INTERNATIONAL STANDARD

ISO/IEC 7816-8

First edition
1999-10-01

Identification cards — Integrated circuit(s) cards with contacts —

Part 8:
Security related interindustry commands

Cartes d'identification — Cartes à circuit(s) intégré(s) à contacts —

Partie 8: Commandes intersectorielles de sécurité

Reference number
ISO/IEC 7816-8:1999(E)

Contents

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 7816-8 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Identification cards and related devices*.

ISO/IEC 7816 consists of the following parts, under the general title *Identification cards — Integrated circuit(s) cards with contacts*:

— *Part 1: Physical characteristics*

— *Part 2: Dimensions and location of the contacts*

— *Part 3: Electronic signals and transmission protocols*

— *Part 4: Interindustry commands for interchange*

— *Part 5: Numbering system and registration procedure for application identifiers*

— *Part 6: Interindustry data elements*

— *Part 7: Interindustry commands for Structured Card Query Language (SCQL)*

— *Part 8: Security related interindustry commands*

— *Part 9: Additional interindustry commands and security attributes*

— *Part 10: Electronic signals and answer to reset for synchronous cards*

Annexes A and B of this part of ISO/IEC 7816 are for information only.

Introduction

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this part of ISO/IEC 7816 may involve the use of a patent concerning smart cards and terminals given in the body of the text.

The ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from:

Director of Intellectual Property
BULL CP8, S.A.
68, route de Versailles
B.P. 45
78431 Louveciennes Cédex
France

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 7816 may be subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Identification cards - Integrated circuit(s) cards with contacts - Part 8: Security related interindustry

1 Scope

This part of ISO/IEC 7816 specifies:

- security protocols for use in cards;
- secure messaging extensions;
- the mapping of the security mechanisms on to the card's security functions/services, including a description of the in-card security mechanisms;
- data elements for security support;
- the use of algorithms implemented on the card (though the algorithms themselves are not described in detail);
- the use of certificates;
- security related commands.

This part of ISO/IEC 7816 does not cover the internal implementation within the card and/or the outside world.

The choice and conditions of use of cryptographic mechanisms may affect card exportability. The evaluation of the suitability of algorithms and protocols is outside the scope of this part of ISO/IEC 7816.

It shall not be mandatory for cards complying to this part of ISO/IEC 7816 to support all the described commands or all the options of supported commands.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 7816. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However parties to agreements based on this part of ISO/IEC 7816 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 7816-3:1997, *Information technology — Identification cards — Integrated circuit(s) cards with contacts — Part 3: Electronic signals and transmission protocols.*

ISO/IEC 7816-4:1995, *Information technology — Identification cards — Integrated circuit(s) cards with contacts — Part 4: Interindustry commands for interchange.*

ISO/IEC 7816-4:1995/Amd.1:1997, *Information technology — Identification cards — Integrated circuit(s) cards with contacts — Part 4: Interindustry commands for interchange — Amendment 1: Impact of secure messaging on the structures of APDU messages.*

ISO/IEC 7816-6:1996, *Identification cards — Integrated circuit(s) cards with contacts — Part 6: Interindustry data elements.*

ISO/IEC 9796:1991, *Information technology — Security techniques — Digital signature scheme giving message recovery.*

ISO/IEC 9798-2:1994, *Information technology — Security techniques — Entity authentication mechanisms — Part 2: Mechanism using symmetric encipherment algorithms.*

ISO/IEC 9798:1991, *Information technology — Security techniques — Entity authentication mechanisms — Part 3: Entity authentication using a public-key algorithm.*

ISO/IEC 9979:1991, *Data cryptographic techniques — Procedures for the registration of cryptographic algorithms.*

3 Terms and definitions

For the purposes of this part of ISO/IEC 7816, the following definitions apply.

3.1
Certification Authority
CA
a trusted third party that establishes a proof that links a public key and other relevant information to its owner

3.2
cryptographic mechanisms
functions provided by the card as a result of its implementation of cryptographic algorithms with a specific set of operational parameters e.g. the mode of operation and the size of data or keys

3.3
secure messaging
provides a means for cryptographic protection on the data exchanged during a command (as described in ISO/IEC 7816-4)

3.4
security environment
a mechanism to specify to the card system the security functions that are available to provide protection to commands for a specific application of the card

4 Symbols (and abbreviated terms)

For the purposes of this part of ISO/IEC 7816, the following abbreviations apply

APDU	Application protocol data unit
AT	Authentication template
BER-TLV	Basic Encoding Rules - Tag Length Value
CA	Certification authority
CC	Cryptographic checksum
CCT	Cryptographic checksum template
CK	Common key
CRDO	Control reference data object
CRT	Control reference template
CT	Confidentiality template
DE	Data element
DF	Dedicated file
DO	Data object
DS	Digital signature
DSI	Digital signature input
DST	Digital signature template
EF	Elementary file
HT	Hash template
IFD	Interface device
PK	Public key
PSO	PERFORM SECURITY OPERATION command
RFU	Reserved for future use
SE	Security environment
SK	Secret key
SM	Secure messaging
SST	Security support template

5 Security environments

5.1 Description

The security environment (SE) in a card is the logical container of a set of fully specified security mechanisms which are available for reference in security related commands and in secure messaging (SM) as defined in this part of ISO/IEC 7816 and in ISO/IEC 7816-4.

Any SE shall specify references to the cryptographic algorithm(s) to be executed, the mode(s) of operation, the key(s) to be used and any additional data needed by a security mechanism. It may specify a template describing data elements (DEs) stored in the card or resulting from some computation, to be included by the algorithms specified in the security environment definition. It also may provide directions for handling the data resulting from the computation, e.g. storage in the card memory. Any relative references to files (keys or data) specified with a mechanism in the environment definition shall be resolved with respect to the dedicated file (DF) selected at the time the mechanism is used to perform a computation.

Absolute references (e.g. absolute path) need not be resolved.

NOTE — ISO maintains a register of cryptographic algorithms (see ISO/IEC 9979) and, separately, provides protocol standards.

5.2 Activation of a security environment

At any time during operation of the card a current SE shall be active, either by default or as a result of commands from the interface device (IFD). The default SE may be empty. The content of the default SE is not defined in this part of ISO/IEC 7816.

The current SE may explicitly be set or replaced with the MANAGE SECURITY ENVIRONMENT command (see clause 10). An SE may contain a mechanism to perform initialisation of non-persistent data used by mechanisms in the environment, e.g. a session key.

In SM, data objects transmitted in a control reference data object (CRDO) shall take precedence over any corresponding data object (DO) present in the current SE.

Definitions of associated SE's may be grouped into the following sets:

- One global SE set, which may be provided by the card. The first SE of this set is the default SE;
- One or more application specific SE sets which are provided by applications.

The global SE set shall be active by default, unless otherwise specified. A SE or set of SEs may be associated with a DF or EF such that after selecting the DF or EF the associated SE or a specific SE in the set is implicitly set. The method of specifying this functional association between a file and a set of SEs is outside the scope of this part of ISO/IEC 7816.

The current SE is valid until there is a change of context (e.g. by selecting a different application with the SELECT FILE command), a MANAGE SECURITY ENVIRONMENT command, a warm reset or deactivation of the contacts (see ISO/IEC 7816-3).

5.3 Components

Control Reference Templates (CRT) may be used to describe the various components of a SE (see Table 2).

Five such templates are defined for:

- cryptographic checksum;
- digital signature;
- confidentiality;
- hash;
- authentication.

Within the SE, components may have two aspects; one being valid for the protection of command APDUs (application protocol data units) and the other for the protection of response APDUs.

SEs may be numbered for storing, restoring (see clause 10) and referencing, in which case the numbering is context specific.

SE numbers represented by:

- all zeroes (0) denote an empty environment, where no authentication no SM procedure is defined;
- all ones (1) denote that no operation can be performed in this environment;
- 11101111 is Reserved for Future Use (RFU).

The current SE contains one or more:

- components belonging to the default stored SE associated with the current DF;
- components transmitted in SM commands (see ISO/IEC 7816-4);
- components transmitted in MANAGE SECURITY ENVIRONMENT commands (see clause 10);
- all the components of a stored SE, invoked by its number in a MANAGE SECURITY ENVIRONMENT command.

5.4 Algorithm referencing

The Algorithm Object Identifier DO is a data object which identifiers the cryptographic algorithm associated with an algorithm reference, as defined in ISO/IEC 7816-4. One or more such DOs may be present in the file control information (FCI) of a DF with a tag 'AC'.

This DO encapsulates two mandatory DOs and an optional DO, in the following sequence:

- the first mandatory DO is the algorithm reference DO, tag '80', as used in Table 3;
- the second mandatory DO is an ASN.1 DO Identifier, tag '06', referencing the algorithm uniquely;
- the optional DO (tag dependent on the Object Identifier) indicates the algorithm parameters.

Example coding (see ISO/IEC 7816-6, Annex B) -

AC II 09 II 80-01-01 II 06-04-28CC4701

This Object Identifier (28CC4701) refers to algorithm 1 in ISO/IEC 9979, with no parameter.

6 Extended headerlist DE

6.1 Construction and use

An extended headerlist DE is a concatenation of tag/lengths without delimiters.

An extended headerlist is normally used for referencing DOs to be signed.

An extended headerlist references a byte string built as follows:

- each tag/length is replaced by data referenced by the tag when the DO is primitive;
- when a tag/length denotes a constructed DO, its value is interpreted as an extended headerlist DE.

According to the conditions of use of an extended headerlist, the data to include in the byte string are

- either the values of the referenced primitive DOs, truncated according to the length indicated in the extended headerlist (Case 1) or
- the primitive DOs themselves, truncated according to the length indicated in the extended headerlist, and nested in the respective template, the length of which is adjusted according to BER-TLV (Basic Encoding Rules - Tag length Value) rules (Case 2).

A constructed tag followed by a length = 00 is ignored.

A primitive tag followed by a length = 00 indicates that the complete DO or DE is to be included in the byte string.

A DO, the value of which is an extended headerlist, uses tag '4D'. According to their use, other DOs may have the implicit type 'extended headerlist'.

6.2 Examples of extended headerlists

Given an extended headerlist:

Primitive T_1	00	Const . tag	L = 4	Primitive T_2	00	Primitive T_3	L = 5

describing 3 primitive DOs:

Primitive T_1	L_1	$Value_1$

Primitive T_2	L_2	$Value_2$

Primitive T_3	L_3 (≥5)	$Value_3$

Result in Case 1 (the headerlist referring to the concatenation of the DEs)

$Value_1$	$Value_2$	$Value_3$ truncated at 5 bytes

Result in Case 2 (the headerlist referring to the concatenation of the DOs)

Primitive T_1	L_1	$Value_1$	Const. tag	$L = L_2 + 9$	Primitive T_2

—

L_2	$Value_2$	Primitive T_3	5	$Value_3$ truncated at 5 bytes

indicated by the appropriate parameter of the command (e.g. 'AC', 'BC' in PERFORM SECURITY OPERATION, see 11.7.3) or by the appropriate structure of the data field: constructed for those containing DOs; primitive for those containing DEs).

7 Security support

7.1 Description and rules

The security support data elements are a collection of specially defined DEs with rules governing the way their values are handled. These DEs may be provided by the card as generic support to cryptographic protection mechanisms performed by an application.

The security support DEs may be referenced by applications for inclusion in operations executed by the card when performing commands e.g. in secure messaging or in the PERFORM SECURITY OPERATION command. The security support DEs extend and refine the auxiliary data elements for secure messaging as defined in ISO/IEC 7816-4.

The rules for maintenance and use of the value of security support DEs shall be governed by the card. They are based on the following principles:

- update is done with new values computed by the card or provided by the outside world, in accordance with the specific rule for a specific type of security support DE;
- update is performed before any output is produced for a command which causes an update. The update is independent of the completion status of the command;
- if the value is to be used by the application in an operation that causes an update, the update is performed before the value is used;
- access to application specific security support DEs is restricted to functions performed by the specific application.

NOTE — the actual security achieved in a data exchange ultimately depends on the algorithms and protocols specified by the application, the card only provides support with these DEs and associated usage rules.

7.2 Data elements

The card may support security of data exchanges with data elements having values that are different each time the card is activated. They include the following:

- A card session counter, that is incremented once during card activation;
- A session identifier, that may be computed from the card session counter and possibly data provided by the outside world.

Cryptographic protection of data exchanges may be supported with data elements, called progression values. Their values are increased at specific events throughout the life of the card.

Two progression value types are specified:

- Internal Progression Values which, if so specified for an application, register the number of times specific events are performed. The data element shall be incremented after the event has occurred; the card may provide a reset function for these counters which if so specified for an application sets its value to zero. Internal progression values cannot be controlled by the outside world and are suitable for use as secured in-card approximate representations of real time. Their values can be used in cryptographic computations.
- External Progression Values which, if so specified for an application, shall only be updated by a data value from the outside world. The new value shall be numerically larger than its current stored value.

7.3 Data element referencing

Access to the value of security support data elements may be provided in a card by:

- an EF contained in the master file (MF), e.g. for card session counter;
- an EF contained in a DF associated with an application, e.g. for application specific progression values;
- a reference as a data object with BER-TLV encoding as defined in the first column of Table 1;
- a reference as auxiliary data (tags '88', '92', '93') in a control reference template (CRT, see Table 3). These tags can be used if the SE supports unambiguous use of these data elements.

Characteristics of the security support data elements, e.g. length of the data, and the algorithms which alter their value are not defined in this part of ISO/IEC 7816.

Table 1 — Tags for security support data objects

Tag	Meaning
'7A'	Security Support Template (SST), to encapsulate DOs with the following tags
'80'	Card session counter
'81'	Session identifier
'82' - '8E'	File selection counter
'93'	Digital signature counter
'9F2X'	Internal progression value
'9F3Y'	External progression value

The coding of 'X' in Table 1 is an index of a specific internal progression value e.g. a counter of file selections.

The coding of 'Y' in Table 1 is an index of a specific external progression value e.g. an external time stamp.

8 Secure messaging extensions

8.1 Secure messaging data objects

Table 2 lists the SM data objects defined in ISO/IEC 7816-4 and Amendment 1, and the SM data objects defined in this part of ISO/IEC 7816.

Table 2 - Secure messaging data objects

ISO/IEC 7816 Part-	Tag	Value
4	'80' '81'	Plain value (non BER-TLV coded data)
4	'B0', 'B1'	Plain value (BER-TLV, including SM related data objects)
4	'B2', 'B3'	Plain value (BER-TLV, but not SM related data objects)
4	'96', '97'	Value of Le in an unsecured command (see ISO/IEC 7816-4, Amendment 1)
4	'99'	Status information (e.g. SW1-SW2)
4	'82' '83'	Cryptogram, the plain value consisting of BER-TLV including SM related data objects
4	'84' '85'	Cryptogram, the plain value consisting of BER-TLV, but not SM related data objects
4	'86' '87'	Padding indicator byte (see ISO/IEC 7816-4) followed by cryptogram (plain value not coded in BER-TLV)
4	'8E'	Cryptographic checksum (at least 4 bytes)
4	'9E'	Digital signature
8	'90'	Hash Code
4	'9A'	Input for Digital Signature (non BER-TLV coded data)
8	'A0'	Input template for Hash Code
8	'A2'	Input template for cryptographic checksum verification
8	'A8'	Input template for DS verification
8	'AC'	Input template for DS (BER-TLV coded data, the concatenation of the value fields are signed)
8	'BC'	Input template for DS (BER-TLV coded data, TLV data are signed)
8	'92'	Certificate (non BER-TLV coded data)
8	'AE'	Input template for certificate verification (signed signature input consisting of non BER-TLV coded data)
8	'BE'	Input template for certificate verification (signed signature input consisting of BER-TLV coded data)
8	'A4', 'A5'	CRT for authentication (AT)
8	'AA', 'AB'	CRT for hash code (HT)
4	'B4', 'B5'	CRT for cryptographic checksum (CCT)
4	'B6', 'B7'	CRT for digital signature (DST)
4	'B8', 'B9'	CRT for confidentiality (CT)
4	'BA', 'BB'	Response descriptor

8.2 Control reference data objects

Table 3 lists the CRDOs defined in ISO/IEC 7816-4 and this part of ISO/IEC 7816. The table indicates to which CRT they are relevant: cryptographic checksum template (CCT), digital signature template (DST), confidentiality template (CT), hash template (HT) and authentication template (AT).

Table 3 - Data objects within control reference templates

Tag	Value	CCT 'B4', 'B5'	DST 'B6', 'B7'	CT - Asym 'B8', 'B9'	CT - Sym 'B8', 'B9'	HT 'AA', 'AB'	AT 'A4', 'A5'
'4D'	L≠0, extended headerlist of DOs as defined in clause 6		x			x	
'5D'	L≠0, Headerlist, as defined in ISO/IEC 7816-6		x			x	
'80'	Algorithm reference *	x	x	x	x	x	x
	File reference *						
'81'	- file identifier or path	x	x	x	x	x	
'82'	- DF name	x	x	x	x	x	
	Key reference *						
'83'	- for direct use in symmetric cases	x			x	x (CK)	x
	- for referencing a public key in asymmetric cases		x	x		x	x
'84'	- for computing a session key in symmetric cases	x			x		x
	- for referencing a private key in asymmetric cases		x	x			
	Initial check block *						
'85'	L=0, null block	x			x	x	
'86'	L=0, chaining block	x			x	x	
'87'	L=0, e.g. previous initial value block + 1	x			x		
	L=k, initial value block (IV block)						
	Auxiliary data						
'88'	L=0, previous challenge + 1	x	x	x	x		
	L≠0, reference data object not specified						
'90'	L=0, hash code provided by the card		x			x	
'91'	L=0, random no. provided by the card	x	x	x			
	L≠0, random number		x	x			
'92'	L=0, time stamp provided by the card		x	x		x	
	L≠0, time stamp		x	x			
'93'	L=0, previous counter + 1**		x	x	x	x	
	L≠0, counter		x	x	x		
'89' to	L=0, index of a proprietary data item				x		
'8D'	L≠0, value of a proprietary data item						
'8E'	Cryptogram contents reference *			x	x		
'94'	Challenge or data item for deriving a key	x			x		x

* = as defined in ISO/IEC 7816-4

** = Digital signature counter.

8.3 Input data objects

Table 4 lists data objects present in input templates to perform security operations, according to this part of ISO/IEC 7816. The table indicates to which input template the DOs are relevant.

Table 4 - Input data objects

Tag	Value	Hash	Cryptographic checksum verification	Digital signature verification	Digital signature	Certificate verification
'80'	Plain value	x	x	x	x	x
'8E'	Cryptographic checksum		x			x
'90'	Hash Code	x		x	x	x
'92'	Certificate					x
'9C'	Public Key			x		x
'9E'	Digital signature			x		x

9 Command chaining

Many security processes are multi-stage. Such processes can be naturally carried out using several consecutive command-response pairs with the same INS code. In this case a different CLA value shall be used for the last (or only) command involved.

Table 5 - CLA coding for command chaining

CLA coding	Meaning
'0X'	for the last (or only) command involved
1X'	for a command which is not the last command

The interpretation of the least significant nibble is as defined in Table 9 of ISO/IEC 7816-4.

During command chaining each command shall have the same value of X in the CLA.

When a security process has been initiated it shall be completed before any other command is issued. Otherwise the behaviour of the card is not specified.

If SW1-SW2 is set to '9000' in a response to a command which is not the last command, then it means that the processing has been successful so far.

If an error occurs in the middle of a series of chained commands, it is indicated by an appropriate value of SW1-SW2.

Table 6 - Command chaining specific status coding

SW1	SW2	Meaning
'68'	'83'	Final command expected
'68'	'84'	Command chaining not supported

10 MANAGE SECURITY ENVIRONMENT command

10.1 Definition and scope

The MANAGE SECURITY ENVIRONMENT command supports the following functions:

— replacing the current SE by a SE stored in the card (RESTORE);

— setting, or replacing, one component of the current SE (SET);

— saving the current SE under a SE number (STORE);

— erasing a SE identified with a SE number (ERASE);

— initializing cryptographic commands.

The usage of a master key concept may require the derivation of a key in the card containing the master key (see 10.6).

10.2 Conditional usage and security

No conditions are described in this part of ISO/IEC 7816.

10.3 Command message

Table 7 - MANAGE SECURITY ENVIRONMENT command APDU

CLA	As defined in ISO/IEC 7816-4 and clause 9
INS	'22'
P1	See Table 8
P2	See Table 9
Lc	Either length of subsequent data field in the case of SET or empty in the case of STORE, RESTORE and ERASE
Data field	Either a concatenation of CRDOs (in the case of SET) or empty in the case of STORE, RESTORE and ERASE
Le	Empty

Table 8 - Coding of P1

b 8	b 7	b 6	b 5	b 4	b 3	b 2	b 1	Meaning
-	-	-	1	-	-	-	-	SM command
-	-	1	-	-	-	-	-	SM response
-	1	-	-	-	-	-	-	Computation, decipherment and internal authentication
1	-	-	-	-	-	-	-	Verification, encipherment and external authentication
-	-	-	-	0	0	0	1	SET
1	1	1	1	0	0	1	0	STORE
1	1	1	1	0	0	1	1	RESTORE
1	1	1	1	0	1	0	0	ERASE

Table 9 - Coding of P2

b8..b1	Meaning
	in case of STORE, RESTORE and ERASE
'xy'	SE number
	in the case of SET
'B4'	value of CCT in data field
'B6'	value of DST in data field
'AA'	value of HT in data field
'B8'	value of CT in data field
'A4'	value of AT in data field
Any other value RFU	

10.4 Response message

Table 10 - MANAGE SECURITY ENVIRONMENT response APDU

Data field	Empty
SW1-SW2	Status bytes

10.5 Status conditions

The following specific error conditions may occur:

— SW1 = '66' with SW2 =

- '00' : The environment cannot be set or modified, no further information;

— SW1 = '69' with SW2 =

- '87': Expected SM data objects missing;
- '88': SM data objects incorrect;

— SW1 = '6A' with SW2 =

- '88': Referenced data not found.

10.6 Computing a derived key with MANAGE SECURITY ENVIRONMENT

Table 11 shows the usage of the MANAGE SECURITY ENVIRONMENT command for deriving a key. It is assumed that in the card the master key and the algorithm is implicitly selected (if not, the key and the algorithm can be selected in the MANAGE SECURITY ENVIRONMENT command additionally).

Table 11 - Command message APDU

CLA	As defined in ISO/IEC 7816-4
INS	'22'
P1	'X4' = SET (see Table 8)
P2	Tag of the related CRT (e.g. 'A4', if an authentication related command follows (e.g. EXTERNAL AUTHENTICATE) or 'B4', if a cryptographic checksum related command follows (e.g. PSO: VERIFY CC))
Lc	Length of subsequent data field
Data field	'94' - L - Data for deriving a key (mandatory - further DOs are possible (see Table 3)
Le	Empty

Table 12 - Response message APDU

Data field	Empty
SW1-SW2	Status bytes

NOTE — Depending on the algorithm reference the data for deriving a key from a master key may be part of the input data of the subsequent command (e.g. EXTERNAL AUTHENTICATE). In this case the usage of the MANAGE SECURITY ENVIRONMENT command for deriving the key is not necessary.

11 PERFORM SECURITY OPERATION command

11.1 Definition and scope

The PERFORM SECURITY OPERATION command initiates the following security operations:

- computation of a cryptographic checksum;
- computation of a digital signature;
- calculation of a hash code;
- verification of a cryptographic checksum;
- verification of a digital signature;
- verification of a certificate;
- encipherment;
- decipherment.

The security operations initiated are related to the DOs specified in P1 and P2. The command may be performed in one or several steps, possibly using the command chaining function (see 9).

11.2 Conditional usage and security

The PERFORM SECURITY OPERATION command may be preceded by a MANAGE SECURITY ENVIRONMENT command. The successful execution of the command may be subject to successful completion of prior commands (e.g. VERIFY before the computation of a digital signature).

The key reference as well as the algorithm reference shall be

- either implicitly known or
- specified in a CRT in a MANAGE SECURITY ENVIRONMENT command.

If present, a headerlist defines the order and the data items which form the input for the security operations.

11.3 Command message

Table 13 - PERFORM SECURITY OPERATION command APDU

CLA	As defined in ISO/IEC 7816-4 and clause 9
INS	'2A'
P1	Tag of the DO, the value field of which is transmitted in the response data field, or
	'00' data field in response is empty
	'FF' = RFU
P2	Tag of the DO, the value field of which is transmitted in the command data field, or
	'00' data field in command is empty
	'FF' RFU
Lc field	Length of the subsequent data field
Data field	Value of the DO specified in P2, or empty
Le field	Empty or maximum length of the data expected in response

11.4 Response message

For response messages see the relevant clause under each operation.

11.5 Status conditions

Unless specifically stated, as defined in ISO/IEC 7816-4 and in this part of ISO/IEC 7816.

11.6 COMPUTE CRYPTOGRAPHIC CHECKSUM operation

11.6.1 Definition and scope

The COMPUTE CRYPTOGRAPHIC CHECKSUM operation initiates the computation of a cryptographic checksum.

11.6.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

11.6.3 Command message

Table 14 - COMPUTE CRYPTOGRAPHIC CHECKSUM parameter and command DOs

P1	'8E'
P2	'80'
Data field	data for which the cryptographic checksum shall be computed

NOTE — this operation may be subject to command chaining

11.6.4 Response message

Table 15 - COMPUTE CRYPTOGRAPHIC CHECKSUM response APDU

Data field	Cryptographic checksum
SW1-SW2	Status bytes

11.7 COMPUTE DIGITAL SIGNATURE operation

11.7.1 Definition and scope

The COMPUTE DIGITAL SIGNATURE operation initiates the computation of a digital signature.

For the computation of a digital signature the data to be signed or to be integrated in the signing process are transmitted in the data field of the PERFORM SECURITY OPERATION command. In P2 the digital signature is specified with tags '9A', 'AC' or 'BC' according to the structure of the input (see Table 2).

The algorithm may be either a digital signature algorithm or a combination of a hash algorithm and a digital signature algorithm.

If auxiliary data (see Table 3) are to be included in the Digital Signature Input (DSI - see Table 3), then a reference has to be made in the CRT (see 6.1). If an empty reference data object for auxiliary data is present, then the auxiliary data are to be inserted by the card.

Auxiliary data present or referenced in the data field take precedence over any headerlist.

The value to be returned by the card is a digital signature specified in P1 ('9E').

11.7.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

11.7.3 Command message

Table 16 - COMPUTE DIGITAL SIGNATURE parameter and command DOs

P1	'9E'
P2	'9A', 'AC' or 'BC'
Data field	If P2 = '9A': data to be signed or integrated in the signature process if P2 = 'AC': DOs relevant for DSI (the value field of these DOs are signed or integrated in the signature process) if P2 = 'BC': DOs relevant for DSI (the DOs are signed or integrated in the signature process)

NOTE — Tags 'AC' and 'BC' are not integrated into the digital signature input.

11.7.4 Response message

Table 17 - COMPUTE DIGITAL SIGNATURE response APDU

Data field	Digital Signature
SW1-SW2	Status bytes

11.8 HASH operation

11.8.1 Definition and scope

The HASH operation initiates the calculation of a hash code by performing:

— either complete hashing inside the card or

— partly hashing inside the card (e.g. last round of computation.

The algorithm reference for computing a hash code (see Table 3) is indicated in the CRT for Hash computation ('AA', 'AB'), see Table 2.

For the further processing of a computed hash code the following cases have to be distinguished:

— the hash code is stored in the card: the calculated hash code is stored in the card and available for use in a subsequent command and the Le field is empty;

— the hash code is delivered by the card in the response: if the hash code is delivered in the response then the Le field has to be set to the appropriate length.

The data to be hashed shall be presented to the card in successive blocks (one or more at a time), the length of which is algorithm dependent. Depending on the hash algorithm the last block presented may have a length equal or shorter than the block length. The padding mechanism, if appropriate, is part of the definition of the hash algorithm.

11.8.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

11.8.3 Command message

Table 18 - HASH parameter and command DOs

P1	'90'
P2	'80' or 'A0'
Data field	If P2 = '80': data to be hashed If P2 = 'A0': DOs relevant for hashing (e.g. '90' for the intermediate hash code, '80' for the last text block)

11.8.4 Response message

Table 19 - HASH response APDU

Data field	Hash code or empty
SW1-SW2	Status bytes

11.9 VERIFY CRYPTOGRAPHIC CHECKSUM operation

11.9.1 Definition and scope

The VERIFY CRYPTOGRAPHIC CHECKSUM operation initiates the verification of a cryptographic checksum.

11.9.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

11.9.3 Command message

Table 20 - VERIFY CRYPTOGRAPHIC CHECKSUM parameter and command DOs

P1	'00'
P2	'A2'
Data field	DOs relevant to the VERIFY CRYPTOGRAPHIC CHECKSUM operation (e.g. DO '80', '8E' see Table 2)

NOTE — this operation may be subject to command chaining.

11.9.4 Response message

Table 21 - VERIFY CRYPTOGRAPHIC CHECKSUM response APDU

Data field	Empty
SW1-SW2	Status bytes

11.10 VERIFY DIGITAL SIGNATURE operation

11.10.1 Definition and scope

The VERIFY DIGITAL SIGNATURE operation initiates the verification of a digital signature to be delivered as a DO in the data field. Other verification relevant data are to be transmitted in a command chaining process or may be present in the card.

The public key as well as the algorithm may be

— either implicitly known or

— referenced in a DST ('B6') of a MANAGE SECURITY ENVIRONMENT command or

— available as a result from a preceding VERIFY CERTIFICATE operation.

The algorithm may be either a digital signature only algorithm or a combined hash code and digital signature algorithm.

If the reference of the algorithm in the card declares a signature only algorithm then the data consists of a hash code, or the signature is of message recovery type according to ISO/IEC 9796. Otherwise the hash code calculation is performed in the card and the algorithm reference additionally contains a reference to a hash algorithm.

11.10.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

11.10.3 Command message

Table 22 - VERIFY DIGITAL SIGNATURE parameter and command DOs

P1	'00'
P2	'A8'
Data field	DOs relevant to the VERIFY DIGITAL SIGNATURE operation (e.g. DO '9A', 'AC' and 'BC', and DO '9E', see Table 2)

If the data field contains an empty DO, the card is expected to know its values for use in the verification.

11.10.4 Response message

Table 23 - VERIFY DIGITAL SIGNATURE Response APDU

Data field	Empty
SW1-SW2	Status bytes

11.11 VERIFY CERTIFICATE operation

11.11.1 Definition and scope

For the verification of a certificate in a card (see Annex A) the digital signature of a certificate to be verified is delivered as a DO in the data field. The public key of the certification authority to be used in the verification process shall be present in the card and is either implicitly selected or may be referenced in a DST using the MANAGE SECURITY ENVIRONMENT command. The algorithm to be applied is implicitly known or may be referenced in a DST. If other DOs are to be used in the verification process (e.g. hash code) then these DOs shall be present in the card or shall be transmitted by the command chaining process described in clause 9.

The following cases have to be distinguished:

- the certificate is self descriptive (P2 = 'BE'): the card retrieves a public key identified by its tag in the (recovered) certificate content;
- the certificate is not self descriptive (P2 = 'AE'): the card retrieves a public key in the certificate either implicitly or explicitly by using the public key tag in a headerlist describing the content of the certificate.

11.11.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

If the public key is stored, it will be the default key for subsequent VERIFY DIGITAL SIGNATURE operation.

11.11.3 Command message

Table 24 - VERIFY CERTIFICATE parameter and command DOs

P1	'00'
P2	'92', 'AE', 'BE'
Data field	DEs or DOs relevant to the VERIFY CERTIFICATE operation (see Table 2)

NOTE — If a limited message recovery algorithm is used and part of the information is already stored in the card, then the DO for auxiliary data shall be sent empty, with the data to be inserted later by the card.

11.11.4 Response message

Table 25 - VERIFY CERTIFICATE response APDU

Data field	Empty
SW1-SW2	Status bytes

11.12 ENCIPHER operation

11.12.1 Definition and scope

The ENCIPHER operation enciphers data transmitted in the command data field.

NOTE — this operation may also be used to generate diversified keys.

11.12.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

The usage of this operation may be restricted.

11.12.3 Command message

Table 26 - ENCIPHER parameter and command DOs

P1	'82', '84', '86' (cryptogram, see Table 2)
P2	'80' (plain value)
Data field	data to be enciphered

11.12.4 Response message

Table 27 - ENCIPHER response APDU

Data field	enciphered data
SW1-SW2	Status bytes

11.13 DECIPHER operation

11.13.1 Definition and scope

The DECIPHER operation deciphers data transmitted in the command data field.

11.13.2 Conditional usage and security

The command can be performed only if the security status satisfies the security attributes for this operation.

The usage of this operation may be restricted.

11.13.3 Command message

Table 28 - DECIPHER parameter and command DOs

P1	'80' (plain value)
P2	'82' ,'84' ,'86' (cryptogram, see Table 2)
Data field	data to be deciphered

11.13.4 Response message

Table 29 - DECIPHER response APDU

Data field	Deciphered data
SW1-SW2	Status bytes

12 Manage verification process

12.1 Introduction

The following commands belong to the manage verification process:

— VERIFY, as defined in ISO/IEC 7816-4;

— CHANGE REFERENCE DATA;

— ENABLE VERIFICATION REQUIREMENT;

— DISABLE VERIFICATION REQUIREMENT;

— RESET RETRY COUNTER.

Specific warning and error conditions for the manage verification process commands are given in clause 12.6.

The security status may be set by the command itself if it contains the appropriate parameters. If not the security status may be set by different means e.g. secure messaging, previous authentication.

The security status may be modified as a result of a comparison. Unsuccessful comparisons may be recorded in the card (e.g. to limit the number of further attempts of the use of the reference data).

12.2 CHANGE REFERENCE DATA command

12.2.1 Definition and scope

The CHANGE REFERENCE DATA command is used

— either to replace the existing reference data with new reference data or

— to initiate the comparison of the verification data with the reference data, and then to conditionally replace the existing reference data with new reference data sent to the card in the command.

12.2.2 Conditional usage and security

This command can be performed only if the security status satisfies the security attributes valid for this command.

12.2.3 Command message

Table 30 - CHANGE REFERENCE DATA command APDU

CLA	As defined in ISO/IEC 7816-4
INS	'24'
P1	'00' = exchange reference data
	'01' = change reference data
	Any other value is RFU
P2	Qualifier of the reference data, see Table 31
Lc field	Length of the subsequent data field
Data field	P1 = '00' existing reference data followed by new reference data
	P1 = '01' new reference data
Le field	Empty

NOTE — The length of the existence reference data is known by the card. Therefore no delimeter between existing and new reference data is present.

Table 31 - Coding of the reference control P2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	- no information is given
0	-	-	-	-	-	-	-	- global reference data
								(e.g. card password)
1	-	-	-	-	-	-	-	- specific reference data
								(e.g. DF specific password)
-	x	x	-	-	-	-	-	00 (other values are RFU)
-	-	-	x	x	x	x	x	- reference data number

NOTE 1 — P2 = '00' is reserved to indicate that no particular qualifier is used, in those cards where the command references the reference data unambiguously.

NOTE 2 — The reference data number may be, for example, a password number or short EF identifier.

12.2.4 Response message (Nominal case)

Table 32 - CHANGE REFERENCE DATA response APDU

Data field	Empty
SW1-SW2	Status bytes

12.3 ENABLE VERIFICATION REQUIREMENT command

12.3.1 Definition and scope

The ENABLE VERIFICATION REQUIREMENT command is used to switch on the requirement to compare the verification data with the reference data.

12.3.2 Conditional usage and security

This command can be performed only if the security status satisfies the security attributes valid for this command.

12.3.3 Command message

Table 33 - ENABLE VERIFICATION REQUIREMENT command APDU

CLA	As defined in ISO/IEC 7816-4
INS	'28'
P1	'00' = verification data present in data field
	'01' = no verification data
	Any other value is RFU
P2	Qualifier of the reference data, see Table 31
Lc field	Empty or length of the subsequent data field
Data field	P1 = '00' verification data
	P1 = '01' empty
Le field	Empty

12.3.4 Response message (Nominal case)

Table 34 - ENABLE VERIFICATION REQUIREMENT response APDU

Data field	Empty
SW1-SW2	Status bytes

12.4 DISABLE VERIFICATION REQUIREMENT command

12.4.1 Definition and scope

The DISABLE VERIFICATION REQUIREMENT command is used to switch off the requirement to compare the verification data with the reference data.

12.4.2 Conditional usage and security

This command can be performed only if the security status satisfies the security attributes valid for this command.

12.4.3 Command message

Table 35 - DISABLE VERIFICATION REQUIREMENT command APDU

CLA	As defined in ISO/IEC 7816-4
INS	'26'
P1	'00' = verification data present in data field '01' = no verification data Any other value is RFU
P2	Qualifier of the reference data, see Table 31
Lc field	Empty or length of the subsequent data field

Data field	P1 = '00' verification data
	P1 = '01' empty

Le field	Empty

12.4.4 Response message (Nominal case)

Table 36 - DISABLE VERIFICATION REQUIREMENT response APDU

Data field	Empty
SW1-SW2	Status bytes

12.5 RESET RETRY COUNTER command

12.5.1 Definition and scope

The RESET RETRY COUNTER command is used to

— either reset the reference data retry counter to its initial value or

— to change the reference data on completion of a successful reset of the reference data retry counter to its initial value.

12.5.2 Conditional usage and security

This command can be performed only if the security status satisfies the security attributes valid for this command.

12.5.3 Command message

Table 37 - RESET RETRY COUNTER command APDU

CLA	As defined in ISO/IEC 7816-4
INS	'2C'
P1	'00' = reset retry counter and set new reference data '01' = reset retry counter, resetting code in data field '02' = reset retry counter and set new reference data (no resetting code) '03' = reset retry counter, data field empty Any other value is RFU
P2	Qualifier of the reference data, see Table 31
Lc field	Empty or length of the subsequent data field

Data field	P1 = '00' resetting code followed by new reference data
	P1 = '01' resetting code
	P1 = '02' new reference data
	P1 = '03' empty

Le field	Empty

NOTE — When P1 = '00' the length of the resetting data is known by the card. Therefore no delimeter between resetting code and new reference data is present.

12.5.4 Response message (Nominal case)

Table 38 - RESET RETRY COUNTER response APDU

Data field	Empty
SW1-SW2	Status bytes

12.6 Status conditions for manage verification process commands

The following specific warning condition may occur:

— SW1 = '63' with SW2 =

• '00' : No information given (verification failed);

• 'CX' : Counter (verification failed; 'X' indicates the number of further allowed retries).

The following specific error condition may occur:

— SW1 = '65' with SW2 =

- '81' : Memory failure (unsuccessful changing);

— SW1 = '67' with SW2 =

- '00' : Wrong length (empty Lc field);

— SW1 = '69' with SW2 =

- '82' : Security status not satisfied;
- '83' : Authentication method blocked
- '84' : Reference data invalidated;

— SW1 = '6A' with SW2 =

- '81' : Function not supported;
- '82' : File not found
- '86' : Incorrect parameter P1-P2;
- '88' : Reference data not found.

13 GENERATE PUBLIC KEY PAIR command

13.1 Definition and scope

The GENERATE PUBLIC KEY PAIR command initiates the generation and storing of a public key (PK) pair in the card.

13.2 Conditional usage and security

The GENERATE PUBLIC KEY PAIR command may be preceded by a MANAGE SECURITY ENVIRONMENT command in order to set key generation related parameters e.g. algorithm reference.

The operation can be performed only if the security status satisfies the security attributes for this operation.

13.3 Command message

Table 39 - GENERATE PUBLIC KEY PAIR command APDU

CLA	As defined in ISO/IEC 7816-4
INS	'46'
P1-P2	'0000' = Generate and store PK pair
	Other values RFU
Lc	Empty or length of subsequent data field
Data field	Empty or proprietary data
Le	Empty or length of expected data

13.4 Response message

Table 40 - GENERATE PUBLIC KEY PAIR response APDU

Data field	Empty or public key
SW1-SW2	Status bytes

13.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '85' : Conditions of use not satisfied.

14 MUTUAL AUTHENTICATE function

14.1 Definition and scope

The MUTUAL AUTHENTICATE function uses the same functionality as the INTERNAL and EXTERNAL AUTHENTICATE commands, and uses the INS code for EXTERNAL AUTHENTICATE (see ISO/IEC 7816-4).

MUTUAL AUTHENTICATE is based on a previous GET CHALLENGE command issued by the IFD, and a key, possibly secret, stored in the card. Authentication related data are shared by the IFD and the card. These data include a challenge issued by the card and a further challenge generated by the IFD.

NOTE — This command may be used to implement authentication as described in ISO/IEC 9798, Parts 2 and 3.

14.2 Conditional usage and security

The operation can be performed only if the security status satisfies the security attributes for this operation.

14.3 Command message

Table 41 - MUTUAL AUTHENTICATION command APDU

CLA	As defined in ISO/IEC 7816-4
INS	'82'
P1	'00' or reference of the algorithm
P2	'00' or reference of the key
Lc	Length of subsequent data field
Data field	Authentication related data
Le	Length of expected authentication related data

14.4 Response message

Table 42 - MUTUAL AUTHENTICATION response APDU

Data field	authentication related data
SW1-SW2	Status bytes

14.5 Status conditions

Status conditions as EXTERNAL AUTHENTICATION, see ISO/IEC 7816-4.

15 Tags defined in ISO/IEC 7816-8

The following tags, (shown in Table 43), are defined, or have special meaning, in this part of ISO/IEC 7816.

Table 43 - Tags defined in ISO/IEC 7816-8

Tag	Description
'4D'	Extended headerlist
'7A'	Security support template (SST) The following are encapsulated under '7A': '80' Card session counter '81' Session identifier '9F2X' Internal progression value '9F2Y' External progression value
'5F4E'	Certificate content (concatenation of DEs according to an extended headerlist)

Annex A
(informative)

Structure and usage of certificates interpreted by the card

A.1 Terms and definitions

For the purposes of this annex, the following definitions apply:

A.1.1
Public Key Certificate

A public key certificate is an unforgeable piece of digital information that guarantees a binding between a particular person or object and its associated public key. It is issued by a certification authority which acts also as a tag allocation authority with respect to the data items present in a certificate.

A.1.2
Card Verifiable (CV) Certificate

A CV certificate is a PK certificate, which can be interpreted in a card by applying the VERIFY CERTIFICATE operation. It consists of BER-TLV coded data objects (see Table A.1).

A.2 Symbols (and abbreviated terms)

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

$ID_{Cardholder}$	The ID of the cardholder
$PK_{Cardholder}$	The public key of the cardholder
ID_{CA}	The ID of a certification authority
ll	Concatenation

A.3 Data objects for CV certificates

Table A.1 shows data objects relevant for CV certificates.

Table A.1 - Interindustry DOs (examples) relevant for CV certificates

Tag	Data element
'42'	Certification authority reference, e.g. name or id of issuer authority*
'5F20'	Certificate holder reference, e.g. cardholder name*
'5F37'	Signature of a certificate, produced by the related CA*
'5F49'	Certificate holder public key, e.g. cardholder public key*
'5F4E'	Certificate content
'7F21'	CV certificate, constructed, e.g. cardholder certificate*

* see ISO/IEC 7816-6

Further data objects such as certificate serial number, version number or expiration date may be specified by the related certification authority.

Two different structures of CV certificates are to be distinguished: self - descriptive and non self - descriptive CV certificates.

A.4 Self-descriptive CV certificates

If the content of a certificate contains BER-TLV objects, then it is called a self-descriptive CV certificate. For the signature of a certificate a digital signature scheme with or without message recovery may be used.

Table A.2 shows an example of a self-descriptive CV certificate with a digital signature scheme with message recovery.

Table A.2 - Self-descriptive certificate of a cardholder (example)

'7F21'	L	Value of subsequent DOs	
		'42' - L - ID_{CA} || '5F20' - L - $ID_{Cardholder}$ || '5F49' - L - $PK_{Cardholder}$	'5F37' - L - Digital signature of CA
Tag of the certificate (constructed)	Length of the certificate DO	Value of the certificate DO consisting of DOs to be integrated in the signature of the CA (only present, if an irreversible algorithm is used)	Signature DO. The DOs signed are '42' - L - ID_{CA} '5F20' - L - $ID_{Cardholder}$ '5F49' - L - $PK_{Cardholder}$

NOTE 1 — The IDCA may be used as reference of the public key of the certification authority.

NOTE 2 — The IDCardholder may be used for controlling access rights to data stored in the card.

NOTE 3 — The PKCardholder may be used in a subsequent VERIFY SIGNATURE operation.

A.5 Non self-descriptive CV certificates

Non self-descriptive certificates consist of a concatenation of DEs such as certificate holder reference, certificate holder public key, certification authority reference and possible additional DEs (e.g. certificate serial number).

If present an extended headerlist DO with tag '4D' describes the certificate giving the tag and length of the data objects in exactly the order as in the digital signature. The extended headerlist DO may be present in cards, to verify this type of CV certificate. If not, the headerlist should be protected when delivered to the card.

Table A.3 - Non-self-descriptive certificate of a cardholder (example)

'7F21'	L	Value of subsequent DOs		
		'4D' - L - ('42' - L || '5F20' - L || '5F49' - L)	'5F4E' - L - ID_{CA} || $ID_{Cardholder}$ || $PK_{Cardholder}$	'5F37' - L - Digital signature of CA
Tag of the certificate (constructed)	Length of the certificate DO	Extended headerlist, only present, if the certificate structure is not implicitly known	Certificate content DO containing the DEs according to the extended headerlist and to be integrated in the signature (only present, if an irreversible algorithm is used)	Signature DO. The DEs signed are - ID_{CA} - $ID_{Cardholder}$ - $PK_{Cardholder}$

Annex B
(informative)
Usage of digital signature relevant operations

B.1 Symbols (and abbreviated terms)

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

IDCA	Certification authority identifier
IDCH	ID cardholder
IV	Initial value
KR	Key reference (only valid inside a CRT)
KRPK	Key reference of the public key (only valid inside the CRT DS)
KRSK	Key reference of secret key (only valid inside the CRT DS)
KRSSK	Key reference of session key (only valid inside a CRT)
PKCH	Public key cardholder
MSE	MANAGE SECURITY ENVIRONMENT command
SENO	SE number
II	Concatenation

B.2 Command sequences of managing a security environment

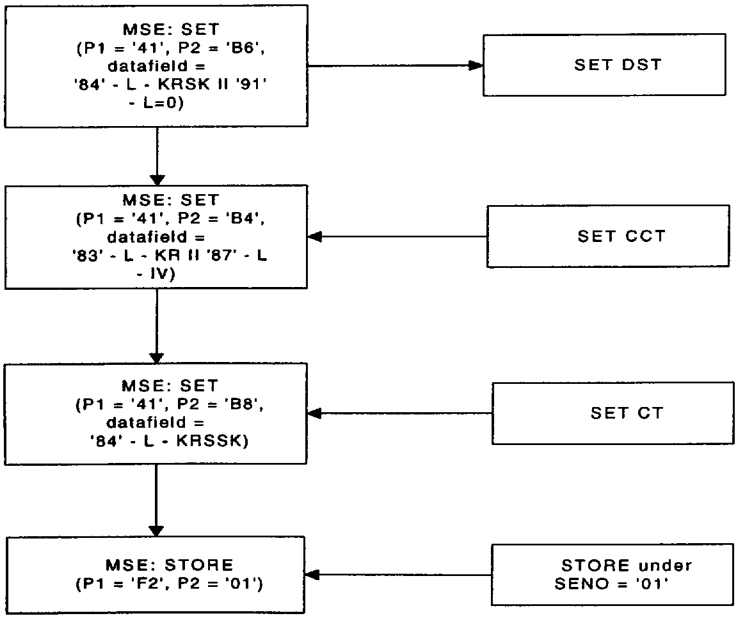
MSE: SET (P1 = '41', P2 = 'B6', datafield = '84' - L - KRSK II '91' - L=0)
SET DST
MSE: SET (P1 = '41', P2 = 'B4', datafield = '83' - L - KR II '87' - L - IV)
SET CCT
MSE: SET (P1 = '41', P2 = 'B8', datafield = '84' - L - KRSSK)
SET CT
MSE: STORE (P1 = 'F2', P2 = '01')
STORE under SENO = '01'

Figure B.1 — Setting of security environment components

In Figure B.1 a sequence of MSE commands is presented to set the DST, CCT and CT components of the current SE. Then the STORE operation is performed, which assigns to the SE a number in P2 .

In the SET operation for DS the secret key to be used in the signature computation and the integration of a random number in the DS input is specified.

B.3 Command sequences of digital signature computation

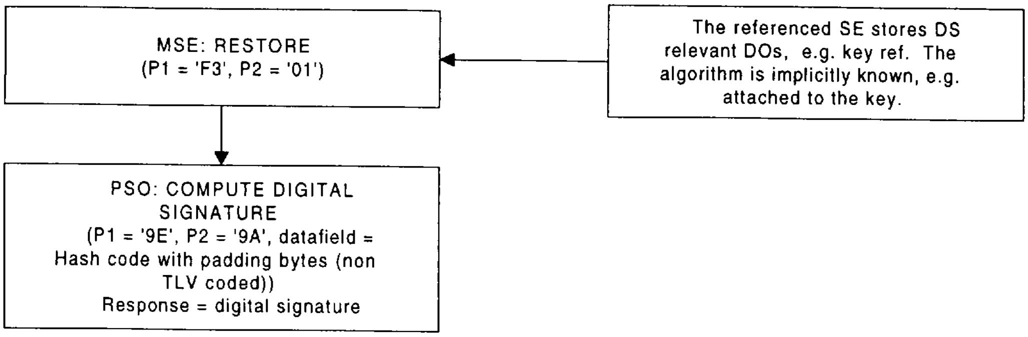
MSE: RESTORE
(P1 = 'F3', P2 = '01')
The referenced SE stores DS relevant DOs, e.g. key ref. The algorithm is implicitly known, e.g. attached to the key.
PSO: COMPUTE DIGITAL SIGNATURE
(P1 = '9E', P2 = '9A', datafield = Hash code with padding bytes (non TLV coded))
Response = digital signature

Figure B.2 — Signature scheme with appendix - data for DS input consist of hash code with padding bytes

Figure B.2 demonstrates the syntax for producing a digital signature by using a signature scheme with appendix. The input consists of the hash value complete with padding bytes. This example demonstrates the calculation of a DS with combined algorithm including a hash operation.

In this example the hash input is delivered to the card.

NOTE — This example is purely illustrative and its value is limited in terms of implementation as a result of possible export restrictions that might apply and indeed for general security reasons (avoidance of repeat signatures is desirable in some circumstances).

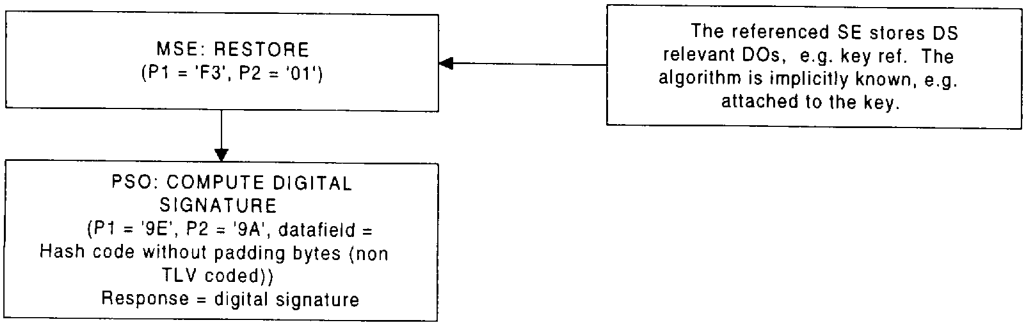
MSE: RESTORE
(P1 = 'F3', P2 = '01')
The referenced SE stores DS relevant DOs, e.g. key ref. The algorithm is implicitly known, e.g. attached to the key.
PSO: COMPUTE DIGITAL SIGNATURE
(P1 = '9E', P2 = '9A', datafield = Hash code without padding bytes (non TLV coded))
Response = digital signature

Figure B.3 — Signature scheme with appendix-data for DS-input consist of hash code without padding bytes

Figure B.3 demonstrates the syntax for producing a digital signature by using a signature scheme with appendix. The DS input consists of the hash code without padding bytes.

NOTE 1 — In order to avoid export restrictions a combined signature and hash algorithm may be used.

NOTE 2 — In some circumstances avoidance of repeat signatures, although desirable, cannot be achieved.

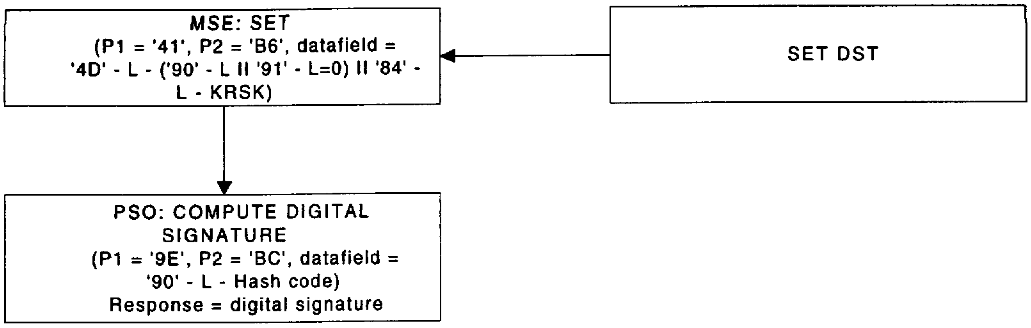
MSE: SET
(P1 = '41', P2 = 'B6', datafield =
'4D' - L - ('90' - L II '91' - L=0) II '84' -
L - KRSK)
SET DST
PSO: COMPUTE DIGITAL
SIGNATURE
(P1 = '9E', P2 = 'BC', datafield =
'90' - L - Hash code)
Response = digital signature

Figure B.4 — Signature scheme with appendix - data for DS input: hash code without padding bytes is delivered to the card, random number for padding is provided by the card

In Figure B.4 a signature scheme with appendix is applied. The data for DS input, hash code without padding bytes, is delivered to the card, and the card is requested to add a random number as required in the extended headerlist of the CRT DS in the data field of the MSE command. As specified in P2 with tag 'BC' a concatenation of DOs (hash code DO and random number DO) is signed.

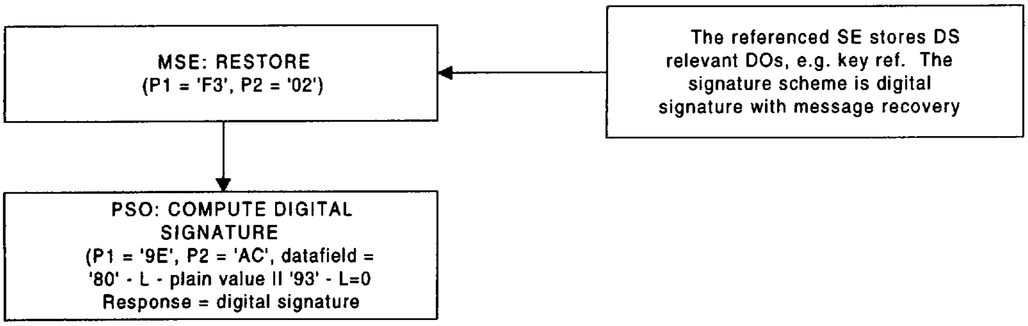
MSE: RESTORE
(P1 = 'F3', P2 = '02')
The referenced SE stores DS
relevant DOs, e.g. key ref. The
signature scheme is digital
signature with message recovery
PSO: COMPUTE DIGITAL
SIGNATURE
(P1 = '9E', P2 = 'AC', datafield =
'80' - L - plain value II '93' - L=0
Response = digital signature

Figure B.5 — Digital signature with limited message recovery - data for DS input: the plain value is delivered to the card, the digital signature counter as internal message is provided by the card

In Figure B.5 the syntax for digital signature with limited message recovery is shown. The data signed are configured in accordance with the signature scheme for limited message recovery using DOs presented in the data field, whereby the digital signature counter is used as internal message provided by the card.

NOTE — Padding for computing the hash code as well as for computing the digital signature are according to ISO/IEC 9796-2.

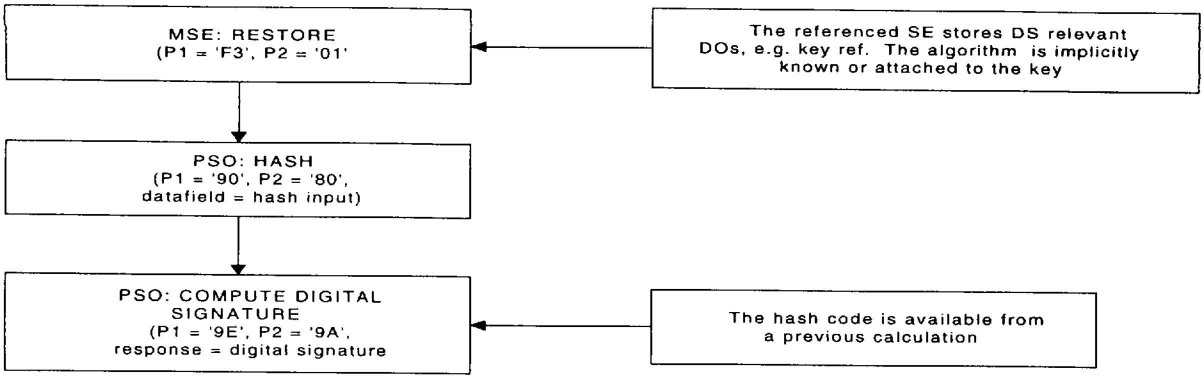
MSE: RESTORE
(P1 = 'F3', P2 = '01'
The referenced SE stores DS relevant DOs, e.g. key ref. The algorithm is implicitly known or attached to the key
PSO: HASH
(P1 = '90', P2 = '80',
datafield = hash input)
PSO: COMPUTE DIGITAL SIGNATURE
(P1 = '9E', P2 = '9A',
response = digital signature
The hash code is available from a previous calculation

Figure B.6 — Digital signature with appendix - all information for DS input is already in the card, the hash code has been computed by a previous HASH operation

In Figure B.6 the card performs the hashing (or the last round of the hash computation).The DS input is empty in the operation PSO: COMPUTE DIGITAL SIGNATURE, since all data needed for DS input are present in the card.

B.4 Command sequences of digital signature verification

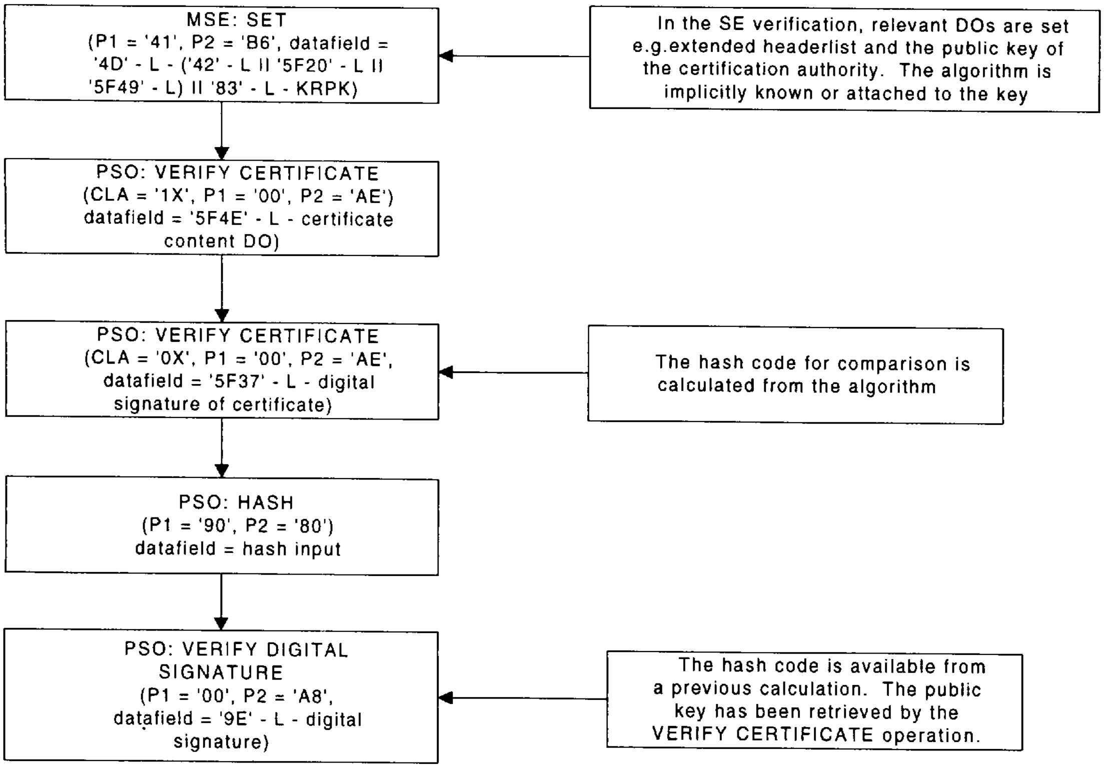
MSE: SET
(P1 = '41', P2 = 'B6', datafield =
'4D' - L - ('42' - L II '5F20' - L II
'5F49' - L) II '83' - L - KRPK)
In the SE verification, relevant DOs are set e.g.extended headerlist and the public key of the certification authority. The algorithm is implicitly known or attached to the key
PSO: VERIFY CERTIFICATE
(CLA = '1X', P1 = '00', P2 = 'AE')
datafield = '5F4E' - L - certificate content DO)
PSO: VERIFY CERTIFICATE
(CLA = '0X', P1 = '00', P2 = 'AE',
datafield = '5F37' - L - digital signature of certificate)
The hash code for comparison is calculated from the algorithm
PSO: HASH
(P1 = '90', P2 = '80')
datafield = hash input
PSO: VERIFY DIGITAL SIGNATURE
(P1 = '00', P2 = 'A8',
datafield = '9E' - L - digital signature)
The hash code is available from a previous calculation. The public key has been retrieved by the VERIFY CERTIFICATE operation.

Figure B.7 — Digital signature verification - the DS input of the certificate consists of DEs

In Figure B.7 the construction of a non self-descriptive certificate (see annex A) is provided by the extended headerlist. The VERIFY CERTIFICATE operation is performed by applying the chained command mechanism. In the first step the certificate content DO is presented (concatenation of the DEs: certification authority reference (tag ´42´), cardholder name (tag ´5F20´), and cardholder public key (tag ´5F49´)). The card performs the hashing using the certificate content as hash input.

In a second step the digital signature belonging to the certificate is re-transformed and the result is compared with the hash code computed before. After that the HASH operation is performed. For verifying the digital signature the public key has been retrieved and verified by the previous VERIFY CERTIFICATE operation. The hash input is dependent on the hash algorithm - either the plain value, possibly presented in chained commands, or a pre-processed hash code if the card performs only the last round of hash computation.

As the final step the VERIFY DS operation is performed. Where in the re-transformed signature the hash code for comparison with the previously computed hash code is located is implicitly known to the card in the given example.

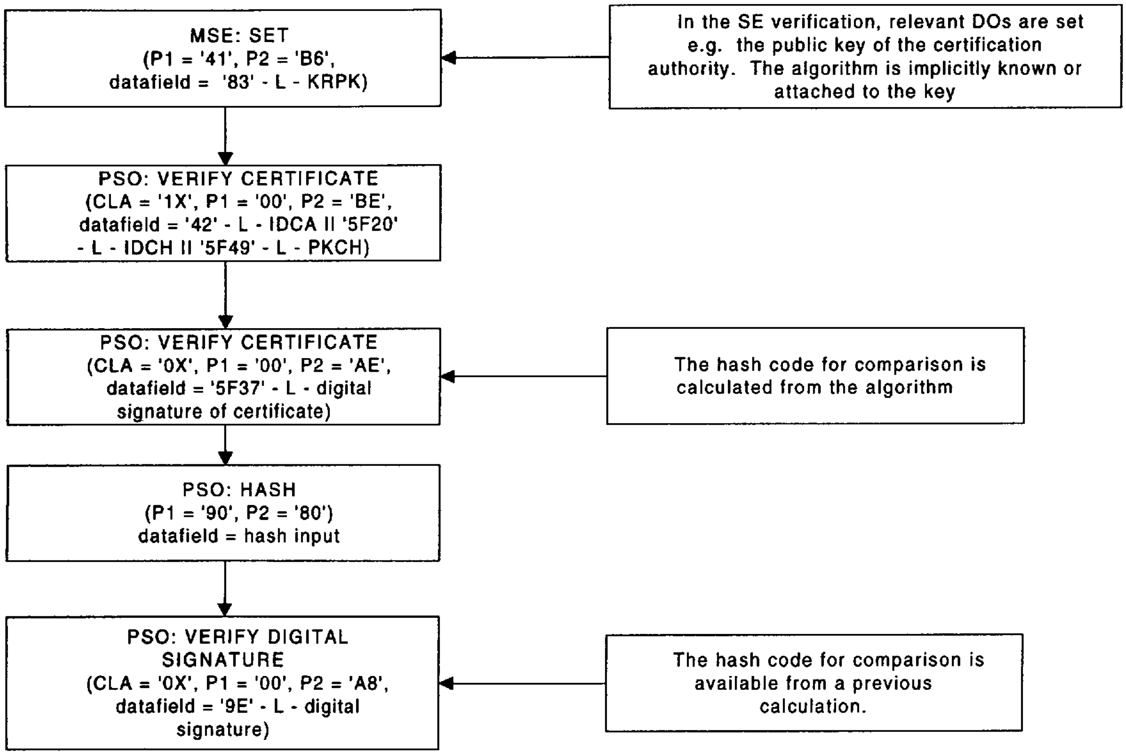
MSE: SET
(P1 = '41', P2 = 'B6',
datafield = '83' - L - KRPK)
In the SE verification, relevant DOs are set e.g. the public key of the certification authority. The algorithm is implicitly known or attached to the key
PSO: VERIFY CERTIFICATE
(CLA = '1X', P1 = '00', P2 = 'BE',
datafield = '42' - L - IDCA II '5F20' - L - IDCH II '5F49' - L - PKCH)
PSO: VERIFY CERTIFICATE
(CLA = '0X', P1 = '00', P2 = 'AE',
datafield = '5F37' - L - digital signature of certificate)
The hash code for comparison is calculated from the algorithm
PSO: HASH
(P1 = '90', P2 = '80')
datafield = hash input
PSO: VERIFY DIGITAL SIGNATURE
(CLA = '0X', P1 = '00', P2 = 'A8',
datafield = '9E' - L - digital signature)
The hash code for comparison is available from a previous calculation.

Figure B.8 — Digital signature verification - the DS input of the certificate consists of DOs

Figure B.8 shows the use of a self-descriptive certificate (see annex A) and is demonstrated for the verifying process of a DS. The VERIFY CERTIFICATE operation thereby is performed by applying the chained command mechanism. In the first step the DOs integrated in the certificate are presented (e.g. a concatenation of the DOs: certification authority reference, cardholder name and cardholder public key). The card uses this concatenation as hash input.

Further steps are identical to those of the previous example.

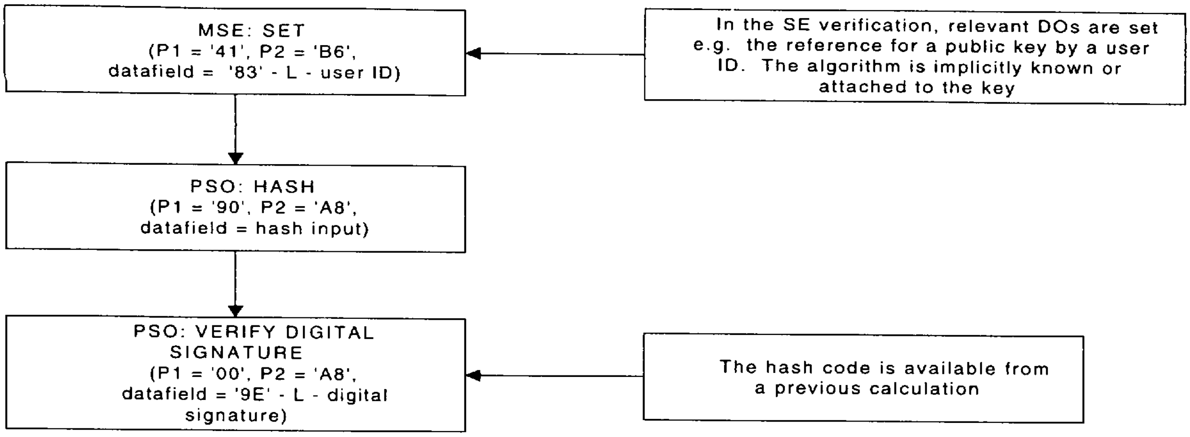
MSE: SET
(P1 = '41', P2 = 'B6',
datafield = '83' - L - user ID)
In the SE verification, relevant DOs are set e.g. the reference for a public key by a user ID. The algorithm is implicitly known or attached to the key
PSO: HASH
(P1 = '90', P2 = 'A8',
datafield = hash input)
PSO: VERIFY DIGITAL SIGNATURE
(P1 = '00', P2 = 'A8',
datafield = '9E' - L - digital signature)
The hash code is available from a previous calculation

Figure B.9 — Digital signature verification - usage of a stored PK

Figure B.9 shows the usage of a public key which has been previously installed in the card.

ICS 35.240.15

Price based on 27 pages