INTERNATIONAL STANDARD

ISO/IEC 7816-8

Second edition
2004-06-01

Identification cards — Integrated circuit cards —

Part 8: Commands for security operations

Cartes d'identification — Cartes à circuit intégré —

Partie 8: Commandes pour des operations de sécurité

Reference number
ISO/IEC 7816-8:2004(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. 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.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 7816-8 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Cards and personal identification.*

This second edition, together with the second editions of ISO/IEC 7816-4, ISO/IEC 7816-5, ISO/IEC 7816-6 and ISO/IEC 7816-9, after an in-depth reorganization of these five parts, cancels and replaces ISO/IEC 7816-4:1995, ISO/IEC 7816-5:1994, ISO/IEC 7816-6:1996, ISO/IEC 7816-8:1999 and ISO/IEC 7816-9:2000. It also incorporates the Amendments ISO/IEC 7816-4:1995/Amd.1:1997, ISO/IEC 7816-5:1994/Amd.1:1996 and ISO/IEC 7816-6:1996/Amd.1:2000 and the Technical Corrigendum ISO/IEC 7816-6:1996/Cor.1:1998.

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

— *Part 1: Cards with contacts — Physical characteristics*

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

— *Part 3: Cards with contacts — Electrical interface and transmission protocols*

— *Part 4: Organization, security and commands for interchange*

— *Part 5: Registration of application providers*

— *Part 6: Interindustry data elements for interchange*

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

— *Part 8: Commands for security operations*

— *Part 9: Commands for card management*

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

— *Part 11: Personal verification through biometric methods*

— *Part 15: Cryptographic information application*

Introduction

ISO/IEC 7816 is a series of International Standards specifying integrated circuit cards and the use of such cards for interchange. These cards are identification cards intended for information exchange negotiated between the outside world and the integrated circuit in the card. As a result of an information exchange, the card delivers information (computation result, stored data), and/or modifies its content (data storage, event memorization).

— Five parts are specific to cards with galvanic contacts and three of them specify electrical interfaces.

- ISO/IEC 7816-1 specifies physical characteristics for cards with contacts.
- ISO/IEC 7816-2 specifies dimensions and location of the contacts.
- ISO/IEC 7816-3 specifies electrical interface and transmission protocols for asynchronous cards.
- ISO/IEC 7816-10 specifies electrical interface and answer to reset for synchronous cards.
- ISO/IEC 7816-12 specifies electrical interface and operating procedures for USB cards.

— All the other parts are independent from the physical interface technology. They apply to cards accessed by contacts and/or by radio frequency.

- ISO/IEC 7816-4 specifies organization, security and commands for interchange.
- ISO/IEC 7816-5 specifies registration of application providers.
- ISO/IEC 7816-6 specifies interindustry data elements for interchange.
- ISO/IEC 7816-7 specifies commands for structured card query language.
- ISO/IEC 7816-8 specifies commands for security operations.
- ISO/IEC 7816-9 specifies commands for card management.
- ISO/IEC 7816-11 specifies personal verification through biometric methods.
- ISO/IEC 7816-15 specifies cryptographic information application.

ISO/IEC 10536 specifies access by close coupling. ISO/IEC 14443 and 15693 specify access by radio frequency. Such cards are also known as contactless cards.

Identification cards — Integrated circuit cards with contacts —

Part 8:
Commands for security operations

1 Scope

This document specifies interindustry commands that may be used for cryptographic operations.

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 document. It does not cover the internal implementation within the card and/or the outside world.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7816-4:—[1)], *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
asymmetric cryptographic technique
cryptographic technique that uses two related operations: a public operation defined by public numbers or by a public key and a private operation defined by private numbers or by a private key

NOTE The two operations have the property that, given the public operation, it is computationally infeasible to derive the private operation.

[ISO/IEC 7816-4]

3.2
certificate
digital signature binding a particular person or object and its associated public key

NOTE The entity issuing the certificate also acts as tag allocation authority with respect to the data elements in the certificate.

[ISO/IEC 7816-4]

1) To be published.

3.3
digital signature
data appended to, or cryptographic transformation of, a data string that proves the origin and the integrity of the data string and protect against forgery, e.g. by the recipient of the data string

[ISO/IEC 7816-4]

3.4
key
sequence of symbols controlling a cryptographic operation (e.g. encipherment, decipherment, a private or a public operation in a dynamic authentication, signature production, signature verification)

[ISO/IEC 7816-4]

3.5
secure messaging
set of means for cryptographic protection of [parts of] command-response pairs

[ISO/IEC 7816-4]

4 Abbreviations and notation

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

CCT	control reference template for cryptographic checksum
CRT	control reference template
CT	control reference template for confidentiality
DSA	digital signature algorithm
DST	control reference template for digital signature
ECDSA	elliptic curve digital signature algorithm
HT	control reference template for hash-code
MSE	MANAGE SECURITY ENVIRONMENT command
PK	public key
PSO	PERFORM SECURITY OPERATION command
GQ	Guillou and Quisquater
RFU	reserved for future use
RSA	Rivest, Shamir, Adleman
SE	security environment
SEID	security environment identifier

5 Interindustry commands for cryptographic operations

It shall not be mandatory for all cards complying with this document to support all those commands or all the options of a supported command.

5.1 GENERATE ASYMMETRIC KEY PAIR command

The GENERATE ASYMMETRIC KEY PAIR command either initiates the generation and storing of an asymmetric key pair, i.e., a public key and a private key, in the card, or accesses an asymmetric key pair previously generated in the card.

The command may be preceded by a MANAGE SECURITY ENVIRONMENT command in order to set key generation related parameters (e.g. algorithm reference). The command may be performed in one or several steps, possibly using command chaining (see ISO/IEC 7816-4).

Table 1 — GENERATE ASYMMETRIC KEY PAIR command-response pair

CLA	As defined in ISO/IEC 7816-4
INS	'46' or '47'
P1	Generation control according to Table 2
P2	'00' (no information provided) or reference of the key to be generated
L_c field	Absent for encoding Nc = 0, present for encoding Nc > 0
Data field	Absent, or Proprietary data if P1-P2 set to '0000', or One or more CRTs associated to the key generation if P1-P2 different from '0000' (see note)
L_e field	Absent for encoding Ne = 0, present for encoding Ne > 0

Data field	Absent, or Public key as a sequence of data elements or data objects, or Sequence of data objects according to an extended header list
SW1-SW2	See ISO/IEC 7816-4, Tables 5 and 6 where relevant, e.g. 6985

NOTE Several CRTs are present when the key pair is generated for several uses. In a data field, a CRT may have a zero length.

Table 2 — Generation control in P1

b8	b7	b6	b5	b4	b3	b2	b1	Value
0	0	0	0	0	0	0	0	No information given
1	0	0	0	0	x	x	x	Additional information given
1	0	0	0	0	-	-	x	**Key generation**
-	-	-	-	-	x	x	0	- Generate asymmetric key pair
-	-	-	-	-	x	x	1	- Access to an existing public key
1	0	0	0	0	-	x	-	**Format of returned public key data**
-	-	-	-	-	x	0	x	- Proprietary format of public key data
-	-	-	-	-	x	1	x	- Output format of public key data according to an extended headerlist
1	0	0	0	0	x	-	-	**Output indicator**
-	-	-	-	-	0	x	x	- Public key data in response data field
-	-	-	-	-	1	x	x	- No response data if Le field absent or proprietary if Le field present
Any other value is reserved for future use by ISO/IEC JTC1/SC17								

For generating a key pair, in the absence of L_e field, the key pair is stored in the card, possibly in an EF the reference of which is known before issuing the command.

For accessing a key pair (no generation), the command data field may be empty.

Depending on the parity of the INS code (see ISO/IEC 7816-4), a public key in the response data field is either a sequence of data elements ('46') or a sequence of data objects ('47').

If an extended header list describes the response data field, it is implicitly known before issuing the command. It covers public key data objects and other requested data objects.

When bit 1 is set to one in INS, i.e., INS set to '47', and when a public key is returned in the response data field, an interindustry template is used for nesting one appropriate set of public key data objects according to Table 3. If the algorithm is not indicated in the command, then the algorithm is known before issuing the command. In the public key template, the context-specific class (first byte from '80' to 'BF') is reserved for public key data objects.

Table 3 — Public key data objects

Tag	Value
'7F49'	Interindustry template for nesting one set of public key data objects with the following tags
'06'	Object identifier of the algorithm, optional
'80'	Algorithm reference as used in control reference data objects for secure messaging, optional
	Set of public key data objects for RSA
'81'	Modulus (a number denoted as n coded on x bytes)
'82'	Public exponent (a number denoted as v, e.g., 65537)
	Set of public key data objects for DSA
'81'	First prime (a number denoted as p coded on y bytes)
'82'	Second prime (a number denoted as q dividing p-1, e.g., 20 bytes)
'83'	Basis (a number denoted as g of order q coded on y bytes)
'84'	Public key (a number denoted as y equal to g to the power x mod p where x is the private key coded on y bytes)
	Set of public key data objects for ECDSA
'81'	Prime (a number denoted as p coded on z bytes)
'82'	First coefficient (a number denoted as a coded on z bytes)
'83'	Second coefficient (a number denoted as b coded on z bytes)
'84'	Generator (a point denoted as PB on the curve, coded on 2z or z+1 bytes)
'85'	Order (a prime number denoted as q, order of the generator PB, coded on z bytes)
'86'	Public key (a point denoted as PP on the curve, equal to x times PB where x is the private key, coded on 2z or z+1 bytes)
'87'	Co-factor
	Set of public key data objects for GQ2
'81'	Modulus (a number denoted as n coded on x bytes)
'83'	Number of basic numbers (a number denoted as m coded on 1 byte. If tag '83' is present, then tag 'A3' shall be absent and the m basic numbers denoted as g, g_2..g_m are the first m prime numbers 2, 3, 5, 7, 11…)
'84'	Verification parameter (a number denoted as k coded on 1 byte)
'A3'	Set of m basic numbers denoted as g, g_2..g_m, each one coded on 1 byte with tag '80'. (If tag 'A3' is present, then tag '83' shall be absent).
— In this context, ISO/IEC JTC 1/SC 17 reserves any other data object of the context-specific class (first byte in the range '80' to 'BF').	

5.2 PERFORM SECURITY OPERATION command

The PERFORM SECURITY OPERATION command initiates the following security operations, according to the data objects specified in P1-P2.

— 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.

If the security operation requires several commands to complete, then command chaining shall apply (see ISO/IEC 7816-4).

The PERFORM SECURITY OPERATION command may be preceded by a MANAGE SECURITY ENVIRONMENT command.

For example, 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.

Such a command can be performed only if the security status satisfies the security attributes for the operation. 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).

If present, a header list or an extended header list defines the order and the data items that form the input for the security operation.

Table 4 — PERFORM SECURITY OPERATION command-response pair

CLA	As defined in ISO/IEC 7816-4
INS	'2A'
P1	Tag (the response data field is the data element, if present) or '00' (the response data field is always absent); 'FF' is RFU
P2	Tag (the command data field is the data element, if present) or '00' (the command data field is always absent); 'FF' is RFU for ISO/IEC JTC1/SC17
L_c field	Absent for encoding Nc = 0, present for encoding Nc > 0
Data field	Absent or value of the data object specified in P2
L_e field	Absent for encoding Ne = 0, present for encoding Ne > 0

Data field	Absent or value of the data object specified in P1
SW1-SW2	See ISO/IEC 7816-4, Tables 5 and 6 where relevant, e.g. 6985

The PERFORM SECURITY OPERATION command uses input templates listed in Table 5. These templates are basic data objects for secure messaging (see ISO/IEC 7816-4).

Table 5 — Input templates

Tag	Value
'A0'	Input template for the computation of a hash-code (the template is hashed))
'A2'	Input template for the verification of a cryptographic checksum (the template is integrated)
'A8'	Input template for the verification of a digital signature (the template is signed)
'AC'	Input template for the computation of a digital signature (the concatenated value fields are signed)
'AE'	Input template for the verification of a certificate (the concatenated value fields are certified)
'BC'	Input template for the computation of a digital signature (the template is signed)
'BE'	Input template for the verification of a certificate (the template is certified)

In the input templates, the context-specific class (first byte in the range '80' to 'BF') is reserved for input data objects. Table 6 lists data objects in the input templates.

Table 6 — Input data objects

Tag	Value	'A0'	'A2'	'A8'	'AC', 'BC'	'AE', 'BE'
'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

5.3 COMPUTE CRYPTOGRAPHIC CHECKSUM operation

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

Table 7 — Parameters and data fields for COMPUTE CRYPTOGRAPHIC CHECKSUM operation

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

Response data field	Cryptographic checksum

5.4 COMPUTE DIGITAL SIGNATURE operation

The COMPUTE DIGITAL SIGNATURE operation initiates the computation of a digital signature. The algorithm may be either a digital signature algorithm or a combination of a hash algorithm and a digital signature algorithm. Annex A provides examples of digital signature operations.

For the computation of a digital signature, the data to be signed or integrated in the signing process are transmitted in the command data field or submitted in a previous command e.g. PSO: HASH. In P2, the digital signature is specified with tags '9A', 'AC' or 'BC' according to the structure of the input (see ISO/IEC 7816-4).

If auxiliary data are to be included in the digital signature input, then a reference shall be present in the CRT (see ISO/IEC 7816-4). If an empty reference data object for auxiliary data is present, then the card shall insert the auxiliary data. Auxiliary data present or referenced in the command data field take precedence over any header list.

The card shall return a digital signature (P1 = '9E').

Table 8 — Parameters and data fields for COMPUTE DIGITAL SIGNATURE operation

P1	'9E'
P2	'9A', 'AC' or 'BC'
Command data field	Absent (data already in the card) or If P2 = '9A', data to be signed or integrated in the signature process, or If P2 = 'AC', data objects, the value fields of which are signed or integrated in the signature process, or If P2 = 'BC', data objects to be signed or integrated in the signature process
Response data field	Digital signature

NOTE — The tags 'AC' and 'BC' are not integrated in the digital signature input.

5.5 HASH operation

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

— either the complete computation inside the card or

— a partial computation inside the card (e.g. last round of hashing).

The HT ('AA', 'AB') indicates the algorithm reference for computing a hash-code (see ISO/IEC 7816-4).

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

For the resulting hash-code, the following two cases have to be distinguished:

— either the card stores the hash-code for a subsequent command; then the L_e field is not present or

— the card delivers the hash-code in the response; then the L_e field has to be set to the appropriate length.

Table 9 — Parameters and data fields for HASH operation

P1	'90'
P2	'80', or 'A0'
Command data field	If P2 = '80', data to hash, or If P2 = 'A0', data objects relevant for hashing (e.g., '90' for intermediate hash-code, '80' for last block)
Response data field	Hash-code or absent

5.6 VERIFY CRYPTOGRAPHIC CHECKSUM operation

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

Table 10 — Parameters and data fields for VERIFY CRYPTOGRAPHIC CHECKSUM operation

P1 P2	'00' 'A2'
Command data field	Data objects relevant to the operation (e.g., '80', '8E')

Response data field	Absent

NOTE The value field of the plain value data object (tag '80') contains the data (data elements or data objects) covered by the cryptographic checksum.

5.7 VERIFY DIGITAL SIGNATURE operation

The VERIFY DIGITAL SIGNATURE operation initiates the verification of a digital signature delivered as a data object in the command data field. Other verification relevant data are either transmitted in a command chaining process or present in the card. The algorithm may be either a digital signature algorithm or a combination of a hash algorithm and a digital signature algorithm. Annex A provides examples of digital signature operations.

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 previous VERIFY CERTIFICATE operation.

If the algorithm reference in the card declares a signature only algorithm then the data consists of a hash-code, or the signature is of message recovery type (see 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.

Table 11 — Parameters and data fields for VERIFY DIGITAL SIGNATURE operation

P1 P2	'00' 'A8'
Command data field	Data objects relevant to the operation (e.g. either '9A', 'AC' or 'BC', and '9E')

Response data field	Absent

If the command data field contains an empty data object, then the card is expected to know its value for use in the verification.

5.8 VERIFY CERTIFICATE operation

For the verification of a certificate in a card (see Annex B), the digital signature of a certificate to be verified is delivered as a data object in the command 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 apply is implicitly known or may be referenced in a DST. If other data objects are to be used in the verification process (e.g. hash-code) then these data objects shall be present in the card or shall be transmitted using the command chaining process.

The following two cases have to be distinguished.

— If the certificate is self-descriptive (P2 = 'BE'), then the card retrieves a public key identified by its tag in the (recovered) certificate content.

— If the certificate is not self-descriptive (P2 = 'AE'), then the card retrieves a public key in the certificate either implicitly or explicitly by using the public key tag in a header-list describing the content of the certificate.

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

Table 12 — Parameters and data fields for VERIFY CERTIFICATE operation

P1 P2	'00' '92', 'AE' or 'BE'
Command data field	Data elements or data objects relevant to the operation)
Response data field	Absent

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

5.9 ENCIPHER operation

The ENCIPHER operation enciphers data transmitted in the command data field. The usage of this operation may be restricted.

NOTE — The operation may be used for generating diversified keys.

Table 13 — Parameters and data fields for ENCIPHER operation

P1 P2	'82', '84', '86' (cryptogram) '80' (plain value)
Command data field	Absent (data already in the card) or Data to be enciphered
Response data field	Enciphered data

5.10 DECIPHER operation

The DECIPHER operation deciphers data transmitted in the command data field. The usage of this operation may be restricted.

Table 14 — Parameters and data fields for DECIPHER operation

P1	'80' (plain value)
P2	'82', '84', '86' (cryptogram)
Command data field	Data to be deciphered
Response data field	Absent (deciphered data remains in the card) or
	Deciphered data

Annex A
(informative)

Examples of operations related to digital signature

A.1 Sequences of commands for managing a security environment

Table A.1 presents a sequence of MANAGE SECURITY ENVIRONMENT commands to SET DST, CCT and CT components of the current SE and finally to STORE the current SE under a SEID indicated in P2.

Table A.1 — Setting of security environment components

Command	Operation	P1-P2	Command data field
MSE	SET DST	'41' - 'B6'	{'84' – L - Key reference} - {'91' - L = 0}
MSE	SET CCT	'41' - 'B4'	{'83' - L - Key reference} - {'87' - L - Initialization value}
MSE	SET CT	'41' - 'B8'	{'83' – L - Key reference}
MSE	STORE (SEID = 1)	'F2' - '01'	-

The SET DST operation references the private key to use in the signature computation and specifies the integration of a random number in the digital signature input. The SET CCT operation references a secret key and an initial value to use for the computation of a cryptographic checksum. The SET CT operation references a secret session key to use for confidentiality.

A.2 Sequences of commands for digital signature computation

Table A.2 shows the syntax for producing a digital signature by using a signature scheme with appendix. The input is a hash-code completed with padding bytes. This example illustrates the calculation of a digital signature with combined algorithm including a hash operation. In this example the hash input is delivered to the card.

Table A.2 — First example of digital signature scheme with appendix

Command	Operation	P1-P2	Command data field	Response data field
MSE	RESTORE	'F3' - '01'		
PSO	COMPUTE DIGITAL SIGNATURE	'9E' - '9A'	Hash-code with padding bytes	Digital signature

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

Table A.3 shows the syntax for producing a digital signature by using a signature scheme with appendix. The digital signature input consists of the hash-code without padding bytes.

Table A.3 — Second example of digital signature scheme with appendix

Command	Operation	P1-P2	Command data field	Response data field
MSE	RESTORE	'F3 - '01'	-	-
PSO	COMPUTE DIGITAL SIGNATURE	'9E' - '9A'	Hash-code without padding bytes	Digital signature

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.

Table A.4 shows a signature scheme with appendix. The digital signature input contains a hash-code without padding bytes delivered to the card and the card is requested to generate a random number as required in the extended header-list of the DST in the command data field of the MSE command. As specified by tag 'BC' in P2, a concatenation of data objects (hash-code provided to the card and random number provided by the card) is signed.

Table A.4 — Third example of digital signature scheme with appendix

Command	Operation	P1-P2	Command data field	Response data field
MSE	SET	'41' - 'B6'	{'4D' - L - ('90' - L - '91' - L=0)} - {'84' - L - Key reference}	-
PSO	COMPUTE DIGITAL SIGNATURE	'9E' - 'BC'	{'90' - L - Hash-code}	Digital signature

Table A.5 shows the syntax for digital signature with limited message recovery. The data to sign are configured in accordance with a signature scheme giving limited message recovery using data objects presented in the command data field, whereby the digital signature counter is used as internal message provided by the card.

Table A.5 — Fourth example of digital signature scheme with appendix

Command	Operation	P1-P2	Command data field	Response data field
MSE	RESTORE	'F3' - '02'	-	-
PSO	COMPUTE DIGITAL SIGNATURE	'9E' - 'AC'	{'90' – L - Hash-code}	Digital signature

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

In Table A.6, the card performs the hashing (or the last round of the hash computation). The digital signature input is empty in the COMPUTE DIGITAL SIGNATURE operation, since all input data are present in the card.

Table A.6 — Fifth example of digital signature scheme with appendix

Command	Operation	P1-P2	Command data field	Response data field
MSE	RESTORE	'F3' - '01'	-	-
PSO	HASH	'90' - '80'	Data to hash	-
PSO	COMPUTE DIGITAL SIGNATURE	'9E' - '9A'	-	Digital signature

A.3 Sequences of commands for digital signature verification

In Table A.7, an extended header list specifies the construction of a non self-descriptive certificate (see annex B): the digital signature input consists of data elements. The VERIFY CERTIFICATE operation uses command chaining.

Table A.7 — First example of digital signature verification

Command	Operation	P1-P2	Command data field
MSE	SET DST	'41' - 'B6'	{'4D' – L - ('42' – L - '5F20' – L - '5F49' - L)} - {'83' – L - Key reference}
PSO	VERIFY CERTIFICATE (CLA='1X')	'00' - 'AE'	{'5F4E' - L - Certificate content}
PSO	VERIFY CERTIFICATE (CLA='0X')	'00' - 'AE'	{'5F37' - L - Digital signature of certificate}
PSO	HASH	'90' - '80'	Hash input
PSO	VERIFY DIGITAL SIGNATURE	'00' - 'A8'	{'9E' – L - Digital signature}

— As the first step, the certificate content data object is presented (concatenation of the data elements: issuer identification number (tag '42'), cardholder name (tag '5F20'), and cardholder public key (tag '5F49')). The card performs the hashing using the certificate content as hash input.

— As a second step, the digital signature belonging to the certificate is re-transformed and the result is compared with the hash-code computed before. Then 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 DIGITAL SIGNATURE operation is performed.

Table A.8 shows the verification of a self-descriptive certificate (see annex B): the digital signature input consists of data objects. The VERIFY CERTIFICATE operation uses command chaining. In the first step the data objects integrated in the certificate are presented (e.g. a concatenation of the data objects: 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.

Table A.8 — Second example of digital signature verification

Command	Operation	P1-P2	Command data field
MSE	SET DST	'41' - 'B6'	{'83' - L - Key reference}
PSO	VERIFY CERTIFICATE (CLA='1X')	'00' - 'BE'	{'42' - L - Issuer identification number} - {'5F20' - L - Cardholder name} - {'5F49' - L - cardholder public key}
PSO	VERIFY CERTIFICATE (CLA='0X')	'00' - 'AE'	{'5F37'-L-Digital signature of certificate}
PSO	HASH	'90' - '80'	Hash input
PSO	VERIFY DIGITAL SIGNATURE	'00' - 'A8'	{'9E'-L-Digital signature}

Table A.9 shows the usage of a public key previously installed in the card.

Table A.9 — Third example of digital signature verification

Command	Operation	P1-P2	Command data field
MSE	SET DST	'41' - 'B6'	{'83' - L - Key reference}
PSO	HASH	'90' - 'A8'	Hash input
PSO	VERIFY DIGITAL SIGNATURE	'00' - 'A8'	{'9E' - L - Digital signature}

Annex B
(informative)

Examples of certificates interpreted by the card

B.1 Data objects for card-verifiable certificates

Table B.1 shows data objects relevant for card-verifiable certificates.

Table B.1 — Interindustry data objects (examples) relevant for card-verifiable certificates

Tag	Data element
'42'	Issuer identification number
'5F20'	Cardholder name
'5F37'	Static internal authentication (signature of a certificate, produced by the issuer)
'5F49'	Cardholder public key
'5F4C'	Certificate holder authorization
'5F4E'	Certificate content
'7F21'	Cardholder certificate

The issuer may specifies further data objects such as certificate serial number, version number, expiration date, etc.

Two different structures of card-verifiable certificates are to be distinguished:

— a self-descriptive card-verifiable certificate consists of a concatenation of BER-TLV data objects;

— a non self-descriptive card-verifiable certificate consists of a concatenation of data elements.

B.2 Self-descriptive card-verifiable certificates

For the signature of a certificate, a digital signature scheme with or without message recovery may be used. Table B.2 shows an example of a self-descriptive card-verifiable certificate with a digital signature scheme with message recovery.

Table B.2 — Self-descriptive card-verifiable certificate of a cardholder (example)

'7F21'	Length	Value of subsequent data objects	
		{'42' - L - Issuer identification number} - {'5F20' - L - Cardholder name} - {'5F49' - L - Cardholder public key}	{'5F37' - L - Digital signature}
Tag of the certificate (constructed)	Length of the certificate	Value of the certificate consisting of data objects integrated in the digital signature (present only in the absence of message recovery)	The data objects are signed: {'42' - L - Issuer identification number} {'5F20' - L - Cardholder name} {'5F49' - L - Cardholder public key}

NOTE 1 — The identification data of the certification authority may reference his public key.

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

NOTE 3 — The public key of the cardholder may be used in a subsequent VERIFY DIGITAL SIGNATURE operation.

B.3 Non self-descriptive card-verifiable certificates

An extended header-list data object may be present in the card to verify this type of certificate; otherwise, it should be protected when delivered to the card. An extended header-list data object (tag '4D', see ISO/IEC 7816-4) describes the concatenation of data elements by tag / length pairs in the same order as in the digital signature.

Table B.3 — Non-self-descriptive card-verifiable certificate of a cardholder (example)

'7F21'	Length	Value of subsequent data objects		
		{'4D' - L - ('42' - L - '5F20' - L - '5F49' - L)}	{'5F4E' - L - Issuer identification number - Cardholder name - Cardholder public key}	{'5F37' - L – Digital signature}
Tag of the certificate (constructed)	Length of the certificate	Extended header-list (present only if the certificate structure is not implicitly known)	Certificate content data object integrated in the signature (present only in the absence of message recovery, it contains the data elements according to the extended header-list)	The data elements are signed: - Issuer identification number - Cardholder name - Cardholder public key

Annex C
(informative)

Examples of asymmetric key import/export

C.1 Usage of the GET DATA command for public key export

It is assumed, that the data objects describing a Public Key (PK) are present in the card coded in a form as shown in Table C.1.

Table C.1 — Coding for PK data objects present in the card

‘A8’	L	T-L pair to indicate a template for digital signature verification				
		‘B6’	L	DST		
				‘83’	L	Key reference to PK.CH.DS
		‘7F49’	L	Public key data object		
				‘81’	L	Modulus
				‘82’	L	Public exponent
		‘9E’	L	Digital signature (all bytes of the digital signature verification template preceding tag ’9E’ are signed)		

With the MSE command, the PK to be retrieved is selected. Then the GET DATA command (odd INS, P1-P2 = ‘3FFF’) is used in 3 steps, whereby the data fields shown in Tables C.2 – C.7 occur at the card interface.

Table C.2 — Data field of the GET DATA command, step 1 of 3

‘4D’	'0B'	Extended header list						
		‘A8’	09	T-L pair to indicate a template for digital signature verification				
				‘B6’	02	T-L pair that indicates a DST data object		
						'83'	00	T-L pair that indicates a public key reference
		‘7F49’	02	T-L pair that indicates the public key data object				
				‘81’	00	T-L pair that indicates the modulus		

Table C.3 — Data field of the GET DATA response, step 1 of 3

‘A8’	L					
		‘B6’	L	DST		
				‘83’	L	Key reference PK.CH.DS
		‘7F49’	L	Public key		
				‘81’	L	Modulus

Table C.4 — Data field of the GET DATA command, step 2 of 3

‘4D’	07	Extended header list						
		‘A8’	07	T-L pair to indicate a template for digital signature verification				
				‘7F49’	02	T-L pair that indicates the public key data object		
						'82'	00	T-L pair that indicates the modulus

Table C.5 — Data field of the GET DATA response, step 2 of 3

‘A8’	L				
	‘7F49’	L	Public key		
			‘82’	L	Public exponent

Table C.6 — Data field of the GET DATA command, step 3 of 3

‘4D’	04	Extended header list			
	‘A8’	02	T-L pair to indicate a template for digital signature verification		
			‘9E’	00	T-L pair that indicates digital signature data object

Table C.7 — Data field of the GET DATA response, step 3 of 3

‘A8’	L		
	‘9E’	L	Digital signature

C.2 Usage of the PUT DATA command for private key import

Initially, an MSE command shall be send to reference the corresponding private key (i.e. the key reference is already known to the card). Then the PUT DATA command (odd INS, P1-P2 = ‘3FFF’) is used with command data field shown in Tables C.9.

Table C.8 — Extended headerlist describing the private key object

‘4D’	L	Extended header list						
	‘A8’	L	T-L pair to indicate a template for digital signature verification					
			‘B6’	L	T-L pair to indicate a DST			
					'84'	L	T-L pair to indicate a key reference to SK.CH.DS	
			‘7F48’	L	T-L pair to indicate a private key data object			
					'92'	L	T-L pair for parameter p	
					'93'	L	T-L pair for parameter q	
					'94'	L	T-L pair for parameter 1/q mod p	
					'95'	L	T-L pair for parameter d mod (p – 1)	
					'96'	L	T-L pair for parameter d mod (q –1)	
			‘9E’	L	T-L pair to indicate a digital signature			

Table C.9 — Data field of the PUT DATA command

‘4D’	L	Extended header list					
		‘A8’	L	T-L pair to indicate a template for digital signature verification			
				‘B6’	L	T-L pair to indicate a DST	
						‘84’	L	T-L pair to indicate a key reference to SK.CH.DS
				‘7F48’	L	T-L pair to indicate a private key data object	
						‘92’	L	T-L pair for parameter p
						‘93’	L	T-L pair for parameter q
						‘94’	L	T-L pair for parameter 1/q mod p
						‘95’	L	T-L pair for parameter d mod (p-1)
						‘96’	L	T-L pair for parameter d mod (q-1)
				‘9E’	L	T-L pair to indicate a digital signature	
'5F48'	L	Concatenation of the key parameter data elements according to the extended header list. Data elements that are associated to filler tags ‘00’ in the extended header list are read, but ignored.					
'9E'	L	Digital signature					

Bibliography

[1] ISO/IEC 7816 (all parts), *Identification cards — Integrated circuit cards*

[2] ISO/IEC 9796 (all parts), *Information technology — Security techniques — Digital signature scheme giving message recovery*

[3] ISO/IEC 9798-5:1999[2)], *Information technology — Security techniques — Entity authentication — Part 5: Mechanisms using zero knowledge techniques*

[4] ISO/IEC 10536 (all parts), *Identification cards — Contactless integrated circuits cards — Close coupled cards*

[5] ISO/IEC 14443 (all parts), *Identification cards — Contactless integrated circuits cards — Proximity cards*

[6] ISO/IEC 15693 (all parts), *Identification cards — Contactless integrated circuits cards — Vicinity cards*

2) To be published.

ICS 35.240.15

Price based on 19 pages