could implement the Generic Identity Command Set specification [GICS].

The content of the application in an ADF is out of scope of this specification and only the Control Parameter Template is described in this document.

The ISO-SD is intended to support only one level of ADF.

The ISO-SD could be any of the following:

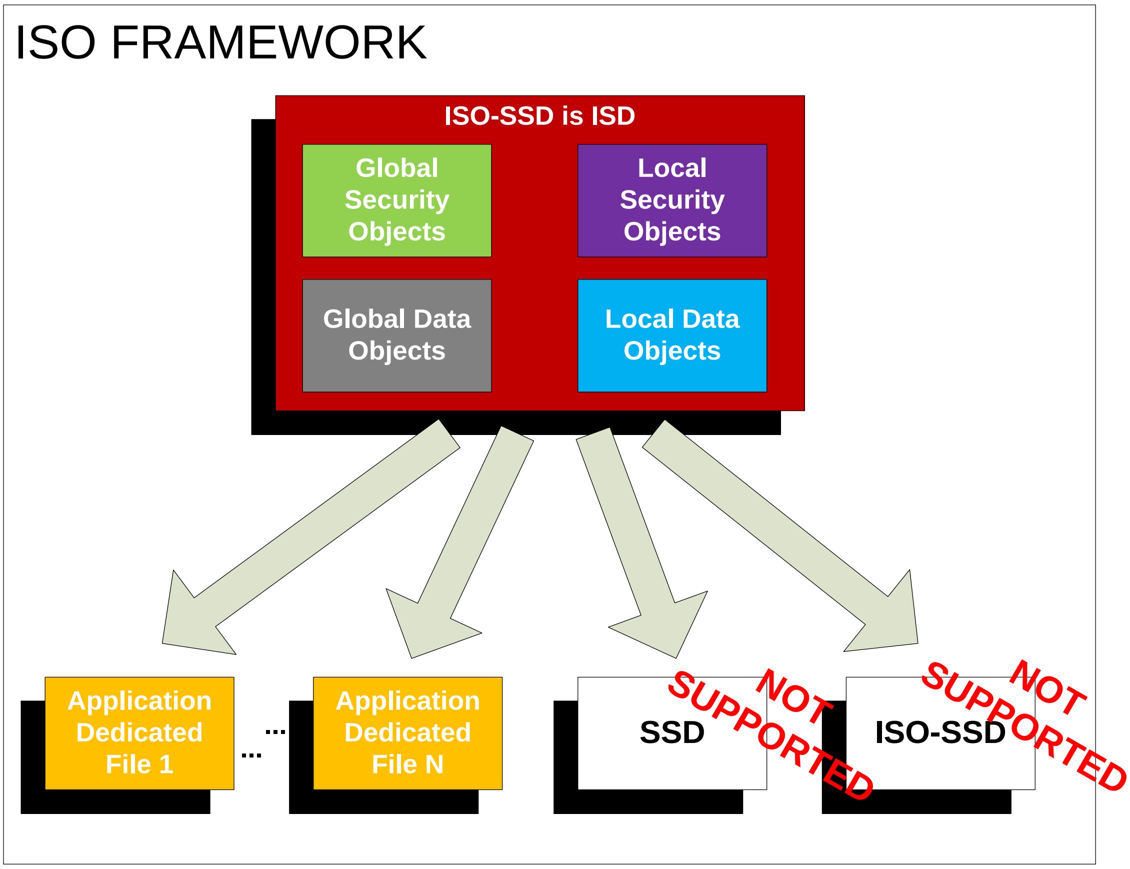
- the ISD
- an independent SSD with Authorized Management privilege (if supported by the card)
- an SSD that belongs to a GlobalPlatform Application (ISD or SSD)

3.2 Examples of ISO-SD Structure

3.2.1 Example 1: ISO-SD Is the ISD

This is the easiest use case where no GlobalPlatform hierarchy could be created:

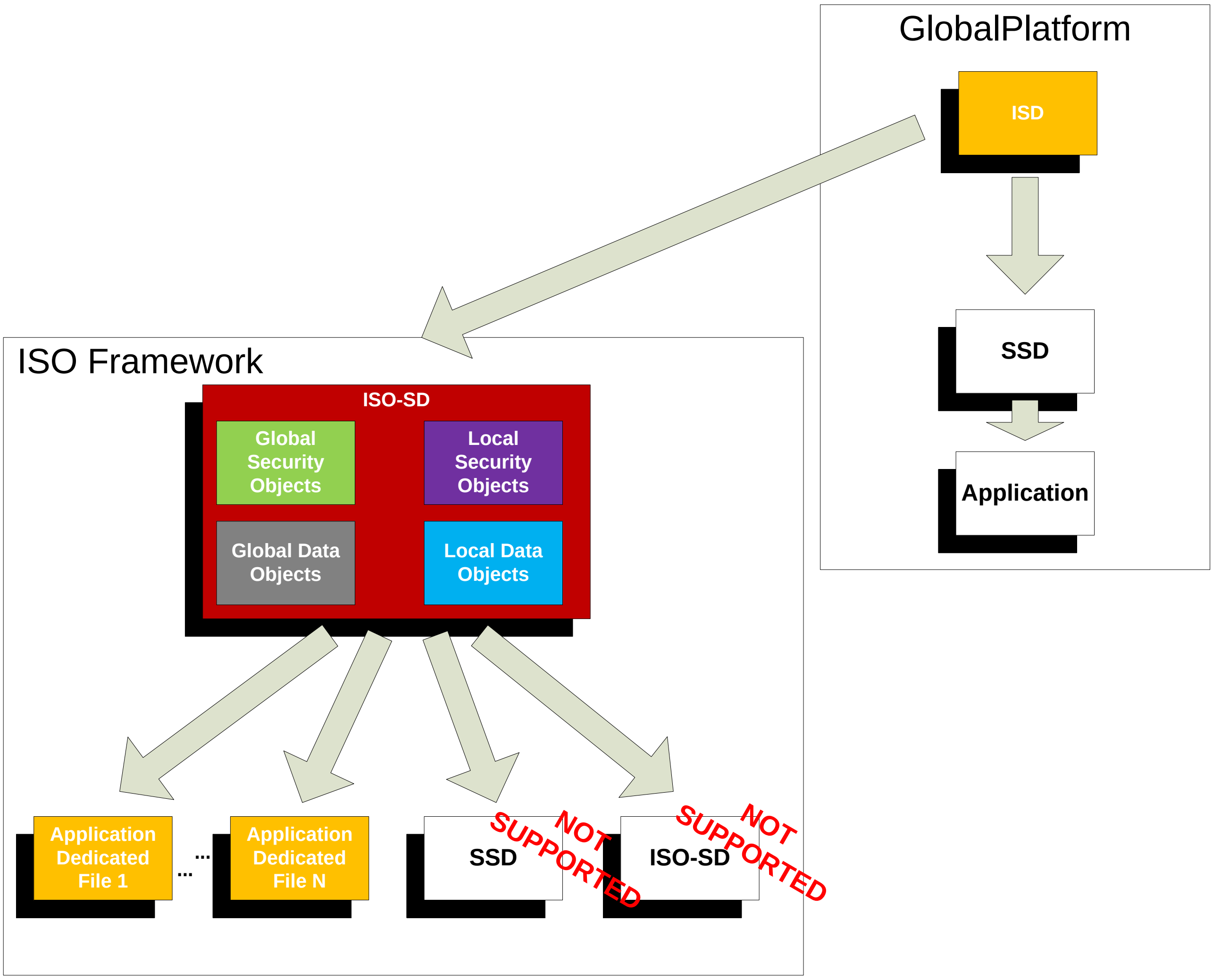
Figure 3-1: Example ISO-SD Structure: ISO-SD is ISD



3.2.2 Example 2: A GlobalPlatform Structure with an ISO Structure

This is an example where a GlobalPlatform and an ISO structure coexist on the smart card:

Figure 3-2: Example of a Complex ISO-SD Structure

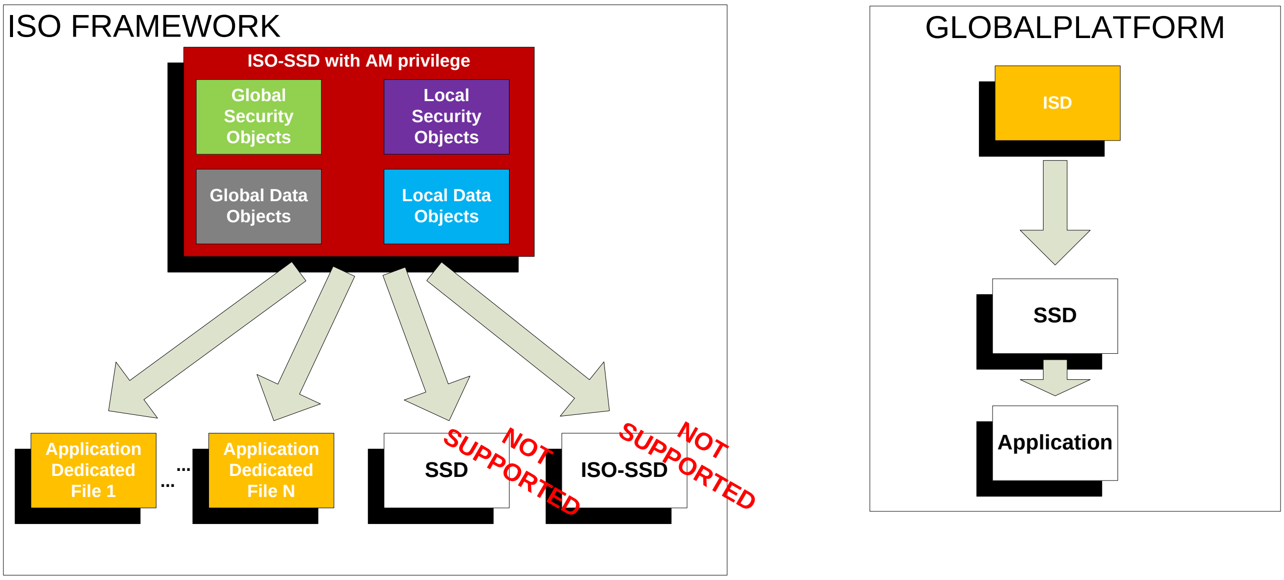


3.2.3 Example 3: ISO-SD Has AM Privilege

This is a use case where ISO-SD has the Authorized Management Privilege and manages its own structure.

In this example, a GlobalPlatform and an ISO structure coexist on the smart card

Figure 3-3: Example of an ISO-SD Structure with AM



3.3 Discovery Mechanism for ISO-SD

The ISO-SD shall expose at its interface some global and some local objects in order to facilitate the integration of a card supporting the ISO-SD command set with ISO/IEC 24727 middleware.

3.3.1 Global Data Objects

The ISO-SD shall provide a mechanism to access the following global Data Objects:

- An ISO/IEC 24727-2 Card Capability Descriptor (CCD) (Tag '7F 62')
- An EF.ATR/INFO
- An EF.DIR

The content of global Data Objects can be retrieved from any card application within the ISO-SD hierarchy, by using the GET DATA command, regardless of the currently selected EF or ADF. Retrieving a global Data Object content does not modify the Validity Area of the current selected ADF.

As a global object is seen by the complete ISO-SD hierarchy, no other elementary file within any application that belongs to the ISO-SD hierarchy shall have the same file ID as a global object.

3.3.1.1 Card Capability Descriptor (Tag '7F 62')

The Card Capability Descriptor (CCD) is retrievable by the GET DATA command with P1-P2 = '3F FF' and command data field containing '5C 02 7F 62'.

The content of the CCD is defined by [24727-2] and for the ISO-SD, the value shall be as described in Table 3-1.

Table 3-1: Card Capability Descriptor for ISO-SD (Tag '7F 62')

<table>
<tr><th>Tag</th><th>Length</th><th>Name</th><th colspan="4">Description</th><th>Presence</th></tr>
<tr><td rowspan="4">'7F 62'</td><td rowspan="4">Var</td><td>CCD</td><td colspan="4">See [24727-2] section 6.1.</td><td>Mandatory</td></tr>
<tr><th>Tag</th><th>Len</th><th>Name</th><th>Value</th><th>Description</th><th>Presence</th></tr>
<tr><td>'80'</td><td>1</td><td>PRO</td><td>'00'</td><td>Profile of [24727-2] with which this CCD complies.</td><td>Mandatory</td></tr>
<tr><td>'A0'</td><td>Var</td><td>SAID</td><td></td><td>Sequence of '4F' Data Objects containing Application Dedicated File Names of card applications</td><td>Mandatory</td></tr>
</table>

3.3.1.2 EF.ATR/INFO Content Description

This standard supports the ISO EF.ATR/INFO file that indicates operating characteristics of the card.

The EF.ATR/INFO may be called EF.ATR in contact card standards or specifications, and may be called EF.INFO in contactless card standards or specifications.

The contents of the EF.ATR/INFO can be freely retrieved by using a GET DATA command with P1-P2 = '2F 01' and a command data field of '5C 00' ('00 CB 2F 01 02 5C 00'). The response data field for the GET DATA command is the concatenation of all Data Objects (DO) which are present in EF.ATR/INFO. It contains the BER-TLV content of the EF.ATR/INFO.

The minimum content of the EF.ATR/INFO is listed in Table 3-2.

Table 3-2: EF.ATR/INFO Data Object Content

Tag	Length	Description	Value	Meaning

'43'	1	Card Service Data	'F4' or 'F5'	b8 = 1: Card-Application selection by full DF name
				b7 = 1: Card-Application selection by partial name
				b6 = 1: BER-TLV DO Present in EF.DIR
				b5 = 1: BER-TLV DO present in EF.ATR/INFO
				b4 = 0: READ BINARY not available to access EF.ATR/INFO and EF.DIR
				b3 = 1: EF.DIR and EF.ATR/INFO access service by GET DATA command (TLV structure)
				b2 = 0: READ RECORD not available to access EF.ATR/INFO and EF.DIR
				b1 = 0: Card with MF[3]
				b1 = 1: Card without MF

'47'	3	Card Capabilities	 '08' '01' 'C0'	See [7816-4] § 12.1.1.9 for the complete definition. Byte 1: Selection Method b4 = 1: Implicit DF selection Byte 2: Data Coding Byte Method b4 b3 b2 b1 = 0001: Data unit size is 1 byte Byte 3: Miscellaneous b8 = 1: Command chaining is supported b7 = 1: Extended Lc and Le fields are supported b6 = 0 b4 = 0: Logical channel number assignment by the interface device b3 b2 b1 from 000 to 100: a maximum of four Logical channels could be supported
'46'	Var	Pre-issuing DO Information		Data Object Information in ASCII to identify the card manufacturer and the product
'7F 62'	Var	Card Capability Descriptor		Data Objects in the CCD shall in accordance with Table 14 of [24727-2].
'5F 52'	Var	ATR Historical Bytes	variable	The historical bytes (string of up to 15 bytes, as defined in ISO/IEC 7816-3) indicate operating characteristics of the card. When a card answers to reset, the Answer-to-Reset may contain historical bytes.

Tag '7F 66' may optionally be used in EF.ATR to indicate the maximum input/output buffer size when using extended length. In this case, this shall be indicated in ATR historical bytes accordingly (3rd software function table).

[3] Support for the MF remains out of the scope of this document.

3.3.1.3 EF.DIR Tag Content Description

This standard supports the ISO EF.DIR file that lists the Application Dedicated File Names present in the ISO-SD hierarchy.

The EF.DIR shall contain only the ISO-SD hierarchy information.

Its content is constructed automatically by the ISO-SD and updated whenever a new application is created or deleted.

The content of EF.DIR can be retrieved without any authentication or secure messaging by using a GET DATA command that has P1-P2 = '2F 00' and a command data field of '5C 00'.

The response data field of the GET DATA command contains the concatenation of all Application Templates.

3.3.2 Local Data Objects

The ISO-SD shall expose at its interface:

- A Card Management Service Template (Tag '7F 64')
- A Card Data Template (Tag '66')
- An Application Template (Tag '61')
- Control Parameters accessible by a GET DATA command (Tag '62')
- An ISO/IEC 24727-2 Application Capability Descriptor (ACD) (Tag '7F 63')

3.3.2.1 Card Management Service Template (Tag '7F 64')

The Card Management Service Template is a collection of Data Objects describing the card management application capabilities.

This template shall be accessible from the ISO-SD with a GET DATA command.

For retrieval, a GET DATA command with P1-P2 = '3F FF', Lc = '04', and Data = '5C 02 7F 64' shall return the mandatory Card Management Service Template in the response data field when the card manager application is selected.

In the scope of this document, the Card Management Service Template shall be as defined in Table 3-3.

Table 3-3: Card Management Service Template for ISO-SD (Tag '7F 64')

Tag	Length	Name				Presence
'7F 64'	Var	Card Management Service Template				Mandatory
		Tag	**Len**	**Name**	**Description**	**Presence**
		'80'	2	Card Management Capabilities	Capabilities supported by the ISO-SD, as defined in section 2.21 See: • section 4.5.1, APPLICATION MANAGEMENT REQUEST (AMR) Command • section 4.5.3, REMOVE APPLICATION Command • section 4.5.4, ACTIVATE FILE (ADF) Command • section 4.5.5, DEACTIVATE FILE (ADF) Command	Mandatory
		'81'	Var	Card Management Scheme	Card management scheme name and version: Object Identifier value (see ISO/IEC 8825-1 [8825-1]) indicating the scheme name and version (major and minor) used to manage the card and its applications See section 3.3.2.2.	Mandatory
		'82'	Var	Card Identification Scheme	Card identification scheme indicator: Object Identifier value (see [8825-1]) indicating the scheme used to identify the card uniquely See section 3.3.2.4.	Mandatory
		'4F'	Var	ISO-SD application AID	Application Dedicated File Name (that is, AID) used to select the ISO-SD	Mandatory

3.3.2.2 Card Management Scheme (Tag '81')

In compliance with [7816-13], a card management scheme is defined.

The Card Management Scheme is unique, fixed, and refers to the card management scheme of a GICS card.

The Card Management Scheme is specified by an OID value within the Data Object (tag '81') of the Card Management Service Template (tag '7F 64').

The Object Identifier (OID) referencing the collection of objects defined in this standard is:

ISO-SD-management-scheme-OID ::={ ISO-SD OID (see chapter 3.3.2.4) parameter (2) }

3.3.2.3 Card Data for ISO-SD (Tag '66')

The Card Data Template is a collection of Data Objects identifying the card, its state, and its applications for the purpose of managing or synchronizing the card from a Card Management System.

The Card Data shall be returned by the card in response to a GET DATA command with P1-P2 = '3F FF' and command data field containing '5C 01 66'.

Table 3-4: Card Data Template for ISO-SD (Tag '66')

Tag	Length	Name			Presence
'66'	Var	Card Data			Mandatory
		Tag	**Len**	**Description**	**Presence**
		'45'	Var	Card Issuer Data This card reference value is used to uniquely identify the card. Its format is referenced by the Card Identification Scheme OID.	Mandatory

3.3.2.4 Card Identification Scheme (Tag '82')

In compliance with [7816-13], a Card Identification Scheme is defined.

The Card Identification Scheme is unique, fixed, and refers to the location, access method, syntax, and semantic rules that specify interoperable card identification.

The Card Identification Scheme is specified by an OID value within the Data Object (tag '82') of the Card Management Service Template (tag '7F 64').

The Object Identifier (OID) referencing the collection of objects defined in this standard is:

ISO-SD OID ::={ iso (1) member-body (2) country-USA (840) incits (114402) incits-committee-identifier (21012) grouping (2) }

ISO-SD-identification-scheme-OID::={ ISO-SD OID parameter (1) }

According to the Card Identification Scheme, the value of the Card Issuer Data Object (tag '45') in the Card Data Template contains interoperable identification information. The value shall be determined by the card issuer to uniquely identify a card in the context of operation. The format is the concatenation of the Issuer Identification Number (IIN – 6 bytes) and the Card Identification Number (CIN – 10 bytes). The method to produce unique values for the IIN and CIN is outside the scope of this specification.

3.3.2.5 Application Template (Tag '61')

The Application Template (tag '61') allows retrieving the characteristics of any application that belongs to the ISO-SD hierarchy or of the ISO-SD itself. It may be accessed as the FCI response to the SELECT command APDU with P1-P2 = '04 00'.

The value shall be as defined in Table 3-5.

Table 3-5: Application Template for ISO-SD (Tag '61')

<table>
<tr><th>Tag</th><th>Length</th><th colspan="3">Description</th></tr>
<tr><td rowspan="4">'61'</td><td rowspan="4">Var</td><td colspan="3">Application Template</td></tr>
<tr><th>Tag</th><th>Len</th><th>Description</th></tr>
<tr><td>'4F'</td><td>Var</td><td>ADF Name (that is, AID) of the ISO-SD</td></tr>
<tr><td>'79'</td><td>Var</td><td>Coexistent tag allocation authority template</td></tr>
</table>

3.3.2.6 Control Parameters (CP) (Tag '62')

The Control Parameters of the ISO-SD application indicate the Security Attributes relative to Card Application Management Commands.

The value shall be returned by the card in response to a GET DATA command with P1-P2 = '3F FF' and command data field containing '5C 01 62'.

The value shall be as defined in Table 3-6.

Table 3-6: Control Parameter Template for ISO-SD Application (Tag '62')

<table>
<tr><th>Tag</th><th>Length</th><th colspan="6">Description</th></tr>
<tr><td rowspan="12">'62'</td><td rowspan="12">Var</td><td colspan="6">Control Parameter Template</td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th colspan="3">Description</th></tr>
<tr><td>'82'</td><td>1</td><td>'38'</td><td colspan="3">ADF File</td></tr>
<tr><td>'84'</td><td>Var</td><td></td><td colspan="3">ADF Name (that is, AID) of the ISO-SD</td></tr>
<tr><td>'8A'</td><td>1</td><td></td><td colspan="3">ISO Life Cycle State (This attribute is read only and shall not be present during the CP creation. It can only be read when retrieving the Control Parameters.)</td></tr>
<tr><td rowspan="5">'A3'</td><td rowspan="5">Var</td><td rowspan="5"></td><td colspan="3">Security Attribute Template</td></tr>
<tr><th>Tag</th><th>Len</th><th>Description</th></tr>
<tr><td>'91'</td><td>1</td><td>Physical interface definition</td></tr>
<tr><td>'9C'</td><td>Var</td><td>Security Attribute in Compact Format (see section 2.18.2)</td></tr>
<tr><td>'5C'</td><td>0</td><td>This empty tag (length equal to 0) indicates that the security rules shall apply to all Data Objects in the base template to which tag '62' belongs.</td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th colspan="3">Description</th></tr>
<tr><td>'A3'</td><td>Var</td><td></td><td colspan="3">Tag 'A3' may be present several times.</td></tr>
</table>

3.3.2.7 Application Capability Descriptor (Tag '7F 63')

See section 4.3.

3.4 Life Cycle State

Table 3-7 shows how the life cycle states of an ISO-SD and an ADF (and allowed transitions) are mapped onto those described in [GPCS].

Table 3-7: Life Cycle Mapping between ISO and GlobalPlatform for ISO-SD

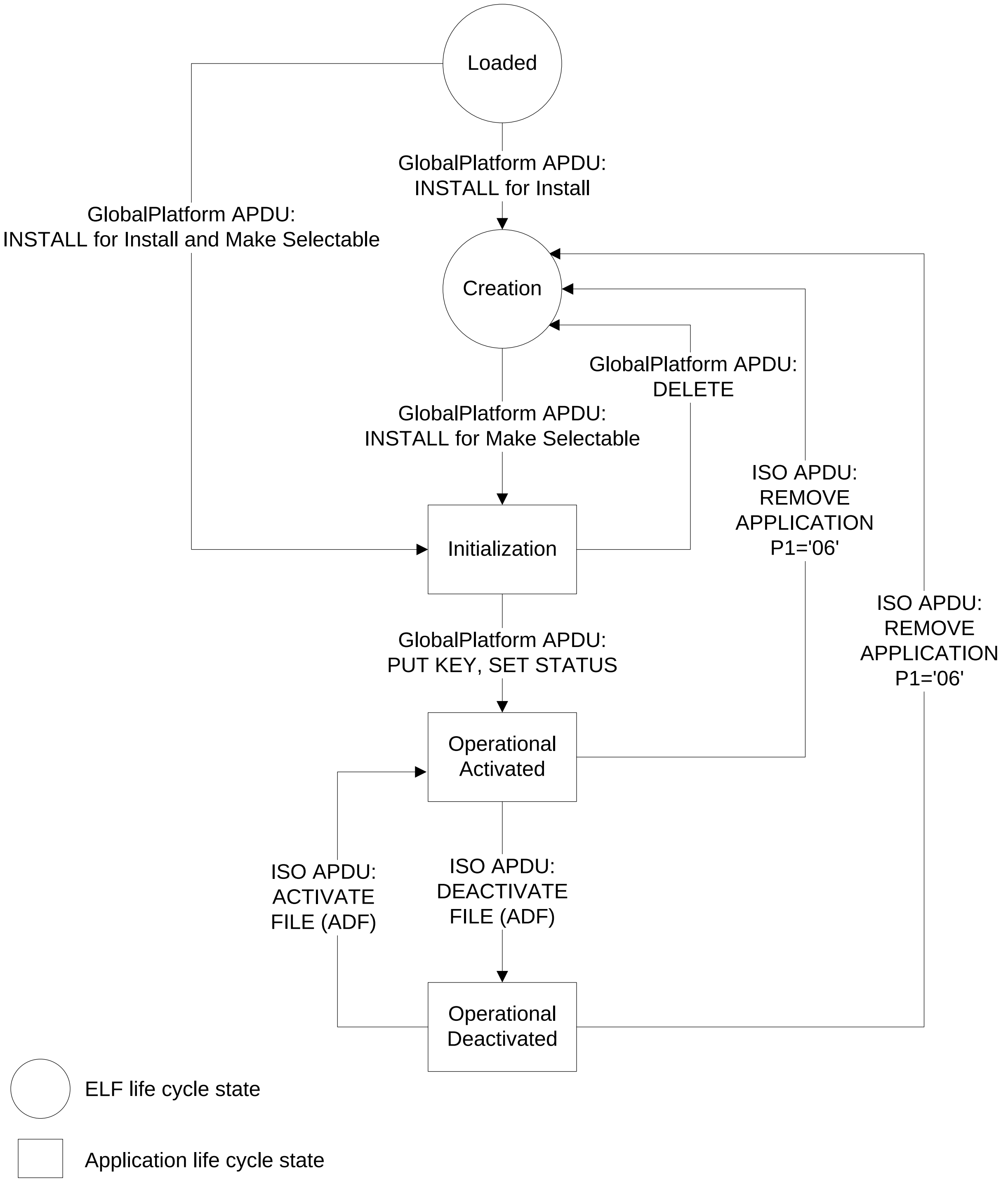
ISO Value	ISO State	GlobalPlatform State	GlobalPlatform Value
'01'	Creation	INSTALLED	'03'
'03'	Initialization	SELECTABLE	'07'
'05'	Operational Activated	PERSONALIZED	'0F'
'04'	Operational Deactivated	LOCKED	bit 8 set to one

When the ISO-SD is in the SELECTABLE state, the ISO-SD shall support the ISO APDU commands defined in this specification. Support for other commands (e.g. with GlobalPlatform CLASS byte) is out of scope of this specification.

When the ISO-SD is in the PERSONALIZED state or LOCKED state, the ISO-SD shall support only the ISO-SD APDU commands defined in this specification.

The ISO Life Cycle State encoding shall be used within ISO-SD APDU commands and responses.

Figure 3-4: Life Cycle State Transitions for ISO-SD


Loaded
GlobalPlatform APDU: INSTALL for Install
GlobalPlatform APDU: INSTALL for Install and Make Selectable
Creation
GlobalPlatform APDU: DELETE
GlobalPlatform APDU: INSTALL for Make Selectable
ISO APDU: REMOVE APPLICATION P1='06'
Initialization
ISO APDU: REMOVE APPLICATION P1='06'
GlobalPlatform APDU: PUT KEY, SET STATUS
Operational Activated
ISO APDU: ACTIVATE FILE (ADF)
ISO APDU: DEACTIVATE FILE (ADF)
Operational Deactivated
ELF life cycle state
Application life cycle state

3.5 Elementary Files and Dedicated Files

ISO-SD does not support EF or DF management; however, any ADF present in the ISO-SD hierarchy could support EF or DF features if needed.

3.6 Data Objects

Two types of Data Objects exist: those that have control parameters and those that do not.

- Data Objects that do not have control parameters belong to an EF or ADF and they inherit the security properties of the EF or ADF in which they belong, as discussed in section 3.5.
- Data Objects that have control parameters are called Security Objects. See section 2.20.

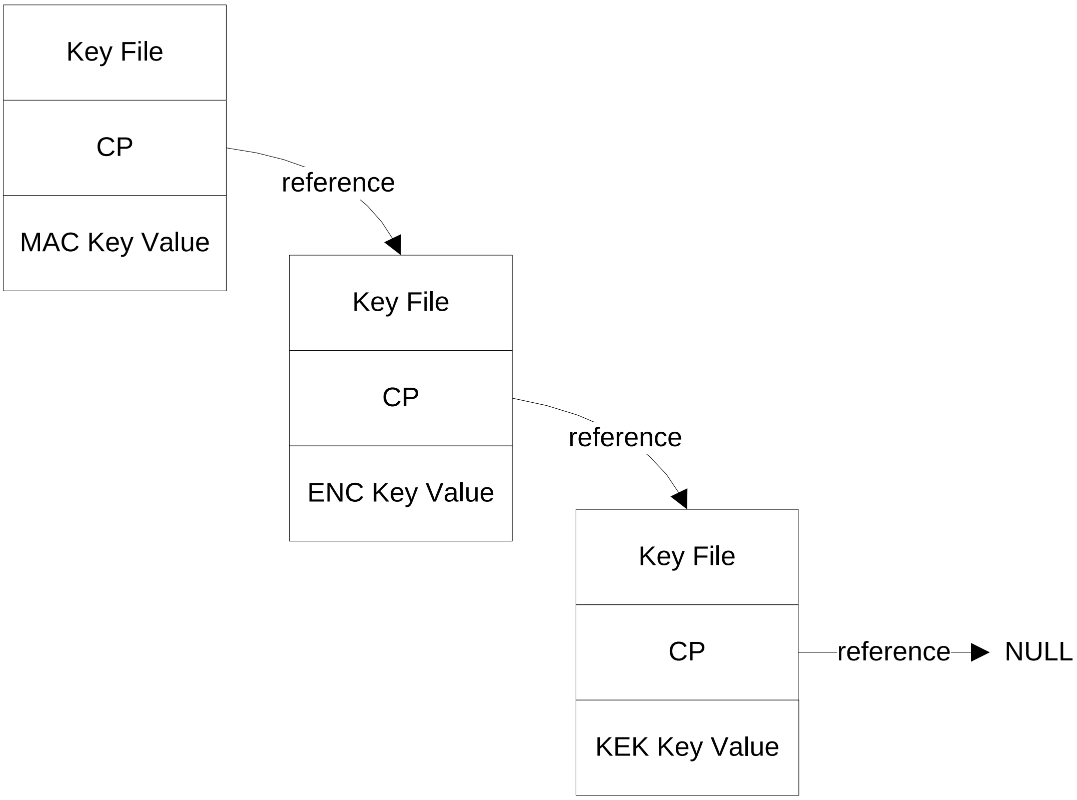
3.7 Key Management

For the ISO-SD, each key is managed in a Security Object.

A Security Object can contain only one key, but because three static keys are needed for SCP '03' authentication, a Security Object is able, by using tag 'AF' in its Control Parameter Template (tag '62'), to reference another Security Object.

Figure 3-5 illustrates the reference mechanism.

Figure 3-5: Key Management



For SCP '03', the command GENERAL AUTHENTICATE will always reference the MAC Key. Others keys can be found by using internal reference inside the Control Parameters (CP) of the MAC Key.

3.8 ISO-SD APDU Commands

3.8.1 Introduction

This section describes the ISO APDUs supported by the ISO-SD.

Table 3-8 lists the ISO-SD APDUs and the corresponding GlobalPlatform APDUs.

Table 3-8: Mapping Between ISO-SD APDUs and GlobalPlatform APDUs

ISO-SD APDU Description	GlobalPlatform APDU Description	ISO-SD APDU Described in:
GET DATA	GET DATA	section 3.8.2
PUT DATA (data management)	STORE DATA TLV	section 3.8.3
PUT DATA (key)	STORE DATA for keys	section 3.8.4
GENERAL AUTHENTICATE	INITIALIZE UPDATE EXTERNAL AUTHENTICATE	section 3.8.5
CREATE DATA [4]	N/A	section 3.8.6
DELETE DATA [4]	N/A	section 3.8.7
APPLICATION MANAGEMENT REQUEST	INSTALL [for load] INSTALL [for install] INSTALL [for install and make selectable]	section 4.5.1
LOAD APPLICATION	LOAD	section 4.5.2
REMOVE APPLICATION	DELETE	section 4.5.3
ACTIVATE FILE	SET STATUS to unlock applications	section 4.5.4
DEACTIVATE FILE	SET STATUS to lock applications	section 4.5.5

[4] The CREATE DATA and DELETE DATA commands are under validation for a new version of ISO 7816-9.

3.8.2 GET DATA Command

3.8.2.1 Description

The GET DATA command retrieves Data Object(s) from the ISO-SD (see section 3.3).

3.8.2.2 Condition of Usage

There is no specific security condition to be able to use this command.

3.8.2.3 APDU Command

Table 3-9: GET DATA Command

CLA	See section 2.1.
INS	'CB'
P1 – P2	One of the following: • '3F FF' for the current selected ADF • File Identifier
Lc	Variable
Command Data Field	Data Object tag listed in Table 3-10
Le	Variable
Response Data Field	Variable

3.8.2.4 Command Data Field

When the P1-P2 is equal to '3F FF', the command data field may be composed of any one of the values listed in Table 3-10.

Table 3-10: GET DATA Command Data Field

Command Data Field	Meaning
'5C 02 7F 62'	Return Card Capability Descriptor
'5C 02 7F 63'	Return Application Capability Descriptor
'5C 02 7F 64'	Return Card Management Service Template
'5C 01 66'	Return Card Data Template
'5C 01 62'	Return Control Parameters of the Security Object identified by P1-P2

When the P1-P2 is not equal to '3F FF', the command data field may be composed of the tag '5C'.

The length of this tag could be either '00', to get the complete object content of the file pointed to by P1-P2, or variable. In that case, the tag '5C' contains the data object to read.

Example:

The APDU '00' 'CB' '2F' '01' '04' '5C' '02' '5F' '52' reads the historical bytes value defined in template '5F 52' contained in the EF ATR whose identifier is '2F 01'.

3.8.2.5 Status Word

Table 3-11: GET DATA Status Word

SW1 SW2	Description
'62 81'	Part of returned data may be corrupted
'67 00'	Wrong length
'69 82'	Security condition not satisfied
'69 85'	Conditions of use not satisfied
'6A 80'	Incorrect parameters in the command data field
'6A 82'	Data Object not found
'90 00'	Successful execution

3.8.3 PUT DATA (Data Management) Command

3.8.3.1 Description

The PUT DATA command can be used for data loading.

It can be used to personalize the Card Capability Descriptor, Application Capability Descriptor, Card Management Service Template, or Card Data Template, and for Data Object management.

It corresponds to the STORE DATA command.

3.8.3.2 Condition of Usage

This command shall be performed within a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

3.8.3.3 APDU Command

Table 3-12: PUT DATA (Data Management) Command

CLA	See section 2.1.
INS	'DB'
P1 – P2	One of the following: • '3F FF' for the selected file • the File Identifier
Lc	Data field length
Command Data Field	Command data field (BER-TLV), see Table 3-13
Le	'00'
Response Data Field	None

3.8.3.4 Command Data Field

Table 3-13: PUT DATA (Data Management) Command Data Field

Tag	Length	Value	Presence
'5C'	Var	Tag of the Data Object to create or modify	Mandatory
'53'	Var	If present, new value to be assigned to the Data Object If omitted: • If the Data Object does not exist, a Data Object with empty value is created. • If the Data Object exists, it shall be deleted. In other words, an existing Data Object can be deleted by issuing a PUT DATA of that Data Object with no value; i.e. with an object BER-TLV length set to zero.	Optional

Only one Data Object may be created, updated, or deleted at a time.

3.8.3.5 Status Word

Table 3-14: PUT DATA (Data Management) Status Word

SW1 SW2	Description
'65 81'	Memory failure
'67 00'	Wrong length
'69 82'	Security status not satisfied
'69 85'	Conditions of use not satisfied
'6A 80'	Incorrect parameters in the command data field
'6A 84'	Not enough memory space
'6A 85'	Lc inconsistent with TLV structure
'90 00'	Successful execution

3.8.4 PUT DATA Key Command

3.8.4.1 Description

The PUT DATA command can be used for key loading or update.

It corresponds to the GlobalPlatform PUT KEY command.

The corresponding Security Object shall be created before pushing a value.

3.8.4.2 Condition of Usage

This command shall be performed inside a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

3.8.4.3 APDU Command

Table 3-15: PUT DATA (Key) Command

CLA	See section 2.1.
INS	'DB'
P1	'3F'
P2	'FF'
Lc	Data field length
Command Data Field	Command data field (BER-TLV), see Table 3-16
Le	Not present
Response Data Field	None

3.8.4.4 Command Data Field

Table 3-16: PUT DATA (Key) Command Data Field

Tag	Length	Value	Presence
'5C'	3	'5F' || Reference Data Qualifier of the key || Cryptographic Mechanism Reference Values	Mandatory
'87'	Var	Ciphered Key Data	Present for symmetric key only
		Key Component Type || Encryption Indicator Byte || Ciphered Key Data	Present for asymmetric public key only
		Key Component Type || Ciphered Key Data	Present for asymmetric private key only
'8E'	3	Key Check Value: Three leftmost bytes of the result of encrypting sixteen consecutive bytes of '01' with the plaintext key	Present for symmetric key only

The Key Component Type is defined as shown in Table 3-17.

Table 3-17: Key Component Type in Tag '87'

Key Component Type	Description
'82'	RSA Public key – Public exponent e component
'81'	RSA Public key – Public modulus N component
'90'	RSA Private key – Private exponent d component
'92'	RSA Private key – Chinese Remainder P component
'93'	RSA Private key – Chinese Remainder Q component
'94'	RSA Private key – Chinese Remainder PQ component
'95'	RSA Private key – Chinese Remainder DP1 component
'96'	RSA Private key – Chinese Remainder DQ1 component
'86'	Q public Key – ECC ('04||X||Y')
'90'	D private key – ECC

As creating or updating a public key component could be considered a non-sensitive operation, the encryption of the key component is optional and is indicated by using the encryption indicator byte.

The Encryption Indicator Byte is '00' for clear mode and '01' for encrypted mode. For padding, the byte string '80 00 00 … 00' is appended to the key value. The padding is applied only when the key length is not a multiple of the encryption block size.

If Data Object '87' is too long for a single command, then command chaining shall apply; the value field of the Data Object is the concatenation of the command data fields.

The keys are imported encrypted in the command data field. The encryption is performed according to the Sensitive Data Encryption algorithm as described in [Amd D] section 6.2.8.

3.8.4.5 Status Word

Table 3-18: PUT DATA (Key) Status Word

SW1 SW2	Description
'67 00'	Wrong length
'69 82'	Security status not satisfied
'69 85'	Conditions of use not satisfied
'6A 81'	Function not supported
'6A 84'	Not enough memory space in the file
'6A 85'	Lc inconsistent with TLV structure
'90 00'	Successful execution

3.8.5 GENERAL AUTHENTICATE Command

3.8.5.1 Description

The GENERAL AUTHENTICATE command is used to establish a Secure Channel session, according to Secure Channel Protocol '03' described in GPCS Amendment D [Amd D].

This command replaces both the INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands described in [Amd D] and therefore shall be issued twice to achieve mutual authentication and establish a Secure Channel session. Although the encoding differs between the two occurrences of the GENERAL AUTHENTICATE command, the same information can be retrieved from either occurrence, and the protocol steps and algorithms remain the same.

3.8.5.2 Condition of Usage

There is no specific security condition to be able to use this command.

When the sequence of this command is correctly issued, the host is considered as AUTHENTICATED, otherwise, the security status is modified to ANY_AUTHENTICATED.

When the sequence of this command is not correctly issued, the security status is modified to not ANY_AUTHENTICATED.

(See [GPCS] section 10.6, Security Level.)

3.8.5.3 APDU Command

Table 3-19: GENERAL AUTHENTICATE Command

CLA	See section 2.1.
INS	'87'
P1	Cryptographic Mechanism Reference value. See Table 2-2 on page 15.
P2	Reference Data Qualifier (or '00' if the information is to be retrieved from the command data field)
Lc	Conditional
Command Data Field	Dynamic Authentication Data Objects: Data related to the authentication protocol. See section 3.8.5.5.
Le	Conditional
Response Data Field	Dynamic Authentication Data Objects: Data related to the authentication protocol. See section 3.8.5.5.

3.8.5.4 Reference Control Parameter P2: Reference Data Qualifier Value

If the value of P2 is '00', then no information is provided by P2 and the actual Reference Data Qualifier value may be retrieved from the BER-TLV structure of the command data field.

Otherwise, P2 defines the Reference Data Qualifier.

3.8.5.5 Command Data Field and Response Data Field

The command data field shall include a Dynamic Authentication Data Objects template (tag '7C') containing one or more of the following Data Objects:

- Host Challenge (length is 8 bytes)
- Response Cryptogram (length is 8 bytes)
- Key Input Information

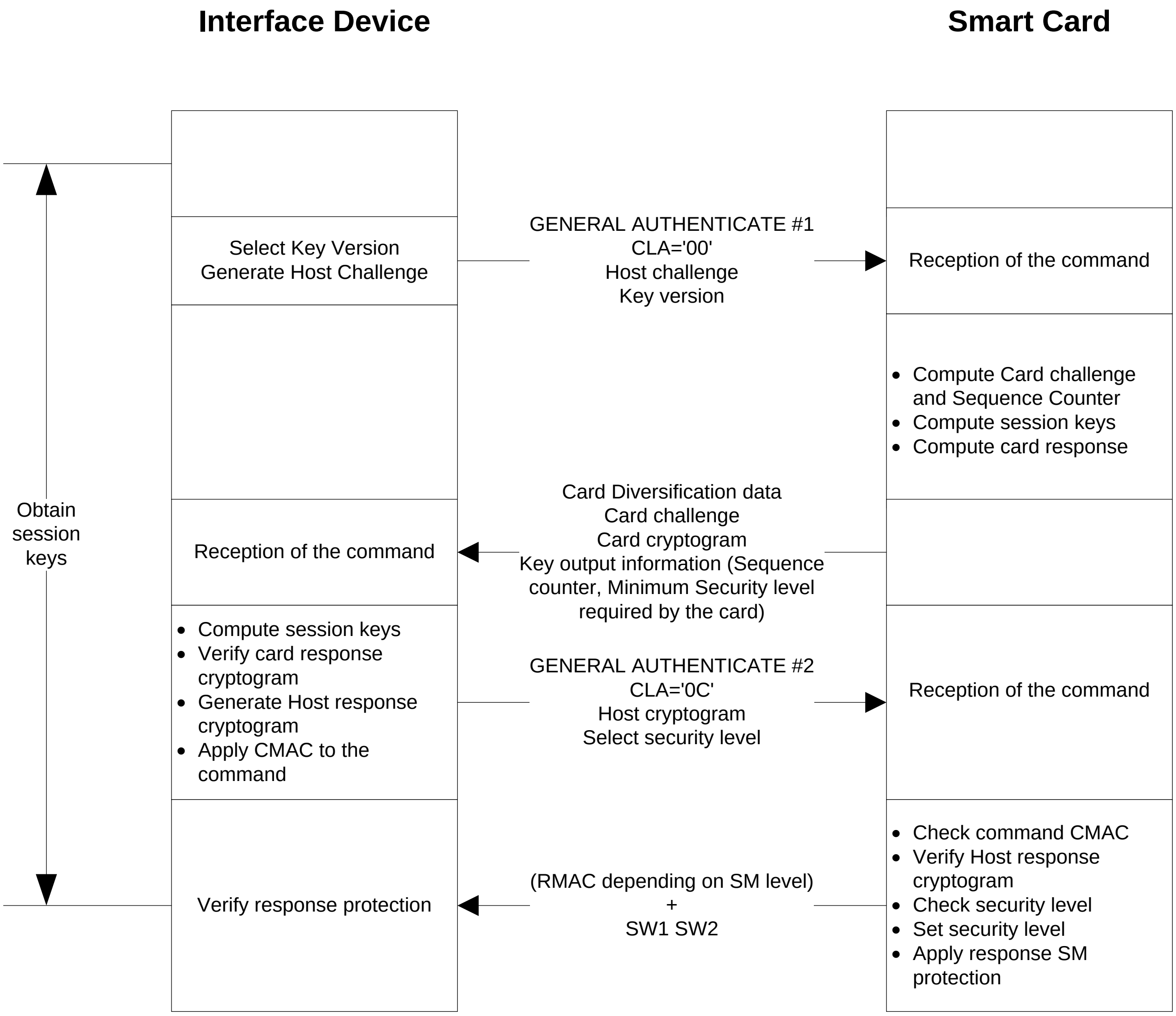
The response data field shall include a Dynamic Authentication Data Objects template (tag '7C') containing one or more of the following Data Objects:

- Key Output Information
- Card Diversification Data
- Card Challenge (length is 8 bytes)
- Response Cryptogram (length is 8 bytes)
- Sequence Counter, included if and only if pseudo-random mode is used for SCP '03'

Table 3-20: GENERAL AUTHENTICATE Command Data Field and Response Data Field

CLA	INS	P1	P2	Command Data Field	Response Data Field
'00'	'87'	'27'	'00'	{ Dynamic Authentication Data Objects ({Key Input Information}-{Host Challenge})}	{ Dynamic Authentication Data Objects ({Card Diversification Data}-{Key Output Information}-{Card Challenge}-{Response Cryptogram}-{Sequence Counter})}
'0C'	'87'	Security level	'00'	{ Dynamic Authentication Data Objects ({Response Cryptogram})}	

Figure 3-6: Session Key Establishment Using SCP '03'



During the opening of a secure channel session, if the security level requested by the host in the GENERAL AUTHENTICATE #2 command does not satisfy the minimum security level requested by the card, the authentication is rejected by the ISO-SD and the secure channel is not opened.

3.8.5.6 Status Word

Table 3-21: GENERAL AUTHENTICATE Status Word

SW1 SW2	Description
'63 00'	Verification failed, no information given
'69 82'	Security status not satisfied
'69 83'	Authentication method blocked
'69 85'	Conditions of use not satisfied
'6A 80'	Incorrect parameter in command data field
'6A 86'	Incorrect parameters P1-P2
'6A 88'	Reference data not found
'90 00'	Successful execution

3.8.6 CREATE DATA Command

3.8.6.1 Description

The CREATE DATA command is used to create Data Objects with control parameters.

This command shall be used to personalize the ISO-SD to create Security Objects. Once the Security Object has been created, its value is manageable by the PUT DATA command.

The CREATE DATA command places the Data Object immediately under the Application DF.

The Data Object created by the command is not selected.

Note: The CREATE DATA command is a new command defined by a new version of ISO/IEC 7816-9 that is still under validation.

3.8.6.2 Condition of Usage

The CREATE DATA command shall not execute unless the security condition associated with the access mode CREATE DATA in the Security Attribute of the currently selected ADF evaluates to TRUE with respect to the current security status.

3.8.6.3 APDU Command

Table 3-22: CREATE DATA Command

CLA	See section 2.1.
INS	'D5'
P1	'00'
P2	'00'
Lc	Var
Command Data Field	Control Parameter Template (tag '62') of the Data Object
Le	'00'
Response Data Field	Absent

3.8.6.4 Command Data Field

The command data field shall be a Control Parameter Template (tag '62') containing only Data Objects described in the Control Parameters for Security Objects (see Table 2-22 on page 28).

3.8.6.5 Status Word

Table 3-23: CREATE DATA Status Word

SW1 SW2	Description
'65 81'	Memory failure
'67 00'	Wrong length
'69 82'	Security status not satisfied
'69 85'	Conditions of use not satisfied
'6A 80'	Incorrect parameters in the command data field
'6A 86'	Incorrect P1-P2
'6A 89'	Data object already exists
'90 00'	Successful execution

3.8.7 DELETE DATA Command

3.8.7.1 Description

The DELETE DATA command is used for Data Object deletion.

The command may be used to delete Security Objects.

Note: The DELETE DATA command is a new command defined by a new version of ISO/IEC 7816-9 that is still under validation.

3.8.7.2 Condition of Usage

The DELETE DATA command shall not execute unless the security condition associated with the access mode DELETE DATA in the Security Attribute of the currently selected ADF evaluates to TRUE with respect to the current security status.

3.8.7.3 APDU Command

Table 3-24: DELETE DATA Command

CLA	See section 2.1.
INS	'EE'
P1	'00'
P2	'00'
Lc	Var
Command Data Field	Data Object Identifier
Le	'00'
Response Data Field	Absent

3.8.7.4 Command Data Field

The command data field is composed of the Data Object Identifier.

3.8.7.5 Status Word

Table 3-25: DELETE DATA Status Word

SW1 SW2	Description
'67 00'	Wrong length
'69 82'	Security status not satisfied
'69 85'	Conditions of use not satisfied
'6A 82'	Data Object not found
'6A 86'	Incorrect P1-P2
'90 00'	Successful execution

3.9 Secure Messaging

This section describes how to apply SCP '03' secure messaging over APDU commands.

The secure messaging is a mechanism to protect input and output data transmission in integrity and in confidentiality. The security level of the SCP '03' session defines the required input and output protection. This security level is defined by the GENERAL AUTHENTICATE command in the template dynamic authentication data objects.

Bit three (b3) and four (b4) in the CLASS byte of the command shall be set to one to indicate that input and output secure messaging is present.

When bit five (b5) is also set to one (i.e. command chaining is used), secure messaging shall be applied to the entire message before command fragmentation for data transportation (see section 3.10.5).

A data field in Secure Messaging Format shall be constructed as described in the following sections.

3.9.1 Secure Message Data Objects

When secure messaging is present, it shall be managed in BER-TLV format as defined in [7816-4] Table 49.

Only data objects with tags listed in Table 3-26 shall be used to construct data objects in a data field in Secure Messaging format.

Table 3-26: Secure Message Data Objects

Tag	Description
'81'	Plain value not encoded in BER-TLV
'87'	Cryptogram (padding-content indicator followed by cryptogram not BER-TLV encoded)
'8E'	Cryptographic checksum
'97'	Le
'99'	Status word

3.9.2 Secure Messaging Format

3.9.2.1 Abbreviations Used in this Section

- PB = Variable padding bytes for block authentication ([7816-4] 10.2.3.1)
- PI = Cryptogram Padding Indicator ([7816-4] 10.2.2, value always '01')
- C-MAC = Command-MAC
- R-MAC = Response-MAC

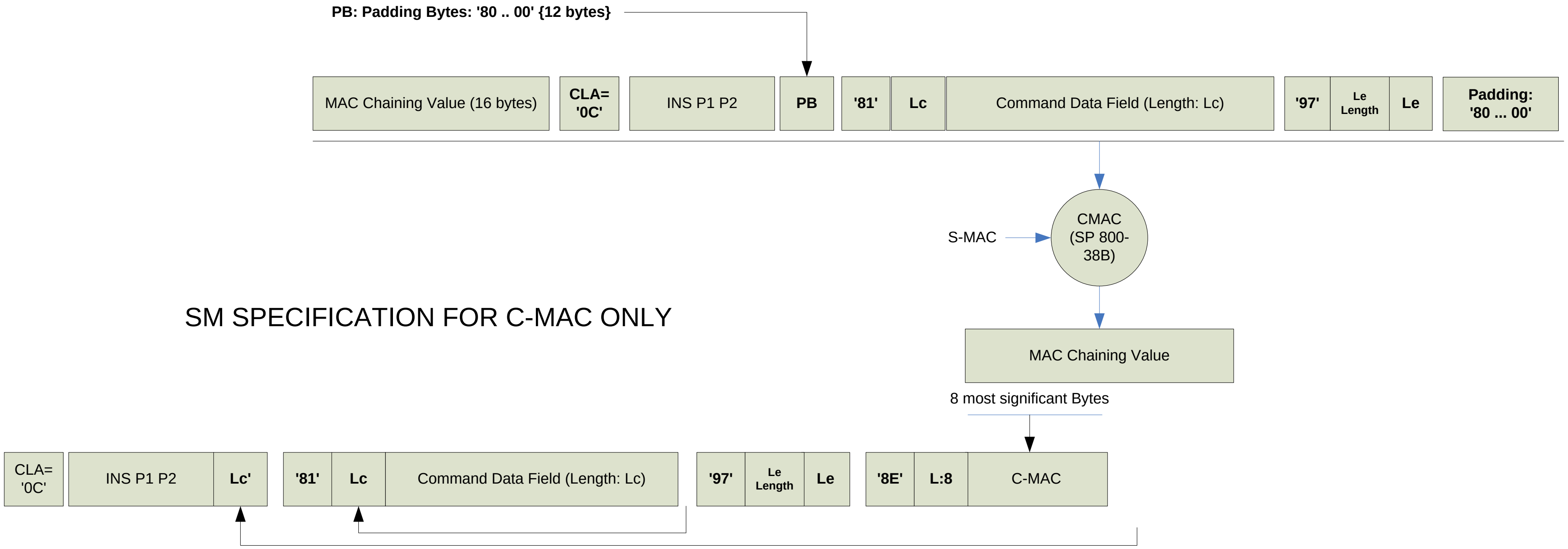
3.9.2.2 Computation Rules

- Comply with [7816-4] 10.2.3.1
- Not encapsulated header is permitted provided CLA is set to:
  - '0C' (Last or Single command)
  - '1C' (First or Intermediate command in a chain)
- The initial MAC chaining value shall be set to '00000000000000000000000000000000'
- MAC chaining value is defined in [Amd D] in section 6.2.4 for C-MAC, section 6.2.5 for R-MAC.
- MAC shall be computed as specified for CMAC in NIST SP 800-38B ([SP800-38b])

3.9.2.3 Command MAC without Confidentiality

Figure 3-7 describes how to compute the C-MAC and how to format the APDU command.

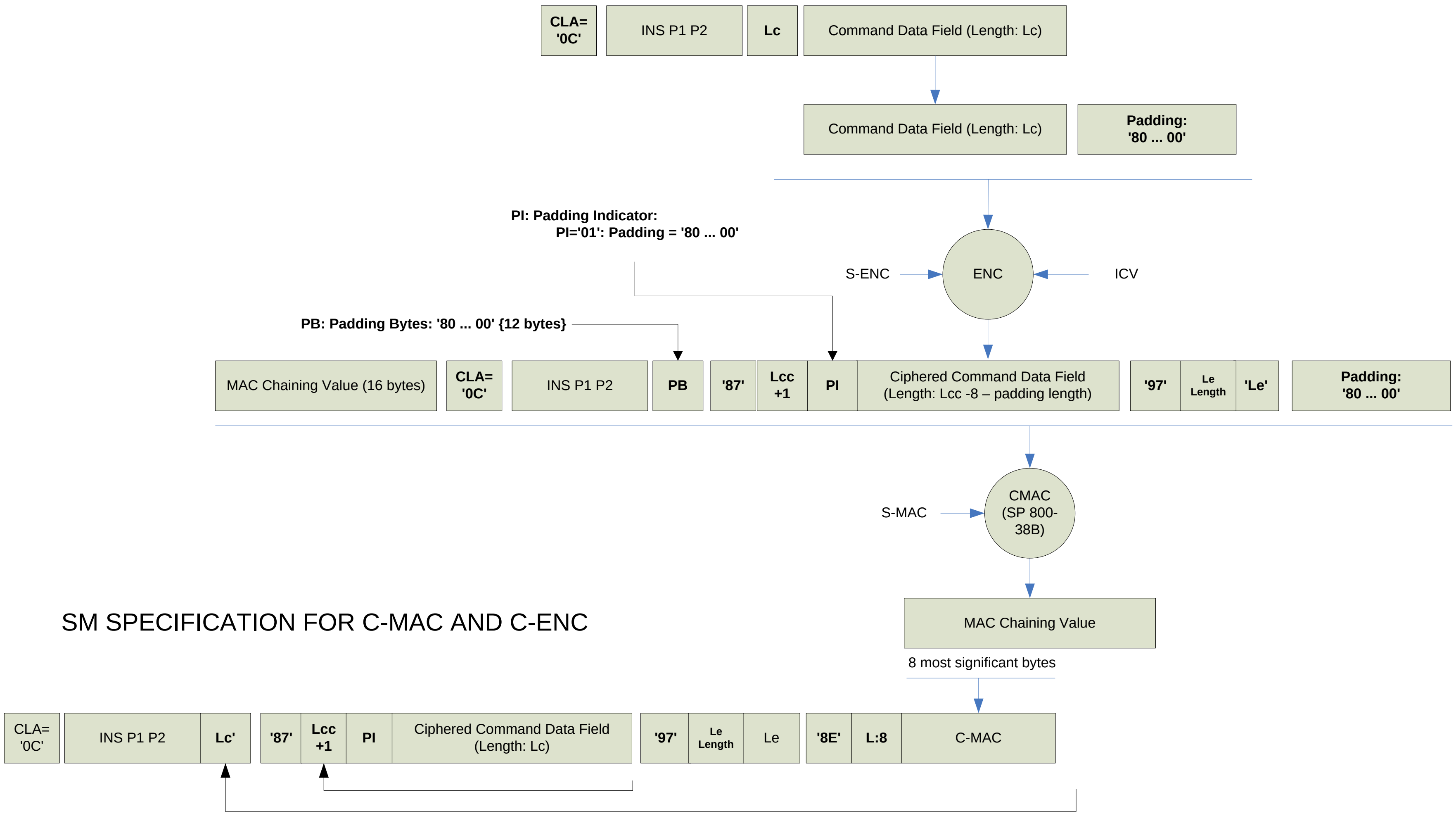
Figure 3-7: Command MAC without Confidentiality



3.9.2.4 Command MAC with Confidentiality

Figure 3-8 describes how to compute the C-MAC and C-ENC, and how to format the APDU command.

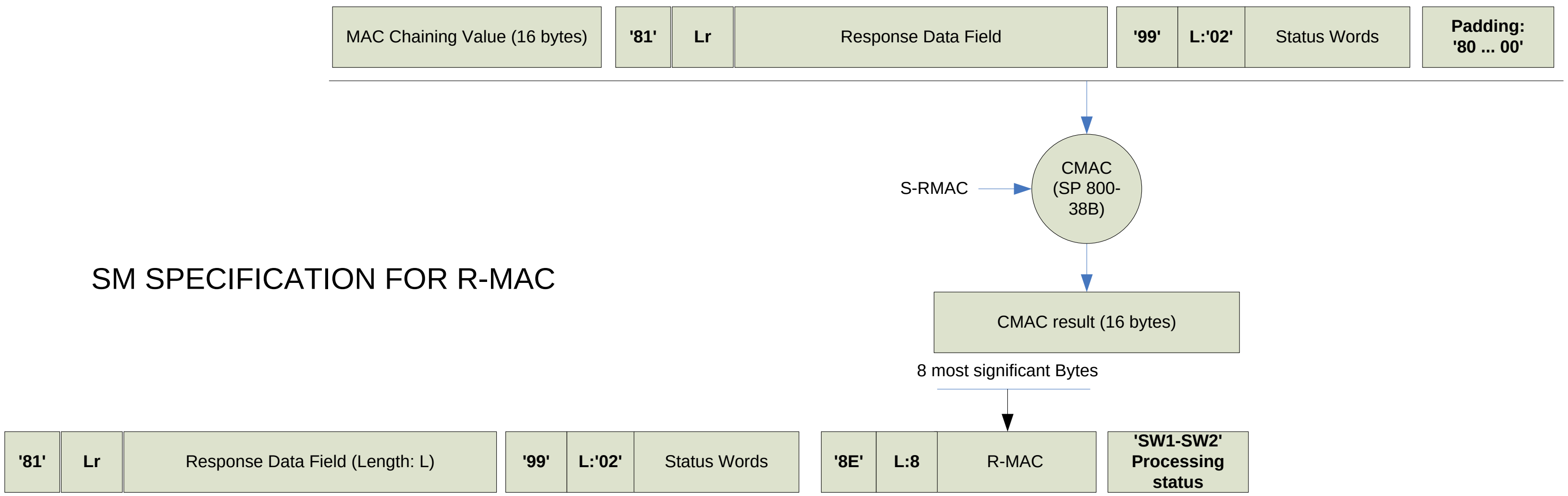
Figure 3-8: Command MAC with Confidentiality


CLA= '0C'
INS P1 P2
Lc
Command Data Field (Length: Lc)
Command Data Field (Length: Lc)
Padding: '80 ... 00'
PI: Padding Indicator:
PI='01': Padding = '80 ... 00'
S-ENC
ENC
ICV
PB: Padding Bytes: '80 ... 00' {12 bytes}
MAC Chaining Value (16 bytes)
CLA= '0C'
INS P1 P2
PB
'87'
Lcc +1
PI
Ciphered Command Data Field (Length: Lcc -8 – padding length)
'97'
Le Length
'Le'
Padding: '80 ... 00'
S-MAC
CMAC (SP 800-38B)
SM SPECIFICATION FOR C-MAC AND C-ENC
MAC Chaining Value
8 most significant bytes
CLA= '0C'
INS P1 P2
Lc'
'87'
Lcc +1
PI
Ciphered Command Data Field (Length: Lc)
'97'
Le Length
Le
'8E'
L:8
C-MAC

3.9.2.5 Response MAC without Confidentiality

Figure 3-9 describes how to compute the R-MAC and how to format the APDU command.

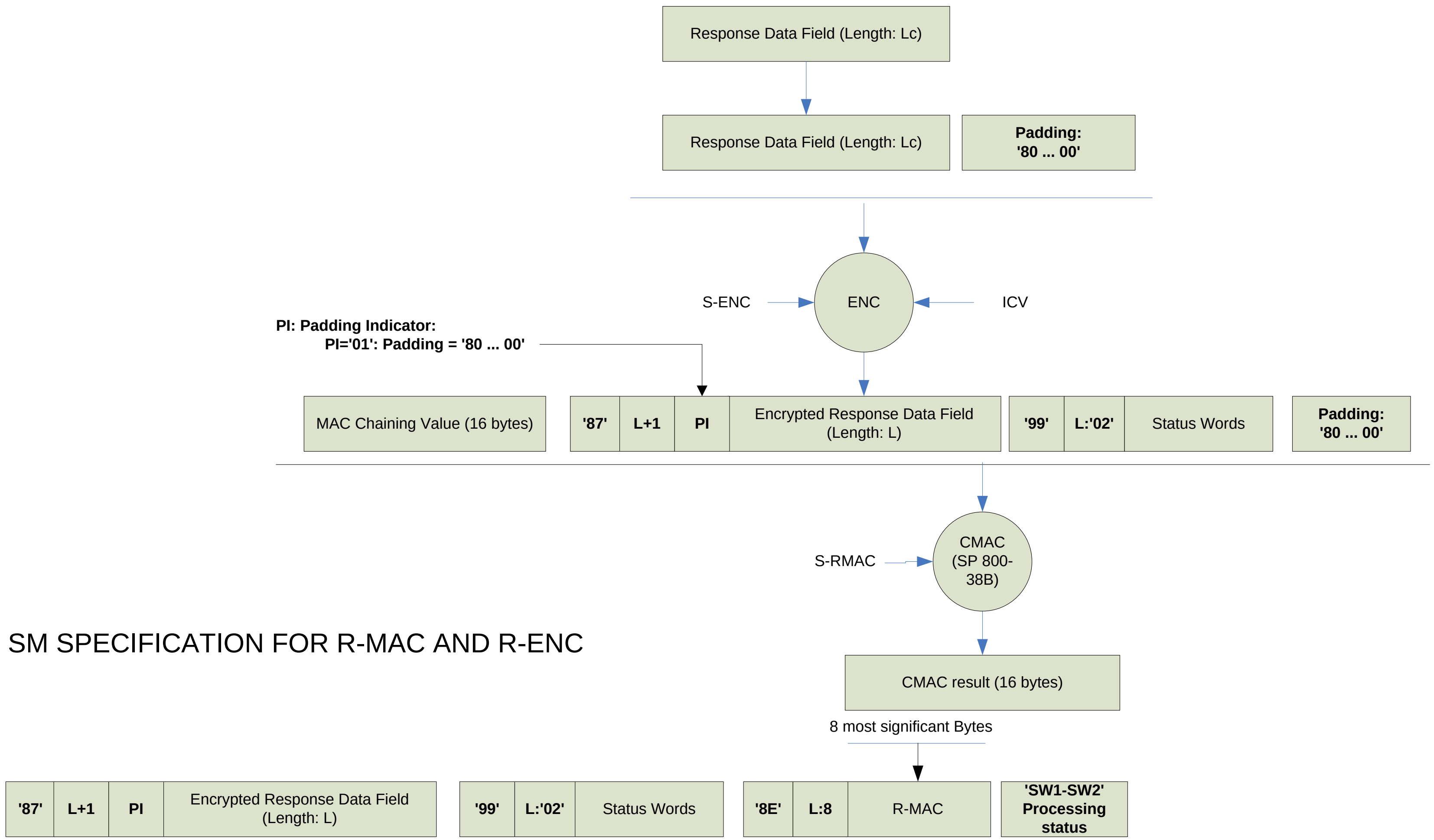
Figure 3-9: Response MAC without Confidentiality



3.9.2.6 Response MAC with Confidentiality

Figure 3-10 describes how to compute the R-MAC and R-ENC, and how to format the APDU command.

Figure 3-10: Response MAC with Confidentiality


Response Data Field (Length: Lc)
Response Data Field (Length: Lc)
Padding: '80 ... 00'
S-ENC
ENC
ICV
PI: Padding Indicator:
PI='01': Padding = '80 ... 00'
MAC Chaining Value (16 bytes)
'87'
L+1
PI
Encrypted Response Data Field (Length: L)
'99'
L:'02'
Status Words
Padding: '80 ... 00'
S-RMAC
CMAC (SP 800-38B)
SM SPECIFICATION FOR R-MAC AND R-ENC
CMAC result (16 bytes)
8 most significant Bytes
'87'
L+1
PI
Encrypted Response Data Field (Length: L)
'99'
L:'02'
Status Words
'8E'
L:8
R-MAC
'SW1-SW2' Processing status

3.9.2.7 MAC Chaining Mechanism

The MAC chaining mechanism is described in Figure 6-3 of [Amd D].

3.10 Command Chaining

This section describes how command chaining may be used to convey long command and response data fields.

Command chaining shall be indicated by setting bit 5 of the CLASS byte of an APDU command. Response chaining shall be managed using the Status Word '61 xx' and the GET RESPONSE command.

3.10.1 GET RESPONSE Command

The GET RESPONSE command shall be managed both at the Operating System level as an ISO 7816-4 APDU command and as an application command to manage output chaining and data transmission, as described below.

T=0

In T=0, the first '61 xx' Status Word shall be seen as a protocol Status Word, directly sent by the Operating System, to complete a case 2 or a case 4 command.

The associated GET RESPONSE command is also managed by the Operating System, and the application neither receives nor manages this APDU.

The next '61 xx' Status Word shall be managed by the application and the associated GET RESPONSE command is an application command.

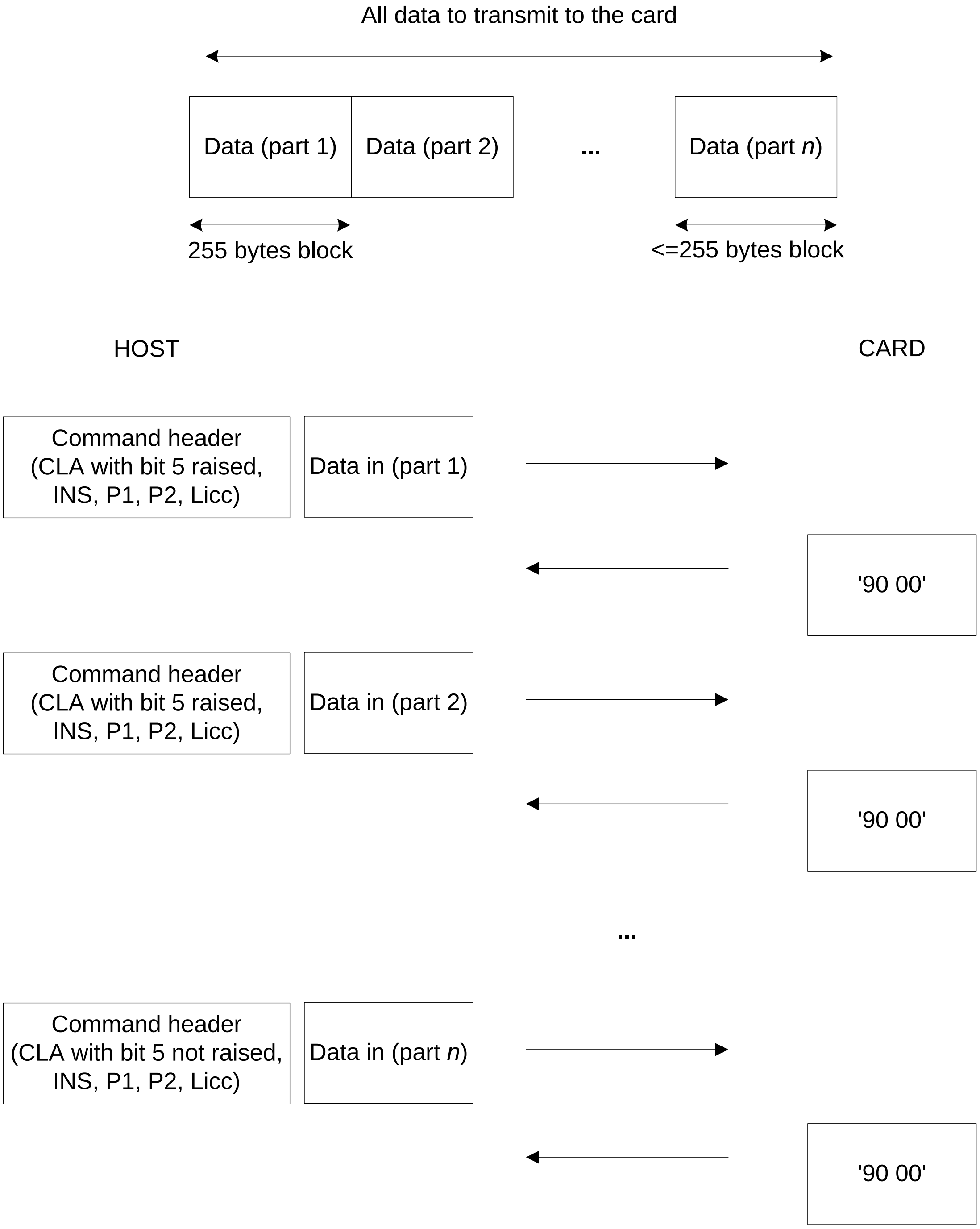
T=1 and T=CL

The T=1 and T=CL protocols don’t need a '61 xx' Status Word to send the first part of the data, so each '61 xx' Status Word shall be sent by the application, and GET RESPONSE commands are received and managed by the application.

3.10.2 Input Chaining (T=0 & T=1)

The input chaining is defined by using bit 5 of the CLASS byte.

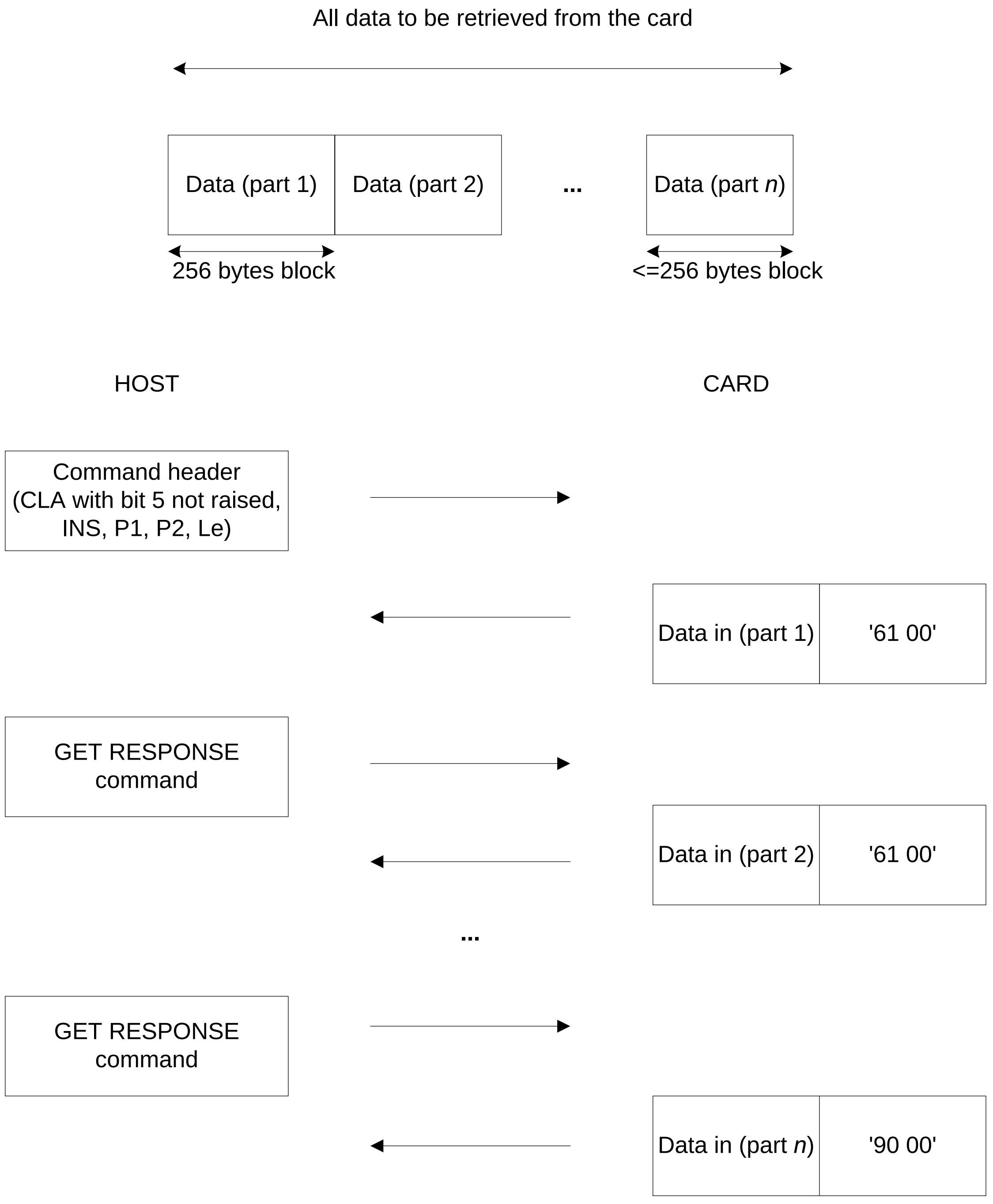
Figure 3-11: Input Command Chaining (T=0 & T=1) for Short APDUs


All data to transmit to the card
Data (part 1)
Data (part 2)
...
Data (part *n*)
255 bytes block
<=255 bytes block
HOST
CARD
Command header (CLA with bit 5 raised, INS, P1, P2, Licc)
Data in (part 1)
'90 00'
Command header (CLA with bit 5 raised, INS, P1, P2, Licc)
Data in (part 2)
'90 00'
...
Command header (CLA with bit 5 not raised, INS, P1, P2, Licc)
Data in (part *n*)
'90 00'

3.10.3 Output Chaining T=0

The output command chaining is managed using the Status Word '61 xx' and the GET RESPONSE command.

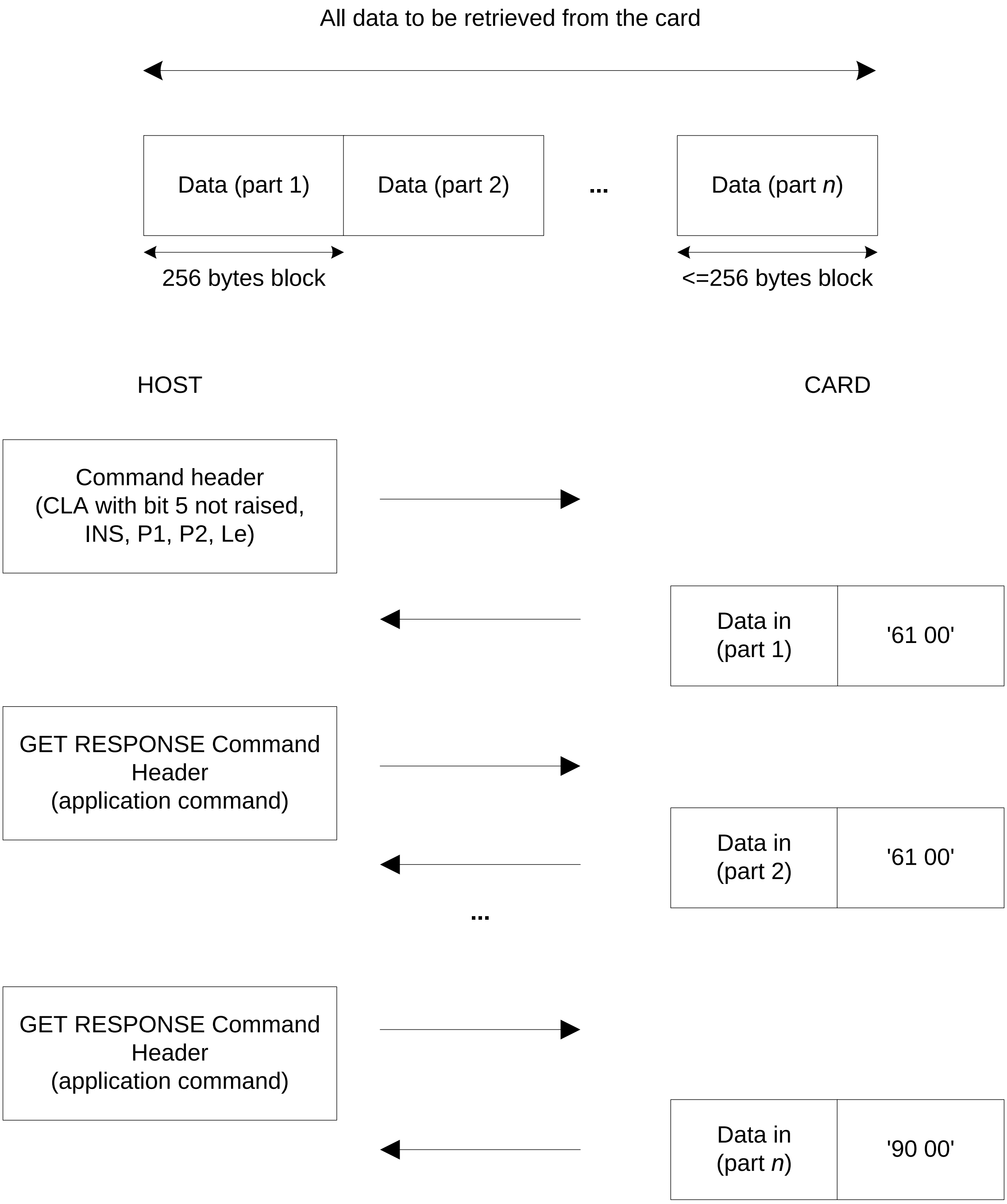
Figure 3-12: Output Command Chaining Example T=0 for Short APDUs



3.10.4 Output Chaining T=1

The output command chaining is managed by the Status Word '61 xx' and the GET RESPONSE command.

Figure 3-13: Output Command Chaining Example T=1 for Short APDUs

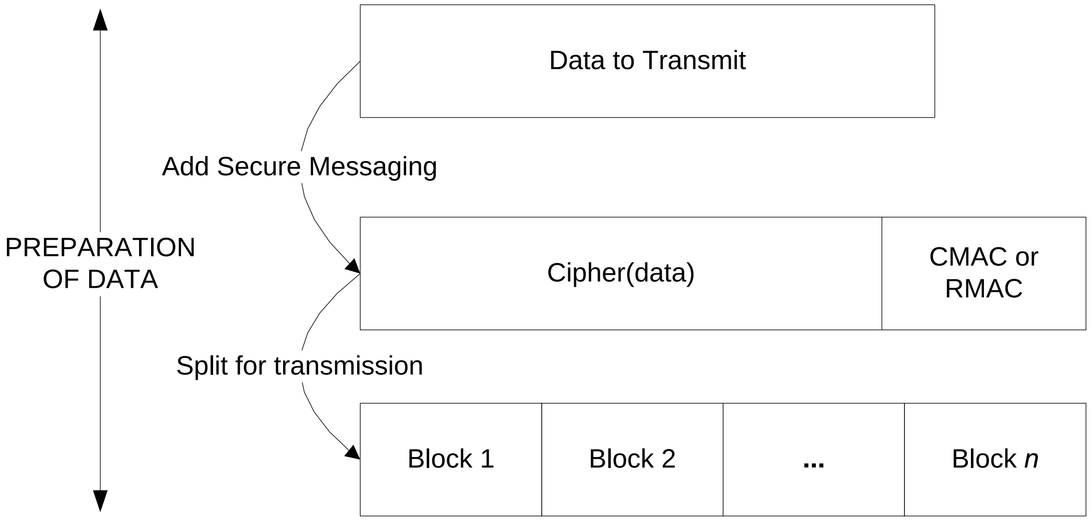


3.10.5 Secure Messaging and Command Chaining

3.10.5.1 General Mechanism

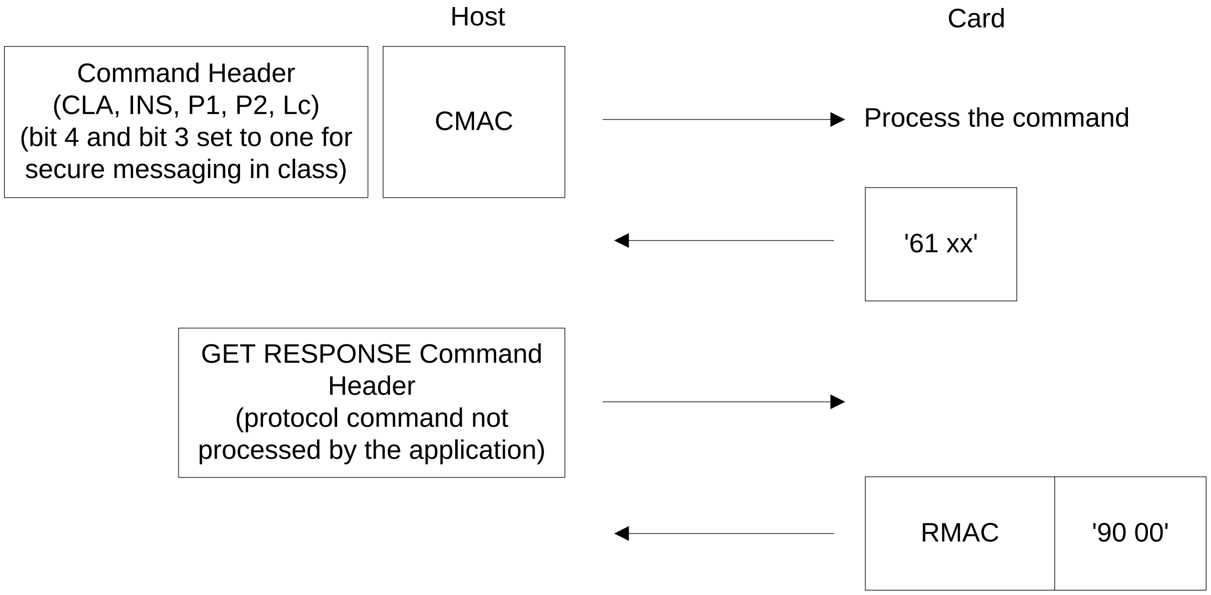
When input or output command chaining is used, the secure messaging is computed on the entire message before command fragmentation for data transportation.

Figure 3-14: Secure Messaging Computation



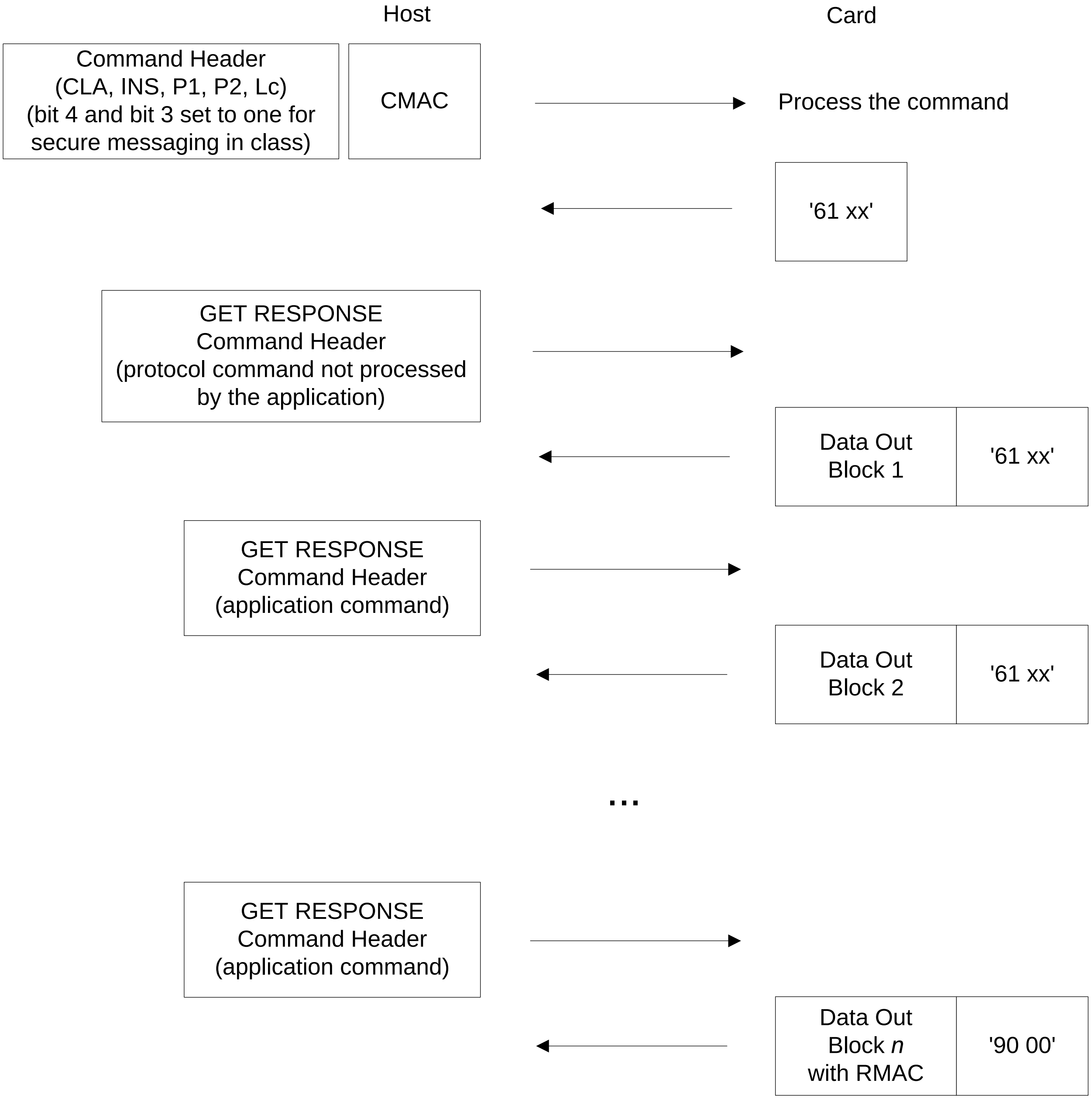
3.10.5.2 T=0, Case 1, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-15: Secure Messaging and Command Chaining in T=0, Case 1


Host
Card
Command Header (CLA, INS, P1, P2, Lc) (bit 4 and bit 3 set to one for secure messaging in class)
CMAC
Process the command
'61 xx'
GET RESPONSE Command Header (protocol command not processed by the application)
RMAC
'90 00'

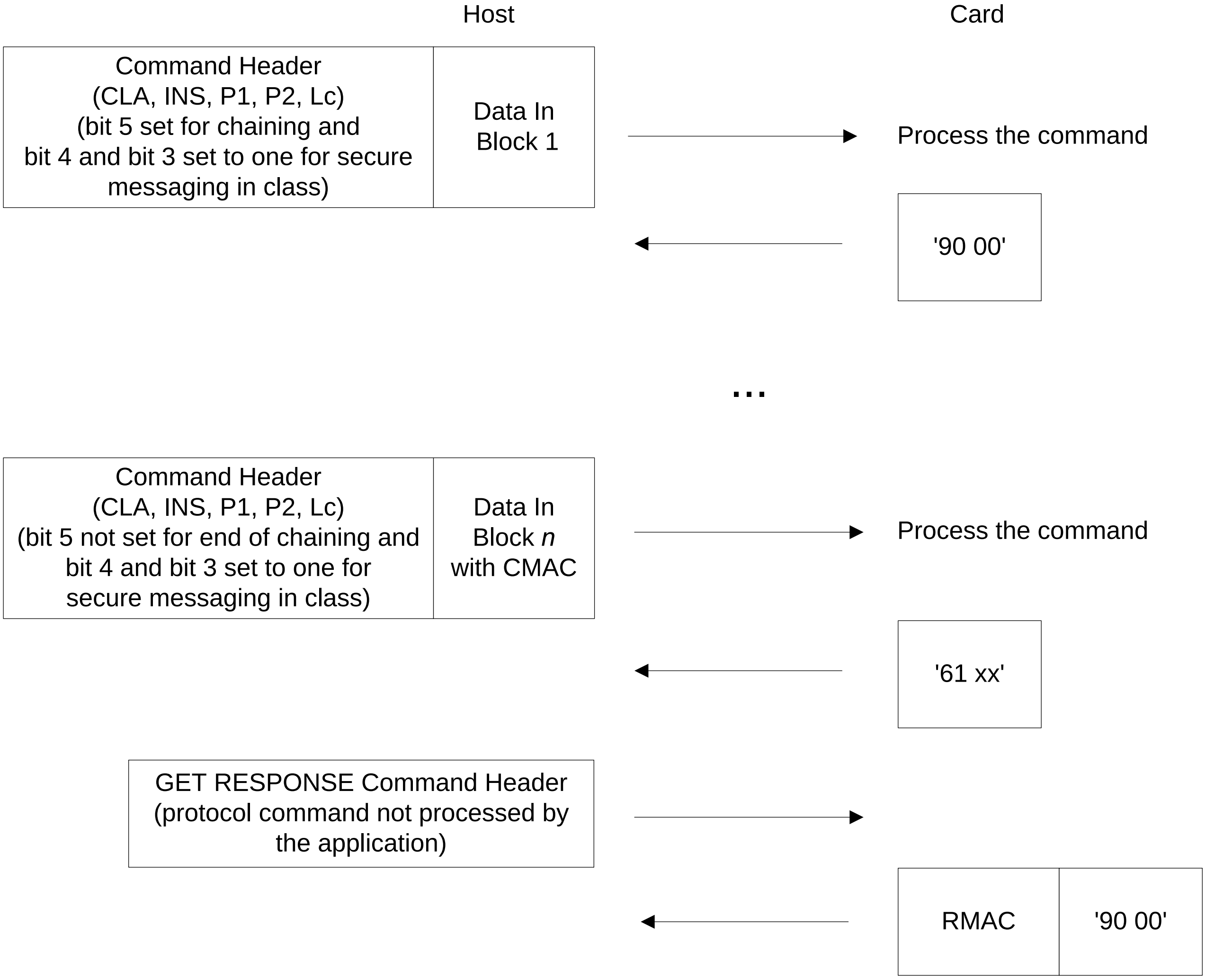
3.10.5.3 T=0, Case 2, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-16: Secure Messaging and Command Chaining in T=0, Case 2



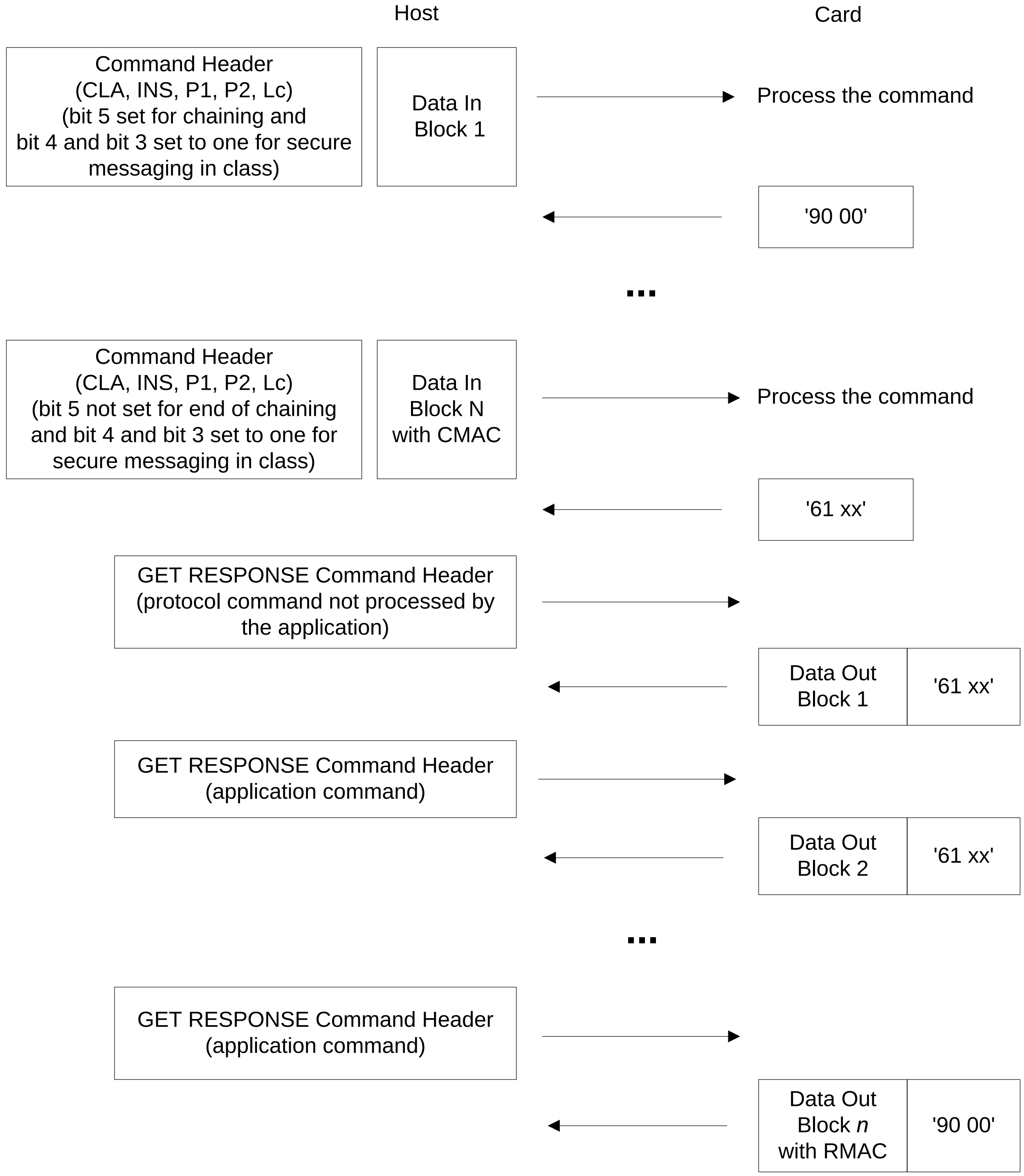
3.10.5.4 T=0, Case 3, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-17: Secure Messaging and Command Chaining in T=0, Case 3


Host
Card
Command Header
(CLA, INS, P1, P2, Lc)
(bit 5 set for chaining and
bit 4 and bit 3 set to one for secure
messaging in class)
Data In
Block 1
Process the command
'90 00'
…
Command Header
(CLA, INS, P1, P2, Lc)
(bit 5 not set for end of chaining and
bit 4 and bit 3 set to one for
secure messaging in class)
Data In
Block *n*
with CMAC
Process the command
'61 xx'
GET RESPONSE Command Header
(protocol command not processed by
the application)
RMAC
'90 00'

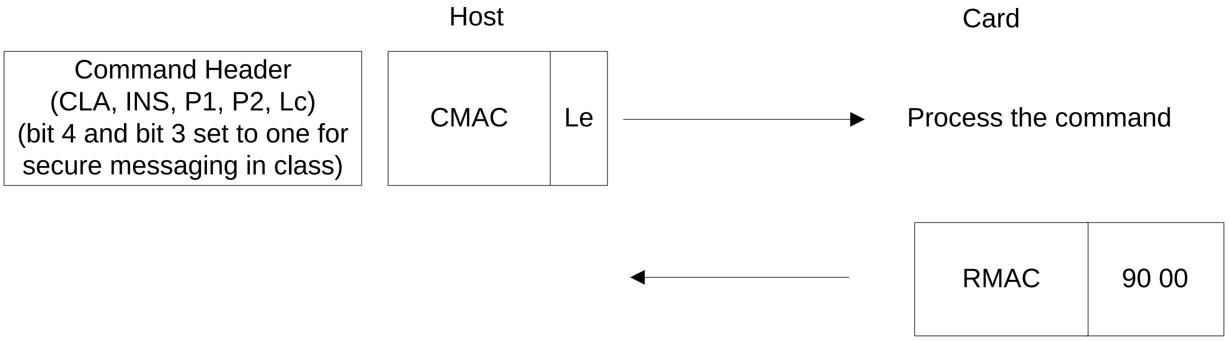
3.10.5.5 T=0, Case 4, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-18: Secure Messaging and Command Chaining in T=0, Case 4


Host
Card
Command Header (CLA, INS, P1, P2, Lc) (bit 5 set for chaining and bit 4 and bit 3 set to one for secure messaging in class)
Data In Block 1
Process the command
'90 00'
...
Command Header (CLA, INS, P1, P2, Lc) (bit 5 not set for end of chaining and bit 4 and bit 3 set to one for secure messaging in class)
Data In Block N with CMAC
Process the command
'61 xx'
GET RESPONSE Command Header (protocol command not processed by the application)
Data Out Block 1
'61 xx'
GET RESPONSE Command Header (application command)
Data Out Block 2
'61 xx'
...
GET RESPONSE Command Header (application command)
Data Out Block *n* with RMAC
'90 00'

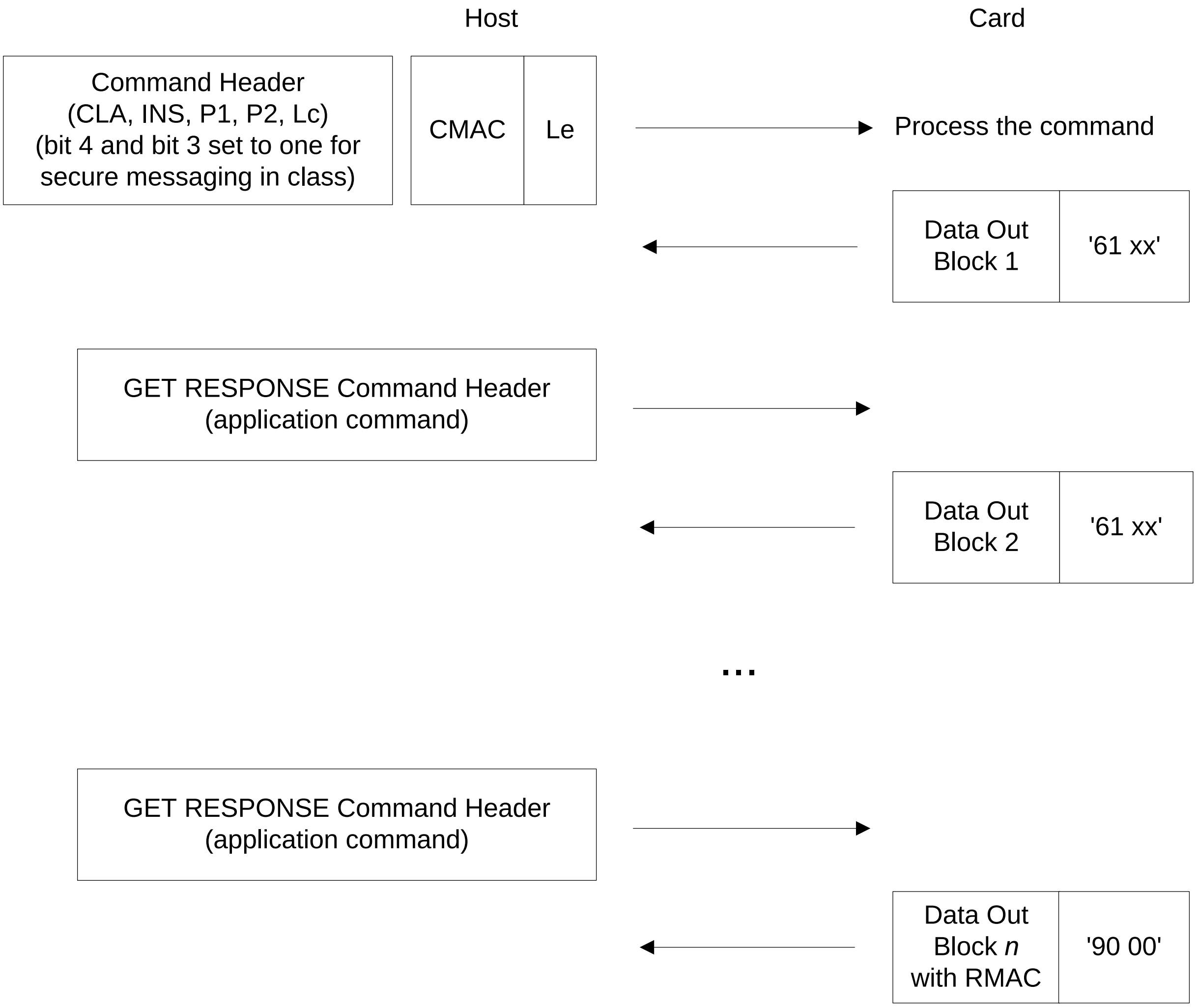
3.10.5.6 T=1, Case 1, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-19: Secure Messaging and Command Chaining in T=1, Case 1



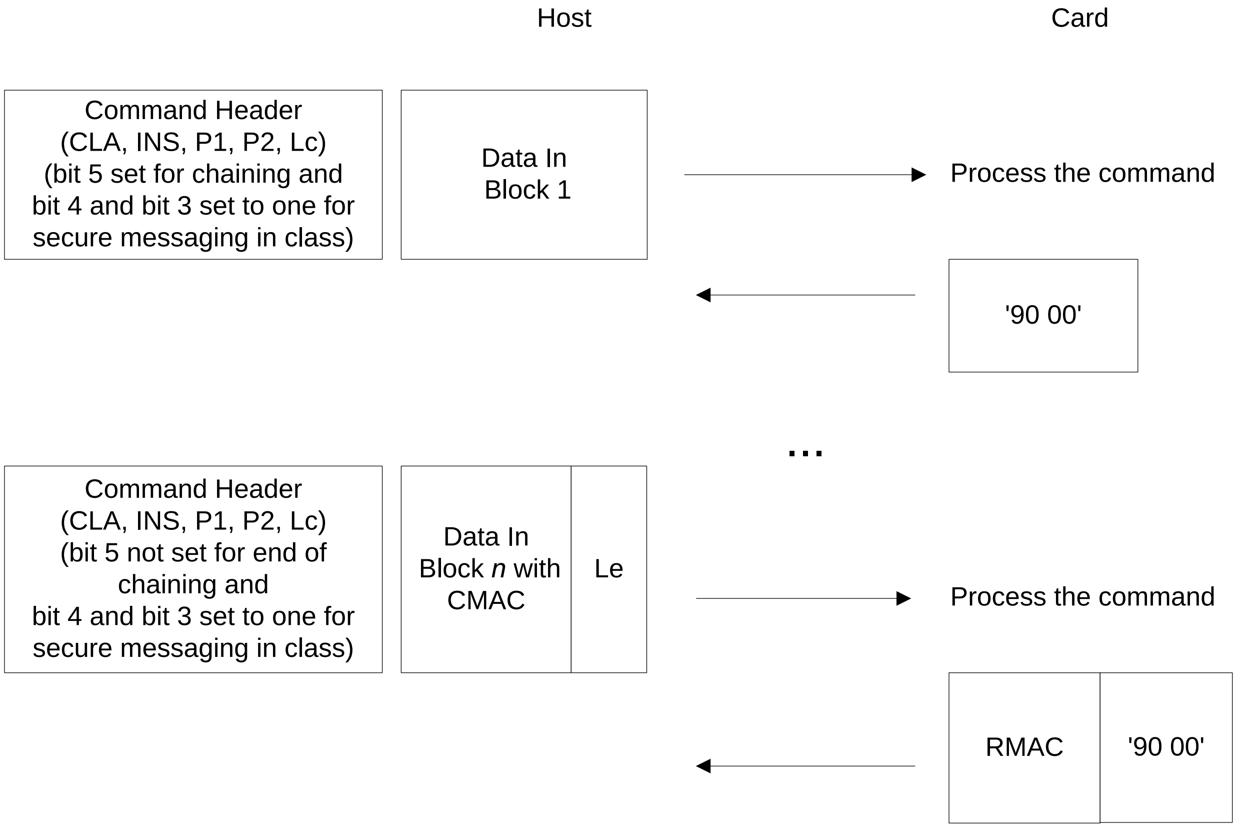
3.10.5.7 T=1, Case 2, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-20: Secure Messaging and Command Chaining in T=1, Case 2



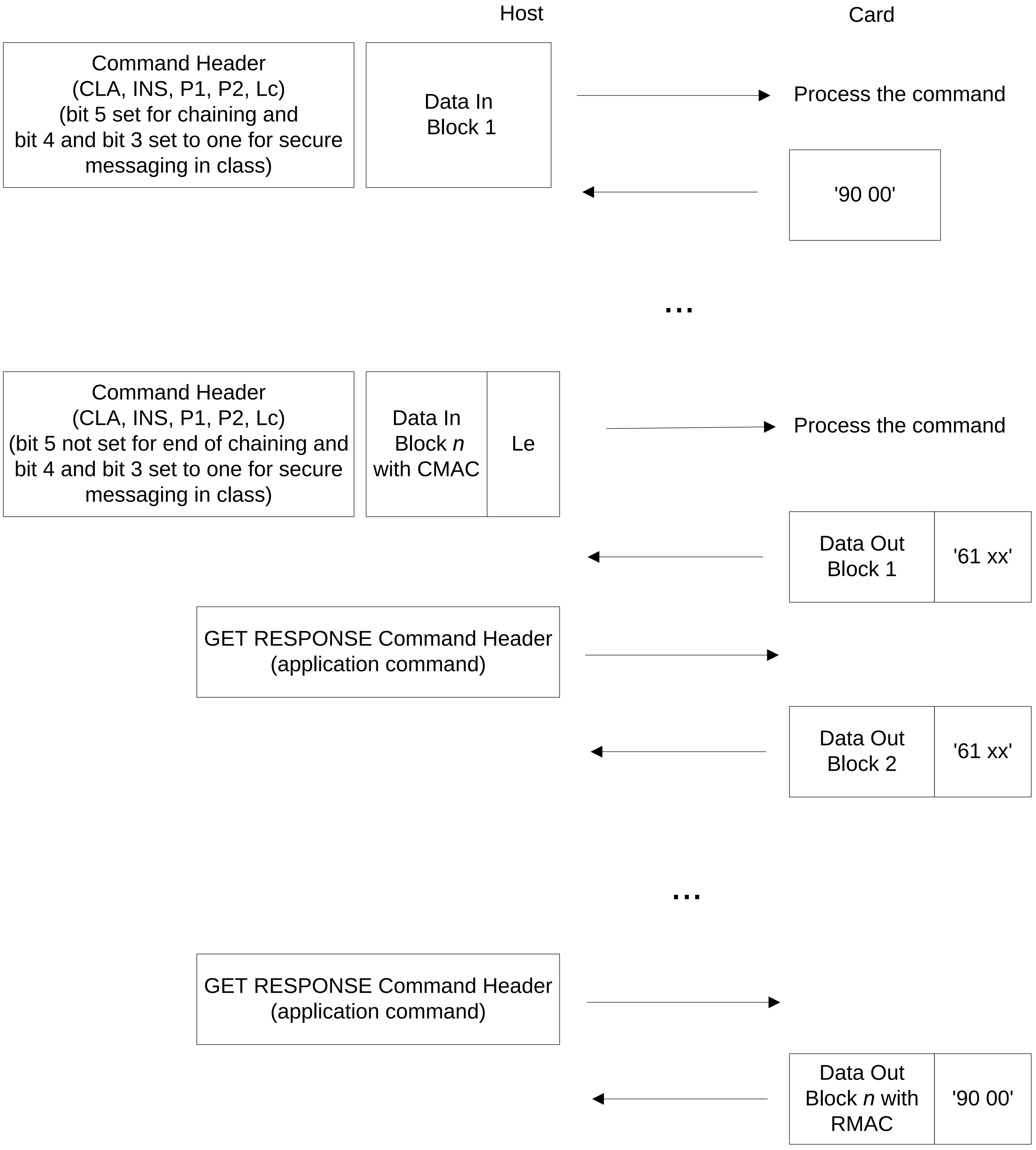
3.10.5.8 T=1, Case 3, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-21: Secure Messaging and Command Chaining in T=1, Case 3



3.10.5.9 T=1, Case 4, with C_MAC, C_DECRYPTION, R_MAC, and R_ENC

Figure 3-22: Secure Messaging and Command Chaining in T=1, Case 4


Host
Card
Command Header (CLA, INS, P1, P2, Lc) (bit 5 set for chaining and bit 4 and bit 3 set to one for secure messaging in class)
Data In Block 1
Process the command
'90 00'
...
Command Header (CLA, INS, P1, P2, Lc) (bit 5 not set for end of chaining and bit 4 and bit 3 set to one for secure messaging in class)
Data In Block *n* with CMAC
Le
Process the command
Data Out Block 1
'61 xx'
GET RESPONSE Command Header (application command)
Data Out Block 2
'61 xx'
...
GET RESPONSE Command Header (application command)
Data Out Block *n* with RMAC
'90 00'

4 Applications and Card Content Management

4.1 Introduction

Card content management commands are used to load, install, and remove applications on the card.

These operations shall be performed only by the ISO-SD.

Card content management commands shall be accepted only if the ISO-SD has been successfully authenticated.

4.2 Application Dedicated File Template

An ADF is a structure hosting an application on the ISO-SD hierarchy. An ADF shall be defined as a named structure that contains control parameters, BER-TLV objects, Security Objects, and elementary files.

The Control Parameter Template (tag '62') associated with an application ADF shall contain Data Objects with tags listed in Table 4-1.

Table 4-1: Control Parameter Template for ADF

<table>
<tr><th>Tag</th><th>Length</th><th colspan="5">Description</th><th>Presence</th></tr>
<tr><td rowspan="14">'62'</td><td rowspan="14">Var</td><td colspan="5">Control Parameter Template</td><td>Mandatory</td></tr>
<tr><th>Tag</th><th>Len</th><th colspan="3">Description</th><th></th></tr>
<tr><td>'82'</td><td>1</td><td colspan="3">File Descriptor Byte</td><td>Mandatory</td></tr>
<tr><td>'84'</td><td>5-16</td><td colspan="3">Application Dedicated File Name</td><td>Mandatory</td></tr>
<tr><td>'87'</td><td>2</td><td colspan="3">Identifier of an EF containing an extension of the File Control Information</td><td>Optional</td></tr>
<tr><td>'8A'</td><td>1</td><td colspan="3">ISO Life Cycle Status byte (read only attribute)</td><td>Mandatory</td></tr>
<tr><td rowspan="4">'A3'</td><td rowspan="4">Var</td><td colspan="3">Security Attribute Template</td><td>Optional</td></tr>
<tr><th>Tag</th><th>Len</th><th>Description</th><th></th></tr>
<tr><td>'91'</td><td>1</td><td>Physical Interface Byte</td><td>Mandatory</td></tr>
<tr><td>'9C'</td><td>Var</td><td>Security Attribute in Compact Format</td><td>Mandatory</td></tr>
<tr><th>Tag</th><th>Len</th><th colspan="3">Description</th><th></th></tr>
<tr><td rowspan="3">'AC'</td><td rowspan="3">Var</td><td colspan="3">Cryptographic Mechanism Identifier Template</td><td>Optional</td></tr>
<tr><th>Tag</th><th>Len</th><th>Description</th><th></th></tr>
<tr><td>'80'</td><td>1</td><td>Cryptographic Mechanism Reference (See Table 2-2.)</td><td>Mandatory</td></tr>
</table>

The File Descriptor Byte for an ADF shall be '38'.

Each card application is identified by its Application Dedicated File Name.

An Application Dedicated File Name corresponds to an Application Identifier (AID) for Java Card and GlobalPlatform specifications.

A card that supports the ISO-SD command set has at least one and possibly several ADF.

Only one level of application is allowed in an ISO-SD hierarchy.

An ADF can be selected only by its Application Dedicated File Name, which can be discovered by using EF.DIR.

The files rooted at an ADF shall contain all the Data Objects encoding the data elements of the application.

Application Data Objects are stored in the ADF or in the EFs under the ADF. Files are readable using an odd instruction (INS) byte form of the GET DATA command.

The cryptographic mechanisms available for use when an ADF is currently selected shall be described in the Cryptographic Mechanism Identifier Template (tag 'AC') in the Control Parameter Template (tag '62') of the ADF.

4.3 Application Capability Descriptor (Tag '7F 63')

To facilitate the integration of a card supporting the ISO-SD command set with ISO/IEC 24727 middleware, the Application Capability Descriptor (ACD) is retrievable from each application in the ISO-SD hierarchy.

The ACD Data Object shall be stored at the ADF level.

The ACD of a selected card application can be retrieved at any time with a GET DATA command with P1-P2 = '3F FF' and a command data field that contains '5C 02 7F 63'.

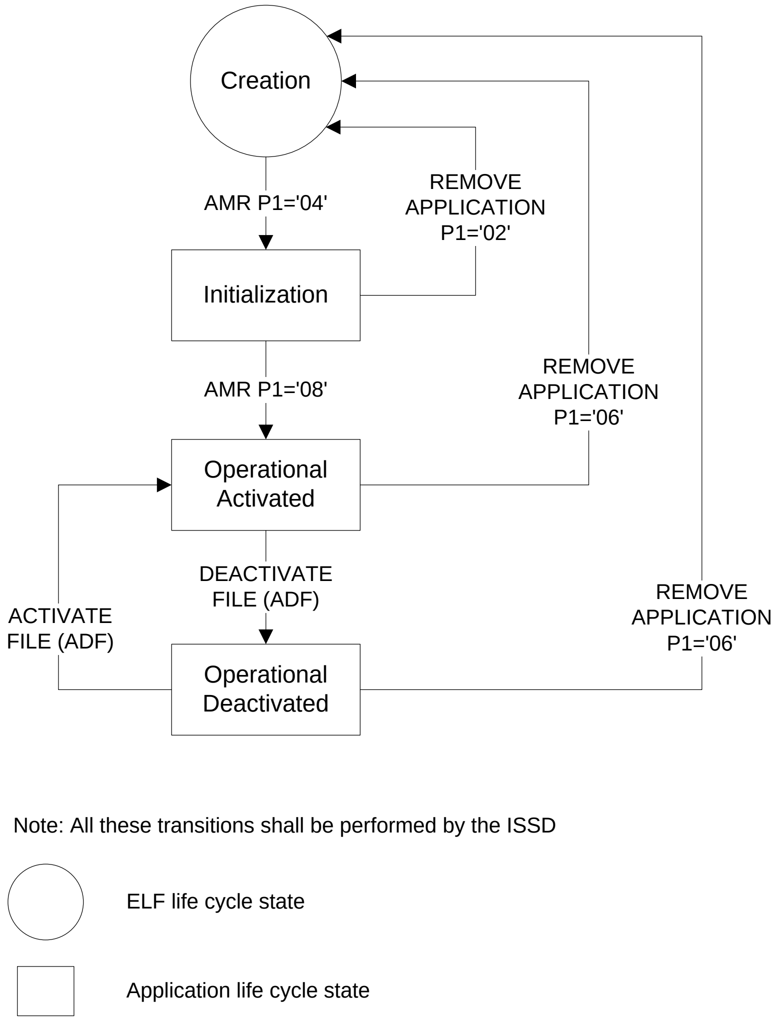
The content of the ACD is defined by [24727-2] and no optional tag is used in this template for the ISO-SD.

Table 4-2: Application Capability Descriptor for ISO-SD (Tag '7F 63')

Tag	Length	Name	Description	Presence
'7F 63'	Variable	Application Capability Descriptor	See [24727-2], section 6.1.	Mandatory

4.4 Application Life Cycle State

Figure 4-1: Application Life Cycle State Transitions



Note: All these transitions shall be performed by the ISSD

Table 4-3 shows the mapping between ISO states and GlobalPlatform states.

Table 4-3: Life Cycle Mapping Between ISO and GlobalPlatform for Applications

ISO Value	ISO State	GlobalPlatform State	GlobalPlatform Value
'01'	Creation[1]	LOADED but not yet INSTALLED	'01'
'03'	Initialization	SELECTABLE	'07'
'05'	Operational Activated	Application specific state[2][3]	'0F'
'04'	Operational Deactivated	LOCKED	bit 8 set to one

(1) The Creation state is not exposed at the interface level. It is an abstract state.

(2) The ISO-SD is able to perform the transition of an application from selectable to an application specific state '0F'.

(3) Any application specific state value shall be mapped onto the Operational Activated ISO state.

4.5 ISO-SD APDU Commands for Card Content Management

Table 3-8 on page 46 provides a mapping between the ISO APDUs and the GlobalPlatform APDUs. This section describes the following ISO-SD APDU commands:

- Application Management Request Command
- Load Application Command
- Remove Application Command
- Activate File (ADF) Command
- Deactivate File (ADF) Command

The remaining ISO-SD commands are described in section 3.8.

4.5.1 APPLICATION MANAGEMENT REQUEST (AMR) Command

4.5.1.1 Description

The APPLICATION MANAGEMENT REQUEST command executes the life cycle transition procedure for an application. The ISO-SD application verifies the application management request information present in the command data field. This command may be followed by the LOAD APPLICATION command.

4.5.1.2 Condition of Usage

This command shall be issued in a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

4.5.1.3 APDU Command

Table 4-4: APPLICATION MANAGEMENT REQUEST (AMR) Command

CLA	See section 2.1.
INS	'40'
P1	Application Life Cycle Status Control P1
P2	'00' (No Information Given)
Lc	Variable
Command Data Field	Application management request information
Le	Variable
Response Data Field	Additional information

4.5.1.4 Reference Control Parameter P1: Application Life Cycle Status Control

Table 4-5: APPLICATION MANAGEMENT REQUEST P1

b8	b7	b6	b5	b4	b3	b2	b1	Description	Mapping to GlobalPlatform Command
0	0	0	0	0	0	1	0	Transition from application nonexistent to Creation state	INSTALL [for load] (P1='02') + LOAD
0	0	0	0	0	1	0	0	Transition from Creation state to Initialization state	INSTALL [for install and make selectable] (P1='0C')
0	0	0	0	0	1	1	0	Transition from application nonexistent to Initialization state	INSTALL [for load, install and make selectable] + LOAD
0	0	0	0	1	0	0	0	Transition from Initialization state to Operational Activated state	SET STATUS (P1='40', P2='0F')
0	0	0	0	1	1	0	0	Transition from Creation state to Operational Activated state	INSTALL [for install and make selectable] (P1='0C') + SET STATUS (P1='40', P2='0F')
0	0	0	0	1	1	1	0	Transition from application nonexistent to Operational Activated state	INSTALL [for load, install and make selectable] + LOAD + SET STATUS (P1='40', P2='0F')

Note: According to GlobalPlatform, SET STATUS P2='0F' shall be performed by an application and not by a Security Domain.

4.5.1.5 Reference Control Parameter P2

P2 shall be equal to '00', which indicates “No information given” (defined in [7816-13]).

4.5.1.6 Command Data Field

The command data field includes the Data Objects listed in Table 4-6.

Table 4-6: APPLICATION MANAGEMENT REQUEST Command Data Field

<table>
<tr><th>Tag</th><th>Len</th><th colspan="3">Value</th><th>Presence</th></tr>
<tr><td rowspan="3">'61'</td><td rowspan="3">Var</td><td colspan="3">Application Template
This value is application dependent.</td><td></td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th>Presence</th></tr>
<tr><td>'4F'</td><td>5-16</td><td>Application AID</td><td>Mandatory for:
• Transition from Creation state to Initialization state
• Transition from application nonexistent to Initialization state
• Transition from Creation state to Operational Activated state
• Transition from Initialization state to Operational Activated state
• Transition from application nonexistent to Operational Activated state</td></tr>
<tr><td>'62'</td><td>Var</td><td colspan="3">Control Parameter (CP) Template
This value is application dependent.</td><td>Mandatory if P1='0C' or P1='04'</td></tr>
<tr><td rowspan="4">'64'</td><td rowspan="4">Var</td><td colspan="3">File Management Data (FMD)
These values are application dependent.</td><td>Mandatory if P1='0C' or P1='04'</td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th>Presence</th></tr>
<tr><td>'7F 69'</td><td>Var</td><td>Identity of Executable Module AID
<b>Note:</b> The Executable Module AID is composed of the Elementary Load File AID plus two bytes for Executable Module version (Major and Minor).</td><td>Mandatory if P1='0C' or P1='04'</td></tr>
<tr><td>'53'</td><td>Var</td><td>Discretionary Data that contains the Executable Load File AID.</td><td>Optional. If not present, an implicit ELF AID shall be used</td></tr>
<tr><td rowspan="3">'7F 3D'</td><td rowspan="3">Var</td><td colspan="3">One or more signature data blocks</td><td>Conditional</td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th>Presence</th></tr>
<tr><td>'9E'</td><td>Var</td><td>Token (for computation description, see [GPCS] section C.4)</td><td>Conditional</td></tr>
</table>

Tags '61', '62', and '64' may contain additional tags that are application dependent.

4.5.1.7 Response Data Field

Table 4-7: APPLICATION MANAGEMENT REQUEST Response Data Field

Length	Data Element	Presence
1-2	Length of Receipt ('00' – '7F' or '81 80' – '81 FF')	Mandatory
Variable	Receipt (for computation description, see [GPCS] section C.5)	Conditional

4.5.1.8 Status Word

Table 4-8: APPLICATION MANAGEMENT REQUEST Status Word

SW1 SW2	Description
'69 82'	Security status not satisfied
'69 85'	Conditions of use not satisfied
'90 00'	Successful execution

4.5.2 LOAD APPLICATION Command

4.5.2.1 Description

The LOAD APPLICATION command transfers an application to the card.

This command is optional and not all smart card are mandated to support it.

An application may be partitioned into multiple components and each component may be partitioned into multiple blocks for transmission to the card. Each LOAD APPLICATION command transfers one block to the card.

An APPLICATION MANAGEMENT REQUEST command shall immediately precede the first LOAD APPLICATION command.

The target application name is specified in the preceding APPLICATION MANAGEMENT REQUEST.

Memory resource assignment and setting of the application life cycle state to an appropriate value is done on the basis of information provided by the sequence of LOAD APPLICATION commands.

4.5.2.2 Condition of Usage

An APPLICATION MANAGEMENT REQUEST command with P1='02' (Transition from application nonexistent to Creation state) shall be issued immediately before the first LOAD APPLICATION command.

That means that this command shall be issued in a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

4.5.2.3 APDU Command

Table 4-9: LOAD APPLICATION Command

CLA	See section 2.1.
INS	'EA'
P1	'00'
P2	'00'
Lc	Variable (number of bytes in the command data field)
Command Data Field	Load Block
Le	Variable
Response Data Field	Additional information

4.5.2.4 Reference Control Parameters P1 P2

P1 and P2 shall be equal to '00'.

An Elementary Load File (ELF) may be partitioned into multiple components and each component may be partitioned into multiple blocks for transmission to the card. Each LOAD APPLICATION command transfers one block to the card.

In that case, it is not possible to send each block of the elementary load file in a random order and bit 5 of the CLASS byte is used as for regular command chaining mechanism.

In case of a bad loading, the loading is aborted.

In case of reception of a command in between two LOAD APPLICATION commands, the loading is aborted.

If loading is aborted, then the card shall not keep any information/data of the previous LOAD APPLICATION command and the memory shall be recovered.

4.5.2.5 Command Data Field

LOAD APPLICATION includes a Load Block in the data field, with length specified by Lc.

Table 4-10: LOAD APPLICATION Command Data Field

<table>
<tr><th>Tag</th><th>Length</th><th colspan="3">Value</th><th>Presence</th></tr>
<tr><td>'C4'</td><td>Var</td><td colspan="3">Elementary Load File</td><td>Mandatory</td></tr>
<tr><td rowspan="3">'7F 3D'</td><td rowspan="3">Var</td><td colspan="3">One or more signature data blocks</td><td>Conditional</td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th>Presence</th></tr>
<tr><td>'9E'</td><td>Var</td><td>Load Token (for computation description, see [GPCS] section C.4)</td><td>Conditional</td></tr>
</table>

4.5.2.6 Response Data Field

Table 4-11: LOAD APPLICATION Response Data Field

Length	Data Element	Presence
1-2	Length of Receipt ('00' – '7F' or '81 80' – '81 FF')	Mandatory
Var	Receipt (for computation description, see [GPCS] section C.5)	Conditional

If the LOAD APPLICATION command does not contain the last block in the sequence, a single byte of '00' shall be returned indicating that no additional data is present.

For a LOAD APPLICATION command containing the last block in the sequence being issued to a Security Domain with the Delegated Management privilege, the data field may contain the confirmation of the load procedure.

4.5.2.7 Status Word

Table 4-12: LOAD APPLICATION Status Word

SW1 SW2	Description
'65 81'	Memory failure
'69 82'	Security status not satisfied
'69 85'	Conditions of use not satisfied
'6A 84'	Memory full
'90 00'	Successful execution

4.5.3 REMOVE APPLICATION Command

4.5.3.1 Description

The REMOVE APPLICATION command shall delete an application.

The card manager application (ISO-SD) shall be selected and verifies the application removing information, when present in the command data field.

The resources supporting the application may be reclaimed.

4.5.3.2 Condition of Usage

This command shall be issued in a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

4.5.3.3 APDU Command

Table 4-13: REMOVE APPLICATION Command

CLA	See section 2.1.
INS	'EC'
P1	Removing state control, see Table 4-14
P2	'00'
Lc	Variable (number of bytes in the command data field)
Command Data Field	Application removing information
Le	Variable
Response Data Field	Additional information

4.5.3.4 Reference Control Parameter P1

Table 4-14: REMOVE APPLICATION P1

b8	b7	b6	b5	b4	b3	b2	b1	Description
0	0	0	0	0	0	0	1	Transition from Creation state to Application Removed state
0	0	0	0	0	0	1	0	Transition from Initialization state to Creation state
0	0	0	0	0	0	1	1	Transition from Initialization state to Application Removed state
0	0	0	0	0	1	0	0	Transition from Operational (Activated or Deactivated) state to Creation state
0	0	0	0	0	1	1	1	Transition from Operational (Activated or Deactivated) state to Application Removed state

4.5.3.5 Command Data Field

Table 4-15: REMOVE APPLICATION Command Data Field

<table>
<tr><th>Tag</th><th>Length</th><th colspan="3">Value</th><th>Presence</th></tr>
<tr><td>'4F'</td><td>Var</td><td colspan="3">AID of the application to be removed</td><td>Mandatory</td></tr>
<tr><td rowspan="3">'7F 3D'</td><td rowspan="3">Var</td><td colspan="3">Digital signature</td><td>Conditional</td></tr>
<tr><th>Tag</th><th>Len</th><th>Value</th><th>Presence</th></tr>
<tr><td>'9E'</td><td>Var</td><td>Delete Token (for computation description, see [GPCS] section C.4)</td><td>Conditional</td></tr>
</table>

4.5.3.6 Response Data Field

Table 4-16: REMOVE APPLICATION Response Data Field

Length	Data Element	Presence
1-2	Length of Receipt ('00' – '7F' or '81 80' – '81 FF')	Mandatory
Var	Receipt (for computation description, see [GPCS] section C.5)	Conditional

4.5.3.7 Status Word

Table 4-17: REMOVE APPLICATION Status Word

SW1 SW2	Description
'69 85'	Conditions of use not satisfied
'6A 88'	Application not found
'90 00'	Successful execution

4.5.4 ACTIVATE FILE (ADF) Command

4.5.4.1 Description

The ACTIVATE FILE command activates the ADF specified in the command data field.

It is the equivalent of a SET STATUS command sent to unlock an application.

4.5.4.2 Condition of Usage

This command shall be performed inside a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

4.5.4.3 APDU Command

Table 4-18: ACTIVATE FILE Command

CLA	See section 2.1.
INS	'44'
P1	'04' (ADF)
P2	'00'
Lc	Variable (number of bytes in the command data field)
Command Data Field	Application Dedicated File Name
Le	Variable
Response Data Field	Additional information

4.5.4.4 Command Data Field

Table 4-19: ACTIVATE FILE Command Data Field

P1 Value	Command Data Field Meaning
'04'	ADF Name

4.5.4.5 Response Data Field

Table 4-20: ACTIVATE FILE Response Data Field

Length	Data Element	Presence
1-2	Length of Receipt ('00' – '7F' or '81 80' – '81 FF')	Mandatory
Var	Receipt (for computation description, see [GPCS] section C.5)	Conditional

4.5.4.6 Status Word

Table 4-21: ACTIVATE FILE Status Word

SW1 SW2	Description
'64 00'	Execution error
'69 82'	Security status not satisfied
'90 00'	Successful execution

4.5.5 DEACTIVATE FILE (ADF) Command

4.5.5.1 Description

The DEACTIVATE FILE command deactivates the ADF specified in the command data field.

It is the equivalent of a SET STATUS command sent to lock an application.

4.5.5.2 Condition of Usage

This command shall be performed inside a secure channel. That is, prior to issuing this command, an authentication shall be performed on the ISO-SD with the command GENERAL AUTHENTICATE.

4.5.5.3 APDU Command

Table 4-22: DEACTIVATE FILE Command

CLA	See section 2.1.
INS	'04'
P1	'04' (ADF)
P2	'00'
Lc	Variable (number of bytes in the command data field)
Command Data Field	Application Dedicated File Name
Le	'00'
Response Data Field	None

4.5.5.4 Command Data Field

Table 4-23: DEACTIVATE FILE Command Data Field

P1 Value	Command Data Field Meaning
'04'	Application Dedicated File Name

4.5.5.5 Status Word

Table 4-24: DEACTIVATE FILE Status Word

SW1 SW2	Description
'69 82'	Security status not satisfied
'6A 80'	Incorrect parameters in the command data field
'90 00'	Successful execution

4.6 SELECT (ISD) Command

4.6.1 Description

The SELECT command as defined in Table 4-25 is used to select the Issuer Security Domain.

This command is directly interpreted by the OPEN that selects the corresponding application.

4.6.2 Condition of Usage

The access condition associated with this command is Always. That is, nothing is necessary prior to using this command.

4.6.3 APDU Command

Table 4-25: SELECT ISD Command

CLA	'00' to '03'
INS	'A4'
P1	'04'
P2	'00'
Lc	Not present
Command Data Field	Empty
Le	'00'
Response Data Field	Data requested in P2, as described in Table 4-28

4.6.4 Status Word

Table 4-26: SELECT ISD Status Word

SW1 SW2	Description
'62 83'	The card is locked
'90 00'	Successful execution

4.7 SELECT (Application Dedicated File) Command

4.7.1 Description

The SELECT command as defined in Table 4-27 shall be processed by the OPEN and shall be used to select an ADF or an ISO-SD.

4.7.2 Condition of Usage

The access condition associated with this command is Always. That is, nothing is necessary prior to using this command.

4.7.3 APDU Command

Table 4-27: SELECT ADF Command

CLA	'00' to '03'
INS	'A4'
P1	'04'
P2	See Table 4-28.
Lc	Var
Command Data Field	Application Dedicated File Name
Le	'00'
Response Data Field	Data requested in P2, as described in Table 4-28

Table 4-28: SELECT ADF P2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning	Support
0	0	0	0	0	0	x	0	Return data formatted according to [7816-4] § 11.1.1 that corresponds to File Control Information Template	Mandatory
0	0	0	0	0	1	x	0	Return the Control Parameter (CP) Template	Optional
0	0	0	0	1	0	x	0	Return the File Management Data (FMD)	Optional
0	0	0	0	1	1	x	0	No response data field	Optional
0	0	0	0	x	x	0	0	Select first or only occurrence	Optional
0	0	0	0	x	x	1	0	Select next occurrence	Optional

The Application is required to provide the requested data to the OPEN. The OPEN will forward it in the response of the SELECT command.

4.7.4 Status Word

Table 4-29: SELECT ADF Status Word

SW1 SW2	Description
'6A 81'	Function not supported (e.g. the card is locked)
'6A 82'	Selected application not found
'90 00'	Successful execution

Annex A Use Cases Example

A.1 SCP '03' Key Loading

A.1.1 Context

The ISO-SD is installed and selected.

A secure channel is open with SCP '03' in Clear Mode.

The Key value pushed is equal to '11223344556677889911223344556600'.

The current Key Encryption Key value is '99887766554422113366554477889966'.

The Key that is updated is an AES 128 bit ECB key with a Reference Data Qualifier equal to '9A'.

A.1.2 APDU Description

Update an AES key with a PUT KEY APDU command.

Table A-1: PUT DATA Key Example with Field Description

Class	Ins	P1	P2	Licc	Command Data Field	Le	Response Data Field	SW

'00'	'DB'	'3F'	'FF'	'xx'	{ '5C' '03' '5F9A08' }
					{ '87' '10' '6E74B2BB6DDBF85080B98B332925299F' }
					{ '8E' '03' 'CDAC4D' }
						'00'	None	'90 00'

A.2 Application Installation

A.2.1 Context

The ISO-SD is installed and selected.

A secure channel is open with SCP '03' in Clear Mode.

The application installed is in an Executable Load File whose identifier is '1122334455'.

The corresponding Executable Module is '112233445566'.

The application instance identifier is '11223344556677'.

The application is installed with access right for PUT DATA and GET DATA commands set to always for contact interface.

The application is installed with access right for PUT DATA and GET DATA commands set to never for contactless interface.

A.2.2 APDU Description

Table A-2: Application Installation Description

Class	Ins	P1	P2	Licc	Command Data Field	Le	Response Data Field	SW

'00'	'40'	'04' (transition from Creation to Initialization state)	'00'	'41'	{ '61' '09' { '4F' '07' '11223344556677' } }
					{ '62' '22' { '82' '01' '38' } { '84' '07' '11223344556677' } { 'A3' '09' { '91' '01' '01' } { '9C' '04' '80' '03' '00' '00' } } { 'A3' '09' { '91' '01' '02' } { '9C' '04' '80' '03' 'FF' 'FF' } } }
					{ '64' '10' { '7F49' '06' '112233445566' } { '53' '05' '1122334455' } }
						'00'	None	'90 00'

A.3 Open SCP '03' Secure Channel with Mutual Authentication

A.3.1 Context

The ISO-SD is installed and selected.

The ISO-SD supports the SCP '03' secure protocol with pseudo random computation.

The ISO-SD does not support the R Secure Messaging (R_MAC and R_ENCRYPTION)

The authentication is performed with the Key 9B and with a security level requested of MAC and ENCRYPTION.

The minimum security level requested by the ISO-SD is C_MAC.

The 9B Key sequence counter is equal to '000105'.

The ISO-SD diversification data is equal to '00112233445566778899AABBCCDDEEFF'.

The ISO-SD card Challenge is equal to '1213141516171819'.

The Host Challenge is equal to '9192939495969798'.

The Host and Card Response Cryptogram are equal to 'xxxxxxxxxxxxxxxx'.

A.3.2 APDU Description

Here is the description of the first GENERAL AUTHENTICATE command to open an SCP '03' secure channel:

Table A-3: GENERAL AUTHENTICATE Command #1 Example with Field Description

Class	Ins	P1	P2	Licc	Command Data Field	Le	Response Data Field	SW
'00'	'87'	Cryptographic algorithm reference	Reference Data Qualifier	'11'	{ '7C' (L) { '88' (L) Key Input Information } { '81' '08' Host Challenge } }	'13'	{ '7C' (L) { '85'-'0A' Card Diversification Data } { '88' (L) Key Output Information } { '81' '08' Card Challenge } { '82' '08' Response Cryptogram } { '89' '03' Sequence Counter } }	'90 00'

Here is the description of the second GENERAL AUTHENTICATE command:

Table A-4: GENERAL AUTHENTICATE Command #2 Example with Field Description

Class	Ins	P1	P2	Licc	Command Data Field	Le	Response Data Field	SW
'0C'	'87'	Cryptographic algorithm reference	'00'	'17'	{ '81' '0B' { '7C' {'82' '08' Response Cryptogram } } } { '8E' '08' C-MAC value }	'00'	None	'90 00'