

Application Interface for smart cards used as Secure Signature Creation Devices

Part 1: Basic services

ICS 35.240.15

National foreword

This British Standard is the UK implementation of EN 14890-1:2008.

The UK participation in its preparation was entrusted to Technical Committee IST/17, Cards and personal identification.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2009

© BSI 2009

ISBN 978 0 580 58269 1

Amendments/corrigenda issued since publication

Date	Comments

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 14890-1

December 2008

ICS 35.240.15

Supersedes CWA 14890-1:2004

English Version

**Application Interface for smart cards used as Secure Signature
Creation Devices - Part 1: Basic services**

Interface applicative des cartes à puces utilisées comme
dispositifs de création de signature numérique sécurisés -
Partie 1 : Services de bases

Anwendungsschnittstelle für Chipkarten, die zur Erzeugung
qualifizierter elektronischer Signaturen verwendet werden -
Teil 1: Allgemeine Dienste

This European Standard was approved by CEN on 27 September 2008.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

© 2008 CEN All rights of exploitation in any form and by any means reserved
worldwide for CEN national Members.

Ref. No. EN 14890-1:2008: E

Contents

Foreword.....	6
1 Scope	7
2 Normative references	8
3 Terms and definitions	8
4 Symbols and abbreviations	11
5 Signature application	13
5.1 Application Flow	13
5.2 Trusted environment versus untrusted environment	16
5.3 Selection of E-SIGN application	16
5.3.1 General.....	16
5.3.2 Exceptions for Secure Messaging	17
5.4 Selection of cryptographic information application	17
5.5 Concurrent usage of signature applications	17
5.5.1 General.....	17
5.5.2 Methods of channel selection	17
5.5.3 Security issues on multiple channels	17
5.6 Security environment selection	18
5.7 Key selection.....	18
5.8 Basic Security Services	18
6 User verification.....	19
6.1 General.....	19
6.2 Knowledge based user verification	19
6.2.1 General.....	19
6.2.2 Explicit user verification	20
6.2.3 Password related mechanisms	20
6.2.4 Presentation formats	21
6.2.5 Retry counters.....	21
6.2.6 Password Change.....	21
6.2.7 Reset of RC and setting a new password	22
6.3 Biometric user verification	23
6.3.1 General.....	23
6.3.2 Retrieval of the Biometric Information Template	24
6.3.3 Performing the biometric user verification	25
6.3.4 Reset of RC.....	27
7 Digital Signature Service	28
7.1 Signature generation algorithms	28
7.2 Activation of digital signature service.....	28
7.3 General aspects	29
7.4 Signature Generation	30
7.4.1 No hashing in Card	30
7.4.2 Partial hashing	31
7.4.3 All hashing in Card	32
7.5 Selection of different keys, algorithms and input formats	33
7.5.1 Restore an existing SE.....	34
7.5.2 Modify the HT of a current SE	34
7.5.3 Modify the DST of a current SE	35
7.6 Read certificates and certificate related information.....	36
7.6.1 Read certificate related CIOs.....	36
7.6.2 Read signer's certificate from ICC	36
7.6.3 Retrieval of the signer's certificate from a directory service	37

8	Device authentication	38
8.1	Certification authorities and certificates.....	38
8.1.1	Certificate chains.....	38
8.1.2	Usage of cross certificates.....	39
8.2	Authentication environments.....	40
8.2.1	SCA in trusted environment.....	40
8.2.2	SCA in untrusted environment	41
8.2.3	Specification of the environment.....	41
8.2.4	Display message mechanism	41
8.2.5	Additional authentication environments.....	41
8.3	Key transport and key agreement mechanisms	41
8.4	Key transport protocol based on RSA	42
8.4.1	Authentication Steps.....	43
8.4.2	Session Key creation	52
8.5	Device authentication with privacy protection.....	52
8.5.1	Authentication steps	53
8.6	Privacy constrained Modular EAC (mEAC) protocol with non-traceability feature (based on elliptic curves)	70
8.6.1	Example for traceability case	70
8.6.2	Notation	70
8.6.3	Authentication steps	71
8.7	Asymmetric Authentication summary.....	82
8.8	Symmetric authentication scheme	83
8.8.1	Authentication steps	83
8.8.2	Session Key creation	86
8.9	Compute Session keys from key seed $K_{IFD/ICC}$	87
8.9.1	Compute TDES session keys	87
8.9.2	Compute AES-128 session keys for CBC mode and EMAC	88
8.9.3	Compute AES-128 session keys for CBC mode and CMAC	88
8.10	Compute send sequence counter SSC	89
8.11	Post-authentication phase.....	89
8.12	Ending the secure session	89
8.12.1	Example for ending a secure session	89
8.12.2	Rules for ending a secure session	89
8.13	Reading the Display Message	90
8.14	Updating the Display Message	92
9	Secure messaging	93
9.1	CLA byte	93
9.2	TLV coding of command and response message	93
9.3	Treatment of SM-Errors	94
9.4	Padding for checksum calculation	94
9.5	Send sequence counter (SSC)	94
9.6	Message structure of Secure Messaging APDUs	95
9.6.1	Cryptograms	95
9.6.2	Cryptographic Checksums.....	97
9.6.3	Final command APDU construction	100
9.7	Response APDU protection.....	101
9.8	Use of TDES and AES	107
9.8.1	TDES/AES encryption/decryption.....	107
9.8.2	CBC mode	108
9.8.3	Retail MAC with TDES.....	108
9.8.4	EMAC with AES	109
9.8.5	CMAC with AES	110
10	Key Generation	111
10.1	Key generation and export using PrK.ICC.AUT	111
10.2	Key generation and export with dynamic or static SM.....	111
10.3	Write certificates.....	112
10.4	Setting keys in static secure messaging	112
11	Key identifiers and parameters.....	112

11.1	Key identifiers (KID).....	112
11.2	Public Key parameters	112
11.3	DSA with ELC public key parameters.....	113
11.4	RSA Diffie-Hellman key exchange parameters.....	114
11.5	ELC key exchange parameters.....	114
12	Data structures.....	115
12.1	CRTs.....	115
12.1.1	CRT AT for selection of internal authentication keys	115
12.1.2	CRT for selection of IFD's PuK.CA _{IFD} .CS_AUT	115
12.1.3	CRT for selection of IFD's PuK.IFD.AUT	116
12.1.4	CRT AT for selection of the public DH key parameters	116
12.1.5	GENERAL AUTHENTICATE DH key parameters	116
12.1.6	CRT AT for selection of ICC's private authentication key	116
12.1.7	CRT for selection of IFD's PuK.IFD.AUT	117
12.1.8	CRT for selection of PrK.ICC.KA.....	117
12.2	Key transport device authentication protocol	117
12.2.1	EXTERNAL AUTHENTICATE	117
12.2.2	INTERNAL AUTHENTICATE.....	118
12.3	Privacy device authentication protocol.....	119
12.3.1	EXTERNAL AUTHENTICATE	119
12.3.2	INTERNAL AUTHENTICATE.....	120
13	AlgIDs, Hash- and DSI Formats.....	121
13.1	Algorithm Identifiers and OIDs.....	121
13.2	Hash Input-Formats	122
13.2.1	PSO:HASH without command chaining	123
13.2.2	PSO:HASH with command Chaining	124
13.3	Formats of the Digital Signature Input (DSI).....	124
13.3.1	DSI according to ISO/IEC 14888-2 (scheme 2).....	124
13.3.2	DSI according to PKCS #1 V 1.5.....	125
13.3.3	Digest Info for SHA-X	127
13.3.4	DSI according to PKCS #1 V 2.x.....	128
13.3.5	DSA with DH key parameters	130
13.3.6	Elliptic Curve Digital Signature Algorithm - ECDSA	130
14	CV_Certificates and Key Management	130
14.1	Level of trust in a certificate	130
14.2	Key Management	130
14.3	Card Verifiable Certificates.....	131
14.3.1	Signature-Certificates	132
14.3.2	Authentication Certificates	132
14.4	Use of the public key extracted from the certificate	132
14.5	Validity of the key extracted from a certificate	132
14.6	Structure of CVC	133
14.6.1	Non-self-descriptive certificates	133
14.6.2	Self-descriptive certificates	134
14.7	Certificate Content.....	134
14.7.1	CPI-Certificate Profile Identifier.....	135
14.7.2	CAR-Certification Authority Reference	136
14.7.3	CHR-Certificate Holder Reference	137
14.7.4	CHA-Certificate Holder Authorization (CHA)	138
14.7.5	Role identifier specifications	139
14.7.6	CHAT-Certificate Holder Authorization Template (CHAT)	142
14.7.7	OID — Object identifier	142
14.7.8	CED — Certificate Effective Date	144
14.7.9	CXD — Certificate Expiration date	144
14.8	Certificate signature	144
14.8.1	Non self-descriptive certificates	144
14.8.2	Self descriptive certificates	146
14.9	Coding of the certificate content.....	146
14.9.1	Non self-descriptive certificates	146

14.9.2 Self-descriptive certificates 147

14.9.3 Self-descriptive certificates for elliptic curve cryptography..... 147

14.10 Steps of CVC verification..... 150

14.10.1 First round: CVC verification from a Root PuK..... 151

14.10.2 Subsequent round(s) 152

14.11 Commands to handle the CVC 152

14.12 C_CV.IFD.AUT (non self-descriptive) 152

14.13 C_CV.CA.CS-AUT (non self-descriptive)..... 154

14.14 C.ICC.AUT..... 155

14.15 Self-descriptive CV Certificate (Example)..... 155

14.15.1 Public Key 156

15 Files..... 157

15.1 File structure 157

15.2 File IDs 158

15.3 EF.DIR..... 158

15.4 EF.SN.ICC 158

15.5 EF.DH 159

15.6 EF.ELC 159

15.7 EF.C.ICC.AUT 160

15.8 EF.C.CA_{ICC}.CS-AUT 160

15.9 EF.C_X509.CH.DS 161

15.10 EF.C_X509.CA.CS (DF.ESIGN) 161

15.11 EF.DM..... 161

16 Cryptographic Information Application 162

16.1 ESIGN cryptographic information layout example 163

16.1.1 EF.CIAInfo 164

16.1.2 EF.AOD 165

16.1.3 EF.PrKD 168

16.1.4 EF.PuKD 170

16.1.5 EF.CD 171

16.1.6 EF.DCOD..... 172

Annex A (informative) Device authentication — Cryptographic view 175

A.1 Algorithms for authentication with key exchange or key negotiation..... 175

A.2 Device authentication with key transport 175

A.2.1 Conformance to ISO/IEC 11770-3 175

A.2.2 Using min(SIG, N-SIG) for the signature token 177

A.3 Device authentication with key negotiation..... 178

A.3.1 Diffie-Hellman Key Exchange..... 179

A.4 Device authentication with privacy protection..... 181

A.4.1 The authenticity of the public DH parameters..... 183

A.5 Device authentication with non traceability 185

A.5.1 Diffie-Hellman Key Exchange..... 185

A.6 The “Grandmaster Chess Attack” 187

Annex B (informative) Personalization scenarios 189

Annex C (informative) Build scheme for mEAC Object Identifiers 191

Bibliography..... 193

Foreword

This document (EN 14890-1:2008) has been prepared by Technical Committee CEN/TC 224 “xxx”, the secretariat of which is held by AFNOR.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by June 2009, and conflicting national standards shall be withdrawn at the latest by June 2009.

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

This document supersedes CWA 14890-1:2004.

This standard additionally provides generic Identification, Authentication and Digital Signature (IAS) services and thereby contains all additional cryptographic services specified in CWA 14890-2.

The standard will enable the development of interoperable cards issued by any card industry sector. The standard will describe an application interface and behavior of the SSCD i.e. it should be possible to implement on e.g. native and interpreter based cards.

The standard is compliant with other European standards developed in the framework of the European Directive 1999/93.

This standard *Application Interface for smart cards used as Secure Signature Creation Devices* consists of two parts:

- Part 1: “Basic Services” describes the mandatory specifications for an SSCD as part of generic IAS Services to be used in compliance to requirements of Article 5.1 of the Electronic Signature Directive,
- Part 2: “Additional Services” describes remaining services for IAS.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

1 Scope

Part 1 of this series specifies the application interface to Smart Cards during the usage phase, used as Secure Signature Creation Devices (SSCD) according to the Terms of the European Directive on Electronic Signature 1999/93 to enable interoperability and usage as SSCD on a national or European level.

This document describes the mandatory services for the usage of Smart Cards as SSCDs based on CEN CWA 14890 (all parts). This covers the signing function, storage of certificates, the related user verification, establishment and use of trusted path and channel, requirements for key generation and the allocation and format of resources required for the execution of those functions and related cryptographic token information.

Thereby the functionality of CWA 14890-1 is enhanced in the following areas:

- Device authentication with Elliptic Curves (ELC) for existing asymmetric authentication protocols (RSA Transport, Privacy Protocol),
- Enhancement of existing asymmetric authentication protocols due to privacy and non-traceability constraints,
- **Card Verifiable (CV) Certificate Formats** (self descriptive) with ELC for all types of authentication and authorization protocols,
- Secure Messaging Tags and use of commands with Odd-INS Code in compliance to the actual ISO/IEC 7816-4,
- Further hash algorithms (SHA2-family) with corresponding Object identifier and Algorithm references,
- Use of AES in authentication protocols,
- Use of AES for secure messaging.

The following items are out of scope:

- 1) The physical, electrical and transport protocol characteristics of the card,
- 2) The external signature creation process and signature environment,
- 3) The elements required to verify an electronic signature produced by a card used as a SSCD,
- 4) The error handling process.

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.

NOTE For future implementations refer to the latest versions of the referenced documents; the following list refers to documents relevant at publication of this standard EN 14890.

EN ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes (ISO 3166-1:2006)*

ISO/IEC 7816-4:2005, *Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*

ISO/IEC 7816-6, *Identification cards – Integrated circuit cards – Part 6: Interindustry data elements for interchange*

ISO/IEC 7816-8:2004, *Identification cards – Integrated circuit cards – Part 8: Commands for security operations*

ISO/IEC 7816-11:2004, *Identification cards – Integrated circuit cards – Part 11: Personal verification through biometric methods*

ISO/IEC 7816-15:2004, *Identification cards – Integrated circuit cards with contacts – Part 15: Cryptographic information application*

ISO/IEC 9796-2:2008, *Information technology – Security techniques, Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms*

ISO 11568-2:2005, *Banking – Key management (retail) – Part 2: Symmetric ciphers, their key management and life cycle*

ISO/IEC 14888-2:2008, *Information technology – Security techniques – Digital signatures with appendix – Part 2: Integer factorization based mechanisms*

ISO/IEC 15946-1, *Information Technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General*

ISO/IEC 18033-3, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 7816-4:2005 and the following apply.

3.1

Answer-to-Reset file

elementary file which indicates operating characteristics of the card

3.2

command-response pair

set of two messages: a command followed by a response

3.3

data element

item of information seen at the interface for which are defined a name, a description of logical content, a format and a coding

3.4

data object

information seen at the interface which consists of a tag, a length and a value (i.e., a data element)

NOTE In this specification, data objects are referred to as BER-TLV data objects.

3.5

data unit

smallest set of bits which can be unambiguously referenced

3.6

dedicated file

file containing file control information and, optionally, memory available for allocation

NOTE It may be the parent of EFs and/or DFs.

3.7

DF name

string of bytes which uniquely identifies a dedicated file in the card

3.8

elementary file

set of data units or records which share the same file identifier

NOTE It cannot be the parent of another file.

3.9

file control parameters

logical, structural and security attributes of a file

3.10

file identifier

2-bytes binary value used to address a file

3.11

master file

mandatory unique dedicated file representing the root of the file structure

3.12

message

string of bytes transmitted by the interface device to the card or vice-versa, excluding transmission-oriented characters as defined in ISO/IEC 7816-3

3.13

parent file

dedicated file immediately preceding a given file within the hierarchy

3.14

password

data which may be required by the application to be presented to the card by its user

NOTE A password in the context of this specification is a string of numbers and/or ASCII characters.

3.15

path

concatenation of file identifiers without delimitation

NOTE If the path starts with the identifier of the master file, it is an absolute path.

3.16

private key

secret part of an asymmetric key pair e.g. signature creation data as specified in the EU directive for electronic signatures

3.17

public key

public part of an asymmetric key pair

3.18

record

string of bytes which can be handled as a whole by the card and referenced by a record number

3.19

record number

sequential number assigned to each record which uniquely identifies the record within its elementary file

3.20

retry counter

counter being used to count the number of erroneous usages of a related (security) object

NOTE If the object (e.g. password entry) was used correctly (correct password entered) then the retry counter is reset to its initial value. A typical value of a retry counter is 3.

3.21

secret key

either a symmetrical key or private part of a public key pair

3.22

trusted channel

means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP

3.23

trusted environment

operating environment that avoids the fraud with data that is seen at the communication interface by the definition of its physical location, physical security protection or physical access conditions or other measures to protect from tampering or intrusion of data

NOTE See 8.2.3.

3.24

trusted path

means by which a user and a TSF can communicate with necessary confidence to support the TSP

3.25

untrusted environment

operating environment that allows tampering and intrusion of data available at the communication interface

NOTE The decision whether an environment is trusted cannot always be made at the system level and therefore might have to be taken by the card holder.

3.26

usage counter

counter updated each time a related object is used

NOTE Opposite to retry counters, usage counters are not reset to their initial value after the object was successfully used.

4 Symbols and abbreviations

AES	Advanced Encryption Standard
AID	Application Identifier
AKI	Authority Key Identifier
AKP	Asymmetric key pair
APDU	Application protocol data unit
AT	Authentication Template, CRT for Authentication
AUT	Authentication
BCD	Binary Coded Decimal
BER	Basic Encoding Rules
C	Certificate
CA	Certification Authority
CAR	CA Reference
CC	Cryptographic Checksum
CG	Cryptogram
CH	Cardholder
CHA	Certificate Holder Authorization
CHR	Certificate Holder Reference
CIA	Cryptographic Information Application
CIO	Cryptographic Information Object
CLA	Class byte
CPI	Certificate Profile Identifier
CRT	Control Reference Template
CS	CertSign, certificate containing the public part of a signature key pair
CV	Card Verifiable
CWA	CEN Workshop Agreement
C/S	Client/Server
D[key](msg)	Decipherment of <msg> with <key>
DE	Data Element
DES	Data Encryption Standard
DF	Dedicated File
DH	Diffie-Hellman
DIR	Directory
DO	Data Object
DS	Digital Signature
DSA	Digital Signature Algorithm
DSI	Digital Signature Input
DST	Digital Signature Template, CRT for DS
E[key](msg)	Encipherment of <msg> with <key>
ECC	European Citizen Card

EDFB	Encrypted Data Formatted Block
EF	Elementary File
ELC	Elliptic Curve Cryptography
FCI	File Control Information
FID	File Identifier
h(msg)	hash-code of <msg>
HT	CRT for hash code
ICC	Integrated Circuit(s) Card
ID	Identifier
IFD	Interface Device
INS	Instruction byte
KE	Key Encipherment
KID	Key Identifier
MAC[key](msg)	Message Authentication Code of <msg> built with <key>
MF	Master File
MSE	MANAGE SECURITY ENVIRONMENT
OID	Object Identifier
PI	Padding Indicator
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PrK	Private Key
PRND	Padding Random Number
PSO	PERFORM SECURITY OPERATION
PSS	Probabilistic Signature Scheme
PuK	Public Key
P1-P2	Parameter bytes
RC	Retry Counter
RCA	RootCA
RD	Reference Data
RFU	Reserved for Future Use
RID	Registered application provider identifier
RND	Random number
RSA	Sig-Alg. of Rivest, Shamir, Adleman
SCA	Signature-Creation Application
SE	Security Environment
SHA	Secure Hash Algorithm
SIG()	Signature of
SK	Secret Key
SM	Secure Messaging

SN	Serial Number
SSC	Send Sequence Counter
SSCD	Secure Signature Creation Device
SW1-SW2	Status bytes
TDES	Triple-DES
TOE	Target of evaluation
TLV	Tag, Length, Value
TSF	TOE Security Functions
TSP	TOE Security Policy

5 Signature application

5.1 Application Flow

The following diagrams demonstrate a typical application flow as described in CWA 14170 [3].

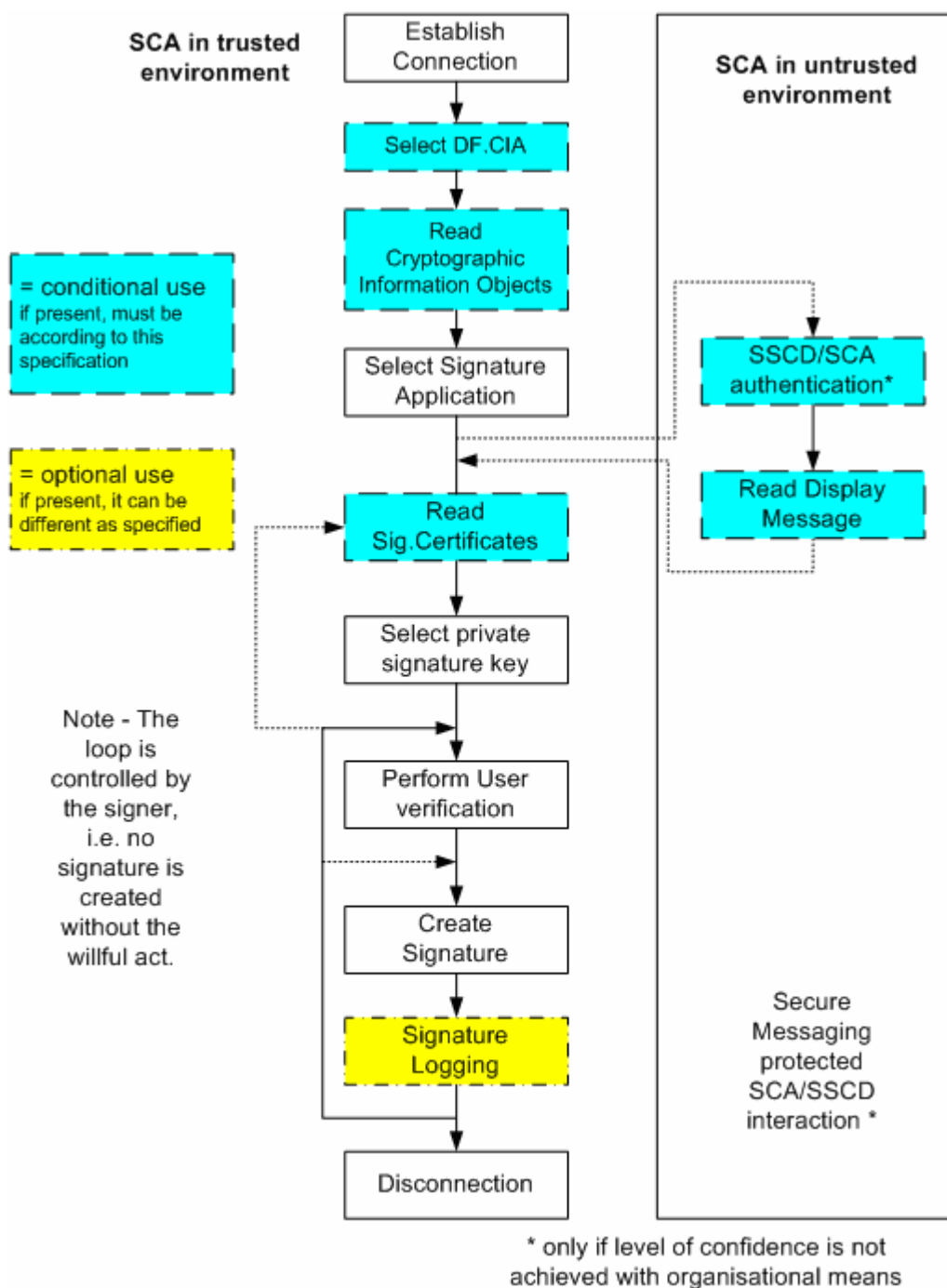


Figure 1 — Interaction sequences between SCA and SSCD in compliance with EU directive

NOTE The device authentication may alternatively be done before the selection of the Signature Application.

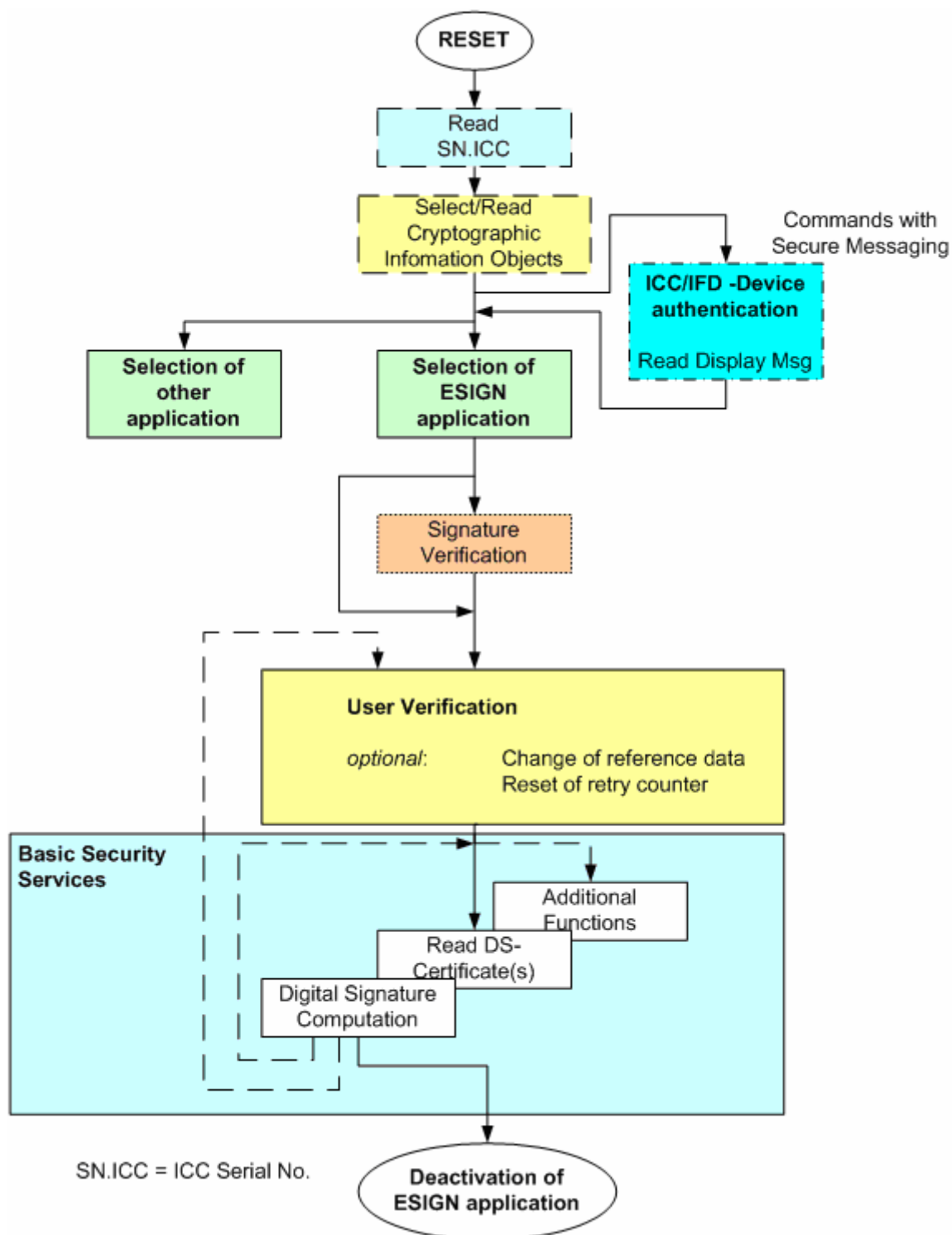


Figure 2 — Flowchart example for ICCs with ESIGN-Application as specified in this standard

5.2 Trusted environment versus untrusted environment

According to the definitions in “trusted environment” and “untrusted environment” this specification describes additional mechanisms in order to accomplish the required security as claimed in the protection profile [2]. The additional mechanisms according to this specification are:

- Device authentication;
- Secure messaging.

The following scenario displays the typical command flow, when an application is started.

5.3 Selection of ESIGN application

5.3.1 General

An ESIGN application is selected by its Application Identifier (AID) [32]. The RID is registered by the ISO registration authority and has the following value:

A0 00 00 01 67.

The entire AID has the following value:

AID = A0 00 00 01 67 || “ESIGN” = A0 00 00 01 67 45 53 49 47 4E

with

Category = ‘A.....’ (international)

PIX = ‘ESIGN’ = 5 bytes

Table 1 — SELECT ESIGN application — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	‘A4’ — SELECT
P1	‘04’ select by DF Name
P2	‘00’ or ‘02’ return FCI info (see remark below)
Lc field	Length of command data field
Data field	AID.ESIGN — application identifier
Le field	‘xx’ expected response length

For the description of the SELECT command refer to ISO/IEC 7816-4:2005, Clause 7.1.1.

Remarks

P2 It is left to the particular application to use other specified values from ISO/IEC 7816-4, this specification does not mandate the retrieval of the file control information. JAVA cards, however, require P2 = ‘00’ or P2 = ‘02’.

data field contains the AID.ESIGN as described above.

Table 2 — SELECT ESIGN application — response APDU

Response Parameter	Meaning
Data field	Data according to P2. An FCI shall at least contain the DO '84' (DF Name).
SW1-SW2	Refer to ISO/IEC 7816-4

5.3.2 Exceptions for Secure Messaging

In case of a card without a MF, the SELECT by AID of an application must always be issued by the terminal in clear, even if a secure messaging session is currently active. For such cards this SELECT by AID in clear shall not break the current secure messaging session (if any). For such cards this behavior is necessary in order to allow the association of one applet instance to one application in a Java Card. See 8.12.

5.4 Selection of cryptographic information application

In compliance with ISO/IEC 7816-15 the cryptographic information application (CIA) belonging to the ESIGN application has the following AID:

AID.CIA_ESIGN = 'E8 28 BD 08 0F A0 00 00 01 67 || "ESIGN" (15 bytes)

The selection of the DF.CIA(ESIGN) is done with a DF Name as shown above.

5.5 Concurrent usage of signature applications

5.5.1 General

Concurrent usage of signature applications can be realized using the mechanisms of logical channel as defined in ISO/IEC 7816-4.

5.5.2 Methods of channel selection

The mechanism for channel selection is optional.

A channel may be selected by:

- implicit selection with CLA byte of SELECT command;
- explicit selection with the MANAGE CHANNEL command.

Both methods are described in ISO/IEC 7816-4 and will not be discussed in the specification.

5.5.3 Security issues on multiple channels

In order to access the resources of the ICC from another channel the 'shareable' bit shall be set in the FCI of the application DF.

Each invocation of an ESIGN-Application has its own security status.

5.6 Security environment selection

A signature card may contain more than one authentication- or signature key and one or more user reference data respectively. In this case the IFD needs to select the appropriate keys to be involved in authentication or signature operations and/or user verification (e.g. PIN, Password or biometric data) respectively. The command **MANAGE SECURITY ENVIRONMENT (SET)** (refer to ISO/IEC 7816-4:2005, Clause 7.5.11) is used in order to specify the desired keys to be involved in the next operation.

A security environment selection by means of an **MSE: RESTORE** command, is applicable in the context of the following operations:

- User verification (PIN, Password or biometric);
- Signature creation as described in Clause 7;
- Device authentication as described in Clause 8;
- Key Generation as described in Clause 10;
- Verification of card verifiable certificates as described in Clause 14;
- Authentication services as described in Clause 6 in EN 14890-2:2008;
- Authentication services as described in Clause 7 in EN 14890-2:2008;
- Key decipherment as described in Clause 8 in EN 14890-2:2008;
- Signature verification as described in Clause 9 in EN 14890-2:2008.

The actual application of the **MANAGE SECURITY ENVIRONMENT** is discussed in the appropriate Clauses.

5.7 Key selection

The selection of keys is performed by the command **MSE:SET** if not specified otherwise (e.g. in the parameters of the command header). An appropriate key ID is submitted and stored in the ICC in order to specify the key to be used with the next command(s). Key selection has to be done prior to a security operation unless the key was previously selected.

A key ID is typically a number represented by one or more bytes describing a non-standardized value which is application- or operating system specific. The IFD may retrieve the link between a certificate and a corresponding KeyID from the DF.CIA application data as described in Clause 16.

5.8 Basic Security Services

A basic security service, as opposed to an extended security service, is a set of SSCD functions that serve the purpose of signature generation, verification, authentication or key en-/decipherment.

An extended security service comprises the installation of data and keys, card life cycle management and other functionalities that extend the scope described in the EU directive [1].

The two volumes of this document specify the following basic security services.

- **User verification** service, refer to Clause 6.
- **Key generation**, refer to Clause 10.
- **Device authentication** service including secure messaging, refer to Clause 8.

- **Digital signature service** for the advanced electronic signatures services of the cardholder, refer to Clause 7.
- **Authentication service** as required for specific authentication procedure, refer to Clause 6 in EN 14890-2:2008.
- **Encryption key decipherment service**, refer to Clause 8 in EN 14890-2:2008.
- **Signature verification service**, refer to Clause 9 in EN 14890-2:2008.

6 User verification

6.1 General

User verification is part of the application flow as described in Figure 2.

Three general methods of user verification are available:

- Knowledge based user verification (e.g. password);
- Biometric user verification.

Specific local verification data, dedicated to protect the use of the private key for signature generation shall be applied. If several private keys for signature generation are in the card, their use may optionally be protected by different verification data.

For protection of the client/server authentication function and the key decipherment function separate (global or local according to the security policy) verification data shall be used.

In either of these cases the verification data might be subject to alteration.

The following commands, as defined in ISO/IEC 7816-4 are used to manage the verification process:

- VERIFY;
- CHANGE REFERENCE DATA;
- RESET RETRY COUNTER.

Regardless of the usage of these commands at any time, it shall be ensured that the usage of the signature key is protected by preceding user verification

If knowledge based or biometric user verification is part of a Common Criteria evaluation a Strength of Function (SOF) analysis has to be performed. The assessment of a knowledge based mechanism is a relatively simple and well established probabilistic analysis. For biometric authentication mechanisms the SOF analysis is more complex as e.g. the proof that a claimed False Acceptance Rate (FAR) is met might need comprehensive data bases of fingerprints.

6.2 Knowledge based user verification

6.2.1 General

Knowledge based user verification requires a password to be entered by the user. The actual verification is done with the VERIFY command.

6.2.2 Explicit user verification

European applications frequently restrict passwords to numbers only. As such they are also known as 'PIN'. Find further information under 6.2.3.

Execution Flow 1 of 1

Table 3 — VERIFY command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'20' VERIFY
P1	'00'
P2	<keyref> password reference
Lc field	Length of command data field
Data field	<verification data>

The structure and coding of the VERIFY command refer to ISO/IEC 7816-4:2005, Clause 7.5.6..

Table 4 — VERIFY — response APDU

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4.

6.2.3 Password related mechanisms

Password reference value (described as key identifier KID)

The parameter P2 codes the reference value of the password.

This standard specifies the following KIDs:

KID = '8x' Key reference to local reference data for protection of the use of a private signature key (min. 6 digits or characters); the value of the KID and the purpose of the signature password shall be indicated in the cryptographic information objects.

Also other passwords may be of the type "local".

KID = '0x' Key reference to global reference data; the value of the KID and the purpose of the password shall be indicated in the cryptographic information objects.

NOTE In order to allow the co-existence of EMV-applications and ESIGN-applications on the same ICC, only the three least significant bits should be used to differentiate keys. Refer to "EMV2000, Integrated Circuit Card Specification for Payment Systems Book 3, "Application Specification Version 4.1", May 2004" [11].

6.2.4 Presentation formats

The data field codes the verification data in one of the following formats.

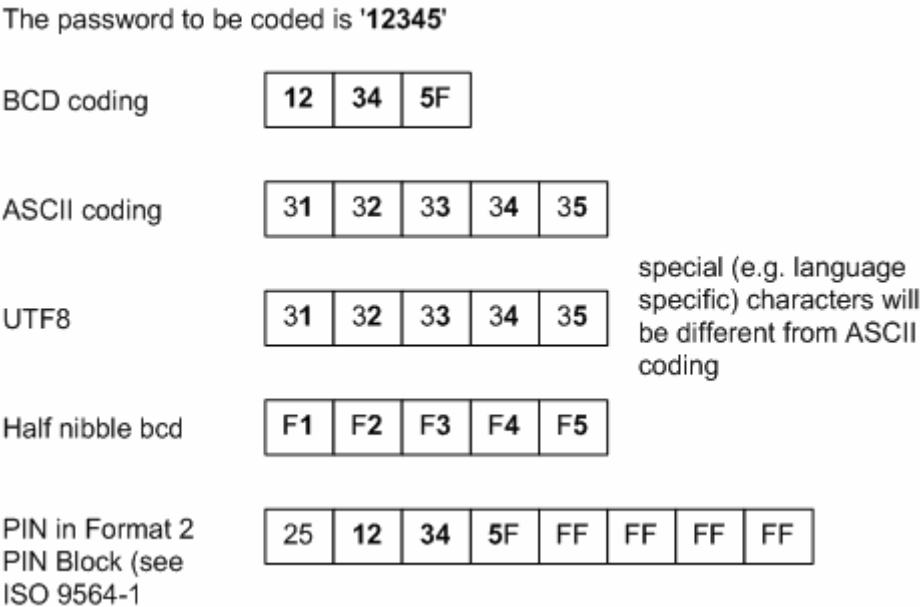


Figure 3 — Example of coding '12345' in different presentation formats

Figure 3 demonstrates the coding when the different presentation formats are used. The actual presentation format may be retrieved from information presented in DF.CIA (refer to Clause 16), unless implicitly known.

Passwords protecting the usage of signature keys shall have a minimum of 6 digits or characters. All other passwords shall have a minimum of 5 digits or characters.

NOTE The specifications are in compliance with appropriate protection profiles e.g. CEN CWA 14169 [2].

6.2.5 Retry counters

In order to avoid exhaustive search attacks, the password shall be protected by an appropriate retry counter.

In order to avoid an exhaustive search attack on the password under a known resetting code, the resetting code should be protected by an appropriate usage counter with a usage counter limit with a low value (typically U = 3).

After successful presentation, the appropriate retry counter is automatically reset to its initial value (typically N = 3).

For the definition of the counter types refer to Clause 3.

6.2.6 Password Change

The CHANGE REFERENCE DATA command for changing the global password may be used at any time convenient for the cardholder.

The change of DF-specific (local) password can only be performed after selection of the appropriate DF.

Execution Flow 1 of 1

Table 5 — Password Change — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'24' CHANGE REFERENCE DATA
P1	'00' no algorithm specified
P2	['0x' '8x'] KID of referenced password
Lc field	Length of command data field
Data field	<password> <new password>

For the description of the command refer to ISO/IEC 7816-4:2005, Clause 7.5.6.

The length of the existing Reference Data is known in the card, so that neither a delimiter nor padding for filling up fixed formats is necessary. The length of the new password therefore computes $L_{\text{new}} = L_c - L_{\text{old}}$.

Table 6 — Password Change — response APDU

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4.

6.2.7 Reset of RC and setting a new password

After N (N as specified by application) subsequent false presentations of the password, the password is locked and does not allow further usage of the protected functions.

With the ISO/IEC 7816-4 command RESET RETRY COUNTER, the cardholder can initiate the reset of the RC to its initial value N. The Resetting Code shall have a minimum length of 6 digits¹⁾.

It is also possible to define a new password with the RESET RC command.

The reset of DF-specific (local) retry counter can only be performed after selection of the appropriate DF.

¹⁾ 'Digit' can also be a character as part of a password.

Table 7 — RESET RETRY COUNTER — command APDU

Command Parameter	Meaning
CLA INS	according to ISO/IEC 7816-4 '2C' RESET RETRY COUNTER
P1 P2	['00' .. '03'] ['0x' '8x'] KID of referenced password
Lc field	Length of command data field
Data field	see below

For the coding of the RESET RETRY COUNTER command refer to ISO/IEC 7816-4:2005, Clause 7.5.10.

The content of the data field depends on the value of P1. The RESET RETRY COUNTER shall not change the security status, i.e. for using the signature creation key a VERIFY command is still required.

P1 = '00' Resetting code followed without delimitation by new reference data

P1 = '01' Resetting code

P1 = '02' New reference data

P1 = '03' Data field absent

Table 8 — Change Password — response APDU

Response Parameter	Meaning
Data field	absent
SW1-SW2	See ISO/IEC 7816-4.

After successful presentation of the Resetting Code, the retry counter of the referenced password is automatically reset to its initial value (typically N = 3).

6.3 Biometric user verification

6.3.1 General

The support of a biometric user verification (on-card matching) is optional.

Two general methods of biometric user verification are available:

- The sensor is off-card, i.e., the biometric verification data is sent from the interface device to the card.
- The sensor is on-card, so there is no need to transmit the biometric verification data from the interface device to the card.

If biometric user verification is supported and the sensor is off-card, the Finger Minutiae Matcher compliant to ISO/IEC 19794-2 shall be used. The compact size finger minutiae format identified by the CBEFF format type '0005' shall be applied as encoding format for finger minutiae.

The ICC shall provide the IFD with information about the supported biometric verification service using a Biometric Information Template (BIT) compliant to ISO/IEC 7816-11 that shall be encoded as presented in Table 14. The BIT shall not contain a data object for minutiae order referenced by tag '82', i.e., the card shall not rely on an ordering procedure performed by the IFD.

6.3.2 Retrieval of the Biometric Information Template

Prior to the verification, the SCA may need information about:

- AlgID of the matching algorithm and key reference of the biometric reference data;
- biometric type (e.g. fingerprint);
- biometric type instance (e.g. left pointer finger enrolled);
- format owner of the biometric data;
- format type of the biometric data;
- on-/off-card sensor.

This information may be found in the DF.CIA description (see Clause 16).

Alternatively this information may be provided in the Biometric Information Template (BIT), refer to ISO/IEC 7816-11. The BIT may be present in the DF.CIA or be retrieved with the GET DATA command.

The enrolment process is out of the scope of the present specification.

Table 9 — Retrieve Biometric Information Template(s) — command APDU (optional)

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'CA' GET DATA
P1-P2	'7F60' Example: Biometric Information Template
Le field	'00'

For the coding of the command GET DATA refer to ISO/IEC 7816-4:2005, Clause 7.4.2.

Table 10 — Retrieve Biometric Information Template(s) — response APDU (optional)

Response Parameter	Meaning
Data field	BIT (Value field of Biometric Information Template(s))
SW1-SW2	Refer to ISO/IEC 7816-4.

For the format of the biometric information template refer to Clause 6.3.3.2.

6.3.3 Performing the biometric user verification

6.3.3.1 General

Two situations have to be distinguished:

- Sensor **off**-card;
- Sensor **on**-card.

Secure messaging (see Clause 9.6) is required when executing commands on biometric user verification with Sensor off-card.

6.3.3.2 Sensor off-card

The sensor is outside the ICC, i.e. the captured biometric data needs to be sent to the ICC for on-card matching.

The IFD retrieves the Biometric Information Template by means of a GET DATA command according to 6.3.2. The corresponding path attribute in the Cryptographic Information Application denotes the DF that shall be selected prior to this GET DATA command.

Then the IFD uses a VERIFY command to process the biometric data structured according to the biometric template.

Table 11 — Biometric user verification — command APDU (Biometric verification)

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'21' VERIFY
P1	'00'
P2	'xx' reference data qualifier
Lc field	Length of command data field
Data field	<biometric data template> refer to Table 13

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.6.

Table 12 — Biometric user verification 2 of 2- response APDU

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4

6.3.3.3 Biometric Templates

Table 13 — Biometric data template

Tag	Length	Value		
'7F 2E'	Var.	Biometric Data Template		
		Tag	Length	
		'81'	Var.	Biometric data with standardized format (primitive encoding)

Table 14 — Biometric Information Template

Tag	Length	Value						Presence
'7F 60'	Var.	Biometric Information Template (BIT)						
		Tag	Length	Value				
		'83'	'01'	Reference data qualifier				Mandatory
		'A1'	Var.	Biometric Header Template (BHT) in compliance with CBEFF				Mandatory
				Tag	Length	Value		
				'81'	'01'	'08' (fingerprint)		Mandatory
				'82'	'01'	Biometric subtype (refer to ISO/IEC 7816-11:2004, Table C.3)		Mandatory
				'87'	'02'	'0101' (ISO/IEC JTC 1)		Mandatory
				'88'	'02'	'00 05'		Mandatory
				'B1'	Var.	Biometric matching algorithm parameters		Optional
						Tag	Length	
						'81'	'02'	Minimum number of minutiae required (1 byte, binary coding) Maximum number of minutiae accepted (1 byte, binary coding)

6.3.3.4 Sensor on-card

For the on-card verification, the VERIFY command selects the biometric data reference through P2.

Execution Flow

Table 15 — Biometric verification — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'21' VERIFY
P1	'00' no algorithm specified
P2	['0x' '8'x] KID of referenced password or biometric data reference
Lc field	Length of command data field
Data field	'5F2E' '00' '4D' '00' empty verification DO and extended. header list DO

For the description of the command refer to ISO/IEC 7816-4:2005, Clause 7.5.6.

The length of the existing Reference Data is known in the card, so that neither a delimiter nor padding for filling up fixed formats is necessary.

Table 16 — Biometric verification — response APDU

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.6.

6.3.4 Reset of RC

For resetting the RC refer to 6.2.6.

With respect to biometric reference data only the RESET RETRY COUNTER command variants that make use of P1 = '01' and P1 = '03' are specified in this document. The usage of the command variants with P1 = '00' and P1 = '02' is outside the scope of the document as the enrolment process is not defined in this document.

7 Digital Signature Service

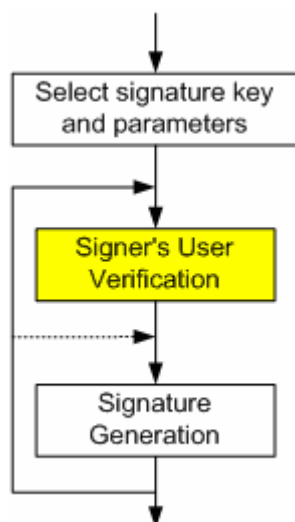
This Clause covers “qualified electronic signatures” according to the EU Directive 1999/93/EC Section 5.1 [1].

7.1 Signature generation algorithms

For signature generation algorithms as referenced in the current European and national algorithm catalogs shall be applied. Examples of existing papers are given in [4], [5], [6], [7] and [8].

7.2 Activation of digital signature service

In order to enable the digital signature service user verification shall be applied prior to signature generation.



NOTE The loop is controlled by the signer, i.e. no signature is created without a willfull act.

Figure 4 — Signer's User Verification

Refer to Figure 1 for the overview of the signature generation flow.

A knowledge based interaction always expresses the explicit will of the card holder to sign a document whereas a biometric identification does not necessarily imply the intention to sign. Therefore knowledge based user verification is a mandatory feature, unless a particular specification in a corresponding protection profile supersedes the claim of this standard.

For the description of the user verification refer to Clause 6.

7.3 General aspects

A digital signature process consists of several steps. Some of them are done outside the ICC, some of them are done inside the ICC, and some could be done on either side.

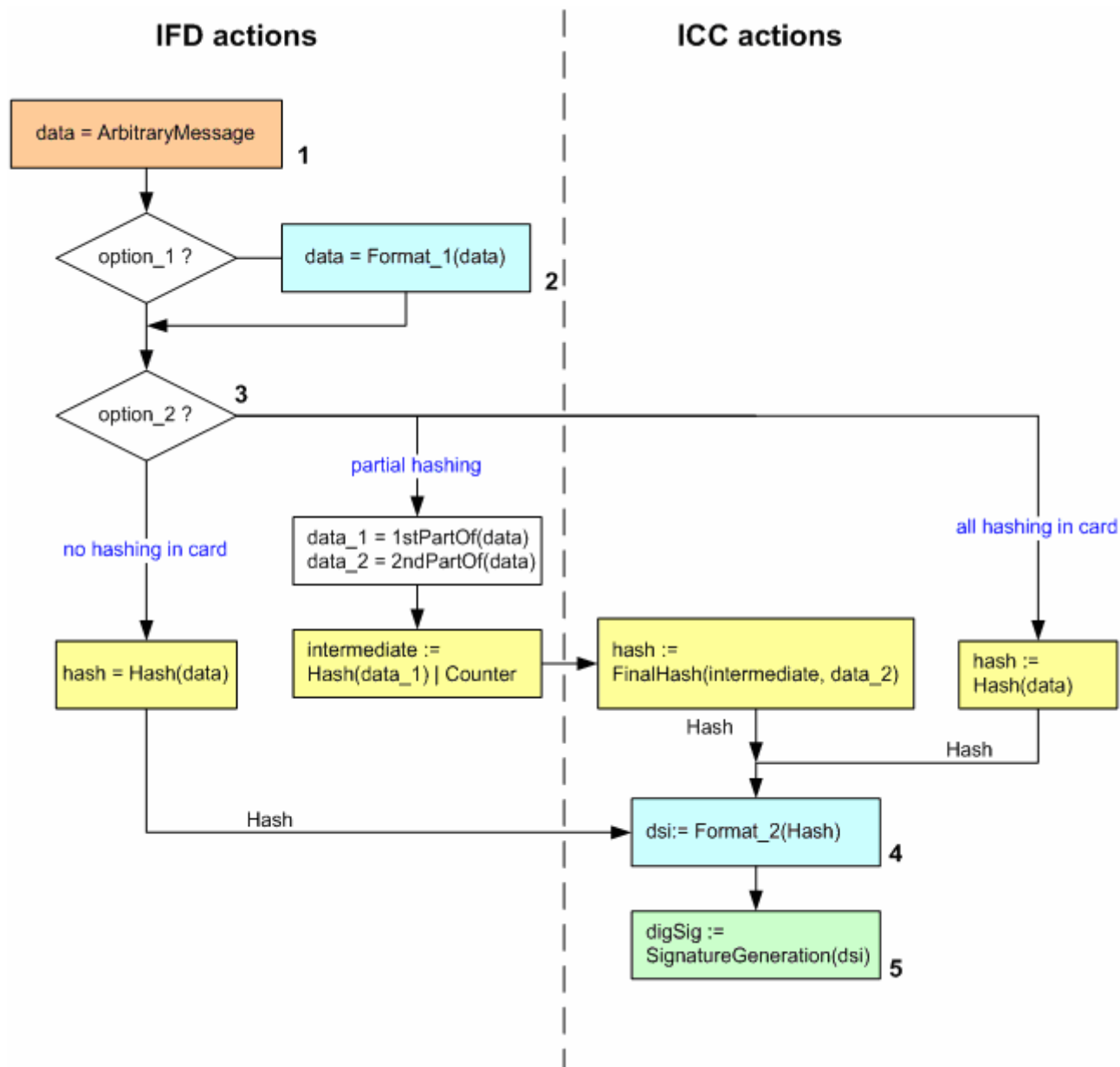


Figure 5 — Data-Flow chart for signature generation process (RSA Example)

Figure 5 shows an example of the alternatives to process a hash. The numbers 1-5 refer to the steps described below. The intermediate hash-code is the digest of preceding hash round followed by the big endian bit count. Only fully hashed blocks are transmitted, therefore the length of `data_1` (on partial hashing) is always a multiple of the digest size, related to the applied algorithm.

In principle, the signature generation process works as follows:

- 1) The signature generation process starts with an arbitrary message.
- 2) Optionally the original message is formatted in an application specific way. Typically signature attributes and other format information besides the actual document to be signed are added to the

arbitrary message. The process of pre-formatting the data to be hashed is expressed as **Format_1** mechanism. The specification of the optional format mechanism **Format_1** is outside the scope of this document.

- 3) The data has to be hashed by applying a hash function which is on the list of the applicable hash functions. The hash-code could be considered as a compressed version of the text. Depending on *option_2* the hashing will be done either:
 - completely outside the ICC,
 - partly outside and partly inside the ICC, thus data is split into two parts,
 - completely inside the ICC depending on the card functionality and the software organization.
- 4) Depending on the signature algorithm the *hash-code* has to be formatted by a format mechanism **Format_2**. Typically this is a *padding mechanism* as described in 13.3. The result of this formatting is a *Digital Signature Input (dsi)*. Several format mechanisms have been defined; some of them are evaluated as “weak” and should not be used. The algorithm catalogues cover these aspects and propose formatting schemes which are approved until a specified date.
- 5) The final *dsi* (Digital Signature Input) is the string to be transformed by the signature algorithm. The result is the Digital Signature DS[<key>](*dsi*).

7.4 Signature Generation

Depending on which data path is used in Figure 5, from the ICC’s point of view the following cases could be distinguished.

7.4.1 No hashing in Card

The data to be signed is completely hashed outside the ICC. Depending on the algorithm the *hash-code* is transferred to and signed by the ICC by the command PSO:COMPUTE DIGITAL SIGNATURE.

Execution Flow 1 of 1

Table 17 — Signature computation (no hashing in card) — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	‘2A’ PERFORM SECURITY OPERATION
P1	‘9E’ COMPUTE DIGITAL SIGNATURE
P2	‘9A’ ‘data to be signed’
Lc field	Length of command data field
Data field	data to be signed
Le field	‘00’

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

The response data field contains the *plain signature* without further TLV coding.

Table 18 — Signature computation (no hashing in card) — response APDU

Response Parameter	Meaning
Data field	plain signature without further TLV coding
SW1-SW2	Refer to ISO/IEC 7816-4.

7.4.2 Partial hashing

The first part of the data to be signed is hashed outside the ICC. The intermediate hash-code is transferred to the ICC by the command PSO:HASH together with the remaining part of the data. Afterwards the signature is generated by the command PSO:COMPUTE DIGITAL SIGNATURE.

Execution Flow 1 of 2

Table 19 — Signature computation 1 of 2 (transfer intermediate hash-code) — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'90' hash
P2	'A0' input template for hash computation
Lc field	Length of command data field
Data field	'90' L ₉₀ <intermediate hash-code followed by big endian bit counter> '80' L ₈₀ <data to be hashed>

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

For the content of the command data field or DO 'A0' refer to 13.2.

The response data field is absent.

Table 20 — PSO-Hash response

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4

Execution Flow 2 of 2

The second command performs a digital signature on the hashed value.

Table 21 — Signature computation 2 of 2 (compute signature) — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'9E' COMPUTE DIGITAL SIGNATURE
P2	'9A' data to be signed
Le field	'xx'

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

The response data field contains the plain signature without further TLV coding.

Table 22 — PSO-COMPUTE DIGITAL SIGNATURE response

Response Parameter	Meaning
Data field	plain signature
SW1-SW2	Refer to ISO/IEC 7816-4.

7.4.3 All hashing in Card

The data to be signed is hashed inside the ICC. This is done by the command PSO:HASH. Finally the signature is generated by the command PSO:COMPUTE DIGITAL SIGNATURE.

Execution Flow 1 of 2

Table 23 — Signature computation 1 of 2 (all hashing in ICC) — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PSO:HASH
P1	'90' hash operation
P2	'80' plain value
Lc field	'xx'
Data field	plain data to be hashed

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

The response data field is absent.

Table 24 — Signature computation 1 of 2 (all hashing in ICC) — response APDU

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4.

Execution Flow 2 of 2

The second step computes the digital signature.

Table 25 — Signature computation 2 of 2 (compute signature) — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'9E' COMPUTE DIGITAL SIGNATURE
P2	'9A' data to be signed
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

The response data field contains the plain signature without further TLV coding.

Table 26 — PSO-COMPUTE DIGITAL SIGNATURE response

Response Parameter	Meaning
Data field	plain signature
SW1-SW2	Refer to ISO/IEC 7816-4.
^a In the case of ECDSA the digital signature is calculated according to ISO/IEC 15946-1; for the format refer to Clause 5.3.	

7.5 Selection of different keys, algorithms and input formats

If the card supports more than one key and an implicit selection is not available, then the private signature key needs to be selected.

If the card supports several digital signature algorithms or different Digital Signature Input formats (refer to Clause 13), then the algorithm or the DSI format respectively has to be selectable. However, it is possible, that e.g. an algorithm is implicitly specified if a key is selected.

It is further possible that an algorithm, a key or a signature input format is not implicitly known, nor uniquely specified.

In all of the above cases the appropriate parameter has to be selected explicitly with a MANAGE SECURITY ENVIRONMENT command as described in ISO/IEC 7816-4:2005, Clause 7.5.11.

This can be achieved by:

- restore an existing SE (refer to 7.5.1);
- modify a KeyID/[AlgID] in a template (refer to 7.5.3).

7.5.1 Restore an existing SE

Execution Flow 1 of 1

Table 27 — MANAGE SECURITY ENVIRONMENT — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'F3' RESTORE
P2	SE# security environment number

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.11.

The response data field is absent.

Table 28 — MANAGE SECURITY ENVIRONMENT response

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4.

NOTE The actual SE# is described in Part 2 of this specification.

7.5.2 Modify the HT of a current SE

If a hash algorithm is not implicitly known by the ICC, it has to be selected with the MANAGE SECURITY ENVIRONMENT command.

Execution Flow 1 of 1

Table 29 — Set hash template — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'41' SET (hash template acc. to p2)
P2	'AA' CRT for HT (hash template)
Lc field	'xx' length of the following data
Data field	'80' L ₈₀ yy AlgId

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.11.

The response data field is absent.

Table 30 — Set hash template — response

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4.

7.5.3 Modify the DST of a current SE

If a signing key is not implicitly known by the ICC, it has to be selected with the MANAGE SECURITY ENVIRONMENT command. Optionally an algorithm identifier can be supplied in order to specify variations of the algorithm (e.g. for padding formats).

Execution Flow 1 of 1

Table 31 — Selection of signature key — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'41' SET for signature creation
P2	'B6' CRT for DST
Lc field	'xx' length of the following data
Data field	'84' L ₈₄ xx ['80' L ₈₀ yy] Key Id [AlgId]

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.11.

For the algorithm ID refer to Table 143.

The response data field is absent.

Table 32 — Selection of signature key — response

Response Parameter	Meaning
Data field	absent
SW1-SW2	Refer to ISO/IEC 7816-4.

7.6 Read certificates and certificate related information

If more than one signature key is present in the card, the user may need to select the signature key for a certain signature creation. In the user certificates there are fields indicating whether it is a private or professional certificate. Therefore the user may want to evaluate these fields in order to select the proper certificate and with it the associated private key. In the CIOs there is a link between a certificate and the associated private key and an indication where the certificates are to be found (in the card or on a server), in the latter case the card will provide an URL and optionally a certificate serial number.

7.6.1 Read certificate related CIOs

CIOs related to certificates denote which certificates are present in the ICC or are to be found on a server and which private key corresponds to the public key present in the certificate (refer to Clause 16).

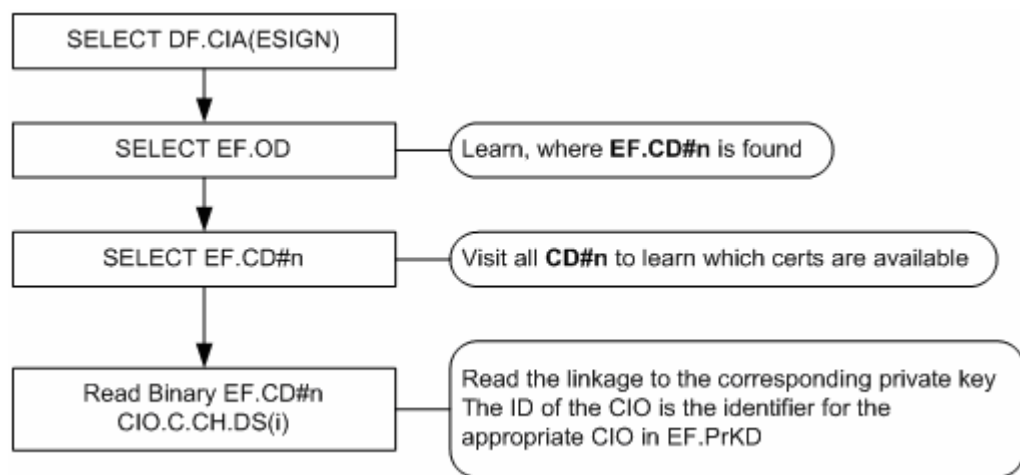


Figure 6 — Reading certificate related information (certificate is present in the ICC)

The certificate related information contains the ID of the appropriate CIO in EF.PrKD which contains the KeyId of the private key in question.

7.6.2 Read signer’s certificate from ICC

If present, the corresponding certificate file(s) may be read from the ICC. Prior to reading the appropriate file needs to be selected. The location (e.g. path or (short) FileID) is denoted in the corresponding CIO. in EF.CD#n.

Certificate related information is stored in EF.C.CH.DS(i). After selection of EF.C.CH.DS(i) according to the path information found in the CIO.C.CH.DS(i), either the certificate or related information may be read from the EF.C.CH.DS(i).

The actual content to be expected in EF.C.CH.DS(i) is described in DF.CIA. If no other description is available, then EF.C.CH.DS(i) contains the certificate.

Execution Flow 1 of 1

Table 33 — Read signer’s certificate — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	‘B0’ READ BINARY
P1-P2	‘xxxx’ Offset
Le field	‘00’ read all data

The response data field codes the data content of EF.C.CH.DS(i).

Table 34 — Read signer’s certificate — response APDU

Response Parameter	Meaning
Data field	data.EF.C.CH.DS(i)
SW1-SW2	Refer to ISO/IEC 7816-4.

The command and response APDUs are described in ISO/IEC 7816-4:2005, Clause 7.2.3.

The structure and coding of EF.C.CH.DS(i) will typically be an X.509 certificate format, but it could also be some other format. The actual format is application specific and out of the scope of this standard.

NOTE If the expected length exceeds the length of the APDU response data field then the READ BINARY might have to be called several times with an appropriate offset.

7.6.3 Retrieval of the signer’s certificate from a directory service

In order to retrieve the signer’s certificate through a directory service the card holder has to identify the certificate uniquely. This reference value information may be retrieved by reading the appropriate CIO.C.CH.DS from the DF.CIA(ESIGN) according to Figure 6.

According to ISO/IEC 7816-15:2004 “8.7.2 X.509 certificate attributes” the appropriate data object CIO.C.CH.DS contains the following fields.

```
X509CertificateAttributes ::= SEQUENCE
{
  value ObjectValue { Certificate },
  subject Name OPTIONAL,
  issuer [0] Name OPTIONAL,
  serialNumber CertificateSerialNumber OPTIONAL,
  ... -- For future extensions
}
```

The interpretation of the components shall be as follows:

X509CertificateAttributes.value: The value shall be a *Referenced Value* either identifying a file containing a DER encoded certificate at the given location, or a URL pointing to some location where the certificate can be found. [27]

8 Device authentication

This clause is related to the assumption/model that a device authentication has (to be/been) performed. Any statements made in this paragraph are conditional to that assumption. A specification for the need of device authentication is given in 8.2.

Device authentication requires mandatory steps in order to provide a secure authentication. A device authentication is mutual and combines two mechanisms:

- an ICC verifies the external world and is as well verified by the external world;
- the devices negotiate or exchange information to establish common session keys for subsequent operations.

After the negotiation of the symmetric keys, a secure session is established. A secure session is a cryptographic protection of the messages from either side. The cryptographic protection can be a cryptographic checksum on a plain text or an encrypted message text. Refer to Clause 9.

For performance reasons, the keys used for this secure messaging are symmetric keys. Therefore this document describes the establishment of symmetric session keys only, and does not consider an option for asymmetric session keys.

Once the session keys are established, a trusted channel is available to protect or conceal the information transmitted over the interface. The application of Secure Messaging (SM) is **mandatory** for **every subsequent operation** to ensure the provision of an entirely trusted channel. For exceptions refer to Clause 5.3.2. The conditions to end a secure messaging session are given in 8.12.

This document describes the following device authentication mechanisms

- asymmetric session key transport mechanism based on RSA;
- asymmetric session key agreement mechanism with privacy protection;
- asymmetric session key agreement mechanism with non-traceability and privacy based on ELC;
- symmetric authentication mechanism

in order to perform a mutual authentication protocol between IFD and ICC. The presentation of the asymmetric schemes is staged in order to trade off the security features with the required complexity of the authentication protocols.

NOTE 1 Authentication in general requires a defined order of steps to be processed. A violation of this order may be subject to abortion by the ICC. The mechanism to control the proper order of execution steps out of the scope of this standard.

NOTE 2 The IFD usually represents any external interface not being particularly specified.

8.1 Certification authorities and certificates

8.1.1 Certificate chains

Asymmetric authentication requires the presentation of public keys in certificates. A chain of certificates might have to be presented. To establish the 'starting point of trust' the first certificate sent to the ICC shall always be verifiable with a public key already existing in the card. The most general of such keys available in the ICC

can be a Root CA public key. Each certificate introduces another public key, while the presented certificate is always signed with a key, known by the ICC.

A certificate, signed by a descendant of Root A that conveys a public key that is a descendant of Root B is called a cross-certificate (refer to 8.1.2). The use of cross certificates allows the use of the ICC in different application environments. Cross certificates are also required for certificate expiration issues.

If a Root certificate is available, then Root CA provides certificates for the subordinate CAs or cross certificates are used in order to convey the public key(s).

Subordinate CAs issue certificates for users with the appropriate ESIGN application.

In order to use an ESIGN application a mutual authentication between card and customer terminal is required by the security policy. As PK algorithms are to be employed ICC and IFD authentication certificates are required.

The card is the carrier of the ICC authentication certificate; carrier of the IFD authentication certificate could be a plug-in card as an example for the realization of an authentication module inside the SCA (refer to Figure 7).

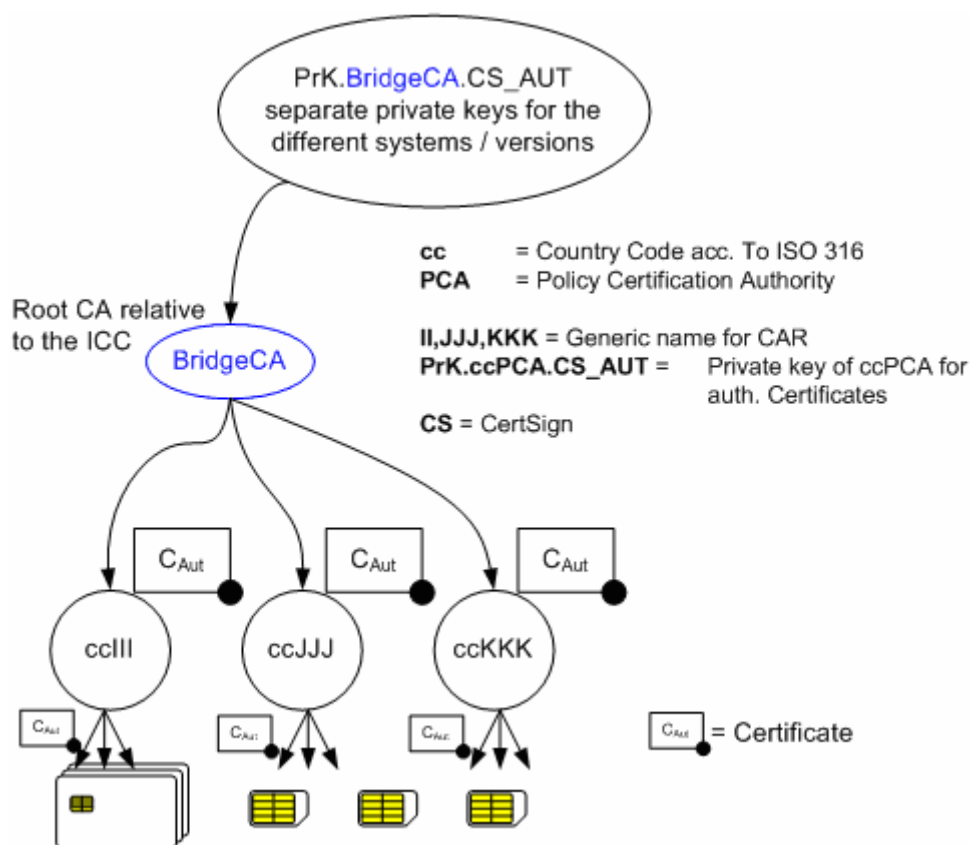


Figure 7 — Example for certificate authorities and certificates for ICC and IFD authentication

The above example repeats for every RCA key that is present in the card.

The handling of the DOs ‘CED — Certificate Effective Date’ and ‘CXD — Certificate Expiration date’ are out of the scope of this standard, refer to [21] for further details.

8.1.2 Usage of cross certificates

A typical use of cross certificates is the link of an existing certificate chain to an existing release of a public key.

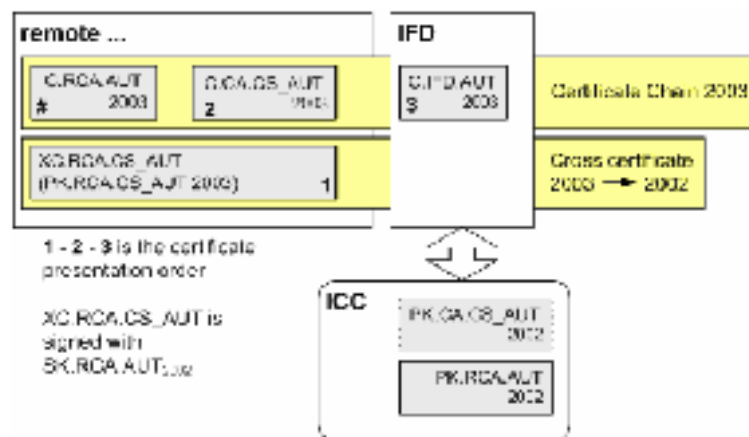


Figure 8 — Usage of cross certificates for different key versions

The example above shows PK.CA.CS_AUT(2002) available at the ICC. This key was brought to the card in 2002, however in 2003 the root key was changed and another key chain

PK.RCA.AUT(2003) → C.CA.CS_AUT(2003) → C.IFD.AUT(2003)

was used when generating the IFD's certificate C.IFD.AUT(2003).

In order to allow authentication of the IFD, in the above example the ICC might have to receive a cross certificate XC.RCA.CS_AUT(2003) first. This certificate is signed with the PrK.RCA.AUT(2002), however it transports the PuK.RCA.AUT(2003).

After having verified the cross certificate XC.RCA.CS_AUT(2003) the ICC may now receive the certificates of the 2003-chain up to the C.IFD.AUT which carries the final public key required for authentication.

The ICC does not recognize whether an incoming certificate is a cross-certificate or part of a certificate chain. If it can verify the incoming certificate with an existing public key, the ICC accepts the transported public key and may use it for the next instance of the chain. The 'start of trust' is always established with the first certificate, being verified with a public key that is available in the ICC.

When verifying a certificate, the verifier has to verify the correct role of the signer. Only those roles being allowed to sign certificates may be accepted for the presentation of a certificate chain. The role of a certificate is specified in Clause 14.

8.2 Authentication environments

In [3] two environments are distinguished with respect to signature creation applications.

8.2.1 SCA in trusted environment

The environment is considered to be trusted by the user. Device authentication is not required as the card holder knows the trusted environment that he will apply the card to.



Figure 9 — Trust of the environment

In this environment, device authentication as described in the rest of this Clause is optional.

8.2.2 SCA in untrusted environment

A device authentication shall be used if the operating environment of the card cannot be entirely trusted. This can be the case in public signature terminals or other devices that cannot provide a trusted channel.



Figure 10 — Communication in untrusted environment

Device authentication is mutual. The ICC shall authenticate the IFD and vice versa. The order of authentication, however, may differ, depending on the implemented scheme.

After successful device authentication, session keys are available on both sides to be used in subsequent transmissions. The appropriate secure messaging is in compliance with ISO/IEC 7816-4 and described in Clause 9.

Examples for an untrusted environment are:

- SCA and SSCD are not at the same location, i.e. the card is remote;
- usage of biometrics if the sensor is off-card;
- usage of contactless cards.

8.2.3 Specification of the environment

Whether an environment is trusted or not cannot be evaluated by the IFD a priori. Therefore it must be left to the card holder to decide, whether he may trust the application environment or not. The initiation of a device authentication in the application environment is out of the scope of this specification.

8.2.4 Display message mechanism

Given, the card holder has initiated a device authentication this specification proposes a 'display message' mechanism in order to verify that the device authentication has been performed successfully. If used, after successful device authentication, a display message shall indicate the successful device authentication in order to inform the user about the existence of a trusted channel. The application of the display message is described in 8.13.

8.2.5 Additional authentication environments

Device authentication may also be applied or necessary for multiple entity personalization (not in the scope of this standard).

8.3 Key transport and key agreement mechanisms

Key transport is the process of transferring a secret key, chosen by one entity (or a trusted center), to another entity, suitably protected by asymmetric techniques. Key transport requires encryption of the key part; this is typically done with the public key of the counterpart. If used, for the establishment of the secure messaging, key transport is used in both directions in order to derive a common key in the IFD and ICC.

Key agreement is the process of establishing a shared secret key between two entities A and B in such a way that neither of them can predetermine the value of the shared secret key. Key agreement mechanisms may be integrated in the device authentication.

In the context of session key establishment, ‘implicit key authentication’ means that after the execution of the mechanism only an identified entity can be in possession of the correct shared secret key.

This specification describes the following protocols:

- 8.4 “Key transport protocol based on RSA”;
- 8.5 “Device authentication with privacy protection”;
- 8.6 “Privacy constrained device authentication with non traceability based on ELC”;
- 8.8 “Symmetric authentication scheme”.

8.4 Key transport protocol based on RSA

The key transport protocol provides the authentication steps for a two-factor authentication. The protocol is already used and referenced in standards prior to the release of EN 14890-1 and is recommended for implementations which required compatibility with external world systems using this key transport protocol.

Three main phases may be distinguished:

- 1) Exchange public keys between ICC and IFD.
- 2) Internal authentication of the ICC.
- 3) External authentication of the IFD.

Internal authentication is processed first. Therefore this protocol is favorable for situations where the ICC should be authenticated prior to the IFD.

The following table shows device authentication with key transport. The information in the fields is simplified. A detailed description of the coding is specified below.

Table 35 — Reference device authentication with Key Transport

Stage	Step	IFD	Trans- mission	ICC
1	A	If PuK.CA_{IFD}.CS_AUT is available in the ICC continue with step D		
	B	Select and/or verify PuK.RCA.AUT.	→ ←	Select and/or verify key. Response OK
	C	PSO:Verify Certificate C_CV.CA.CS_AUT	→ ←	Verify certificate. Store PuK.CA _{IFD} .CS_AUT Response OK
	D	Select and/or verify PuK.CA _{IFD} .CS_AUT.	→ ←	Select and/or verify key. Response OK
	E	PSO:Verify Certificate C_CV.IFD.AUT	→ ←	Verify certificate. Store PuK.IFD.AUT Response OK
The Public Key of the IFD is now known by the ICC, and can be trusted				
2	F	If PuK.CA_{ICC}.CS_AUT is available in the IFD continue with step H		

	G	READ BINARY of file EF.C.CA _{ICC} .CS_AUT ^a	→ ←	Read data from specified file. C.CA.AUT is in response APDU.
	H	READ BINARY of file EF.C.ICC.AUT Verify C.ICC.AUT and store PuK.ICC.AUT temporarily	→ ←	Read data from specified file. C.ICC.AUT is in response APDU.
The Public Key of ICC is now known by the IFD, and can be trusted				
3	I	MSE:SET KeyId(PrK.ICC.AUT) for authentication ^b	→ ←	Select PrK of ESIGN application. Response OK
		MSE:SET KeyId(PuK.IFD.AUT) for encryption	→ ←	Response OK
	J	INTERNAL AUTHENTICATE (C) where $C = \text{RND.IFD} \parallel \text{SN.IFD}$ Decrypt with PrK.IFD.AUT and verify response from ICC with PuK.ICC.AUT	→ ←	ESIGN application calculates: SIG = DS[PrK.ICC.AUT] (... K _{ICC} h(... K _{ICC} C) ...)) E[PuK.IFD](min(SIG, N.ICC — SIG) C is from Internal Auth. command
IFD has now authenticated the card				
4	K	GET CHALLENGE	→ ←	RND.ICC
	L	EXTERNAL AUTHENTICATE IFD calculates: SIG = DS[PrK.IFD.AUT] (... K _{IFD} h(... K _{IFD} S) ...) E[PuK.ICC.AUT] (min(SIG, N.IFD — SIG)) with S = RND.ICC SN.ICC SN.ICC was received in the certificate of Step H.	→ → ←	Verify that K _{IFD} is different from K _{ICC} Verify the signature OK !
Two-way authentication is now complete Key derivation according to 8.4.2				
^a This step is required only if the IFD does not have the public key of the CA. ^b Both MSE commands of this step may be realized with one MSE command that transmits a CRT with both key references.				

8.4.1 Authentication Steps

The following are the essential steps of authentication with key transport. The given reference provides protection of the IFD's identity and data while the ICC is authenticated first.

8.4.1.1 Step A — Skip to authentication (conditional)

If the PuK.CA_{FD}.CS_AUT is available in the ICC, the steps B–C are not required to be executed.

8.4.1.2 Step B — Selection of verification key PuK.RCA.AUT (conditional)

If PuK.CA_{ICC}.AUT is not available in the ICC a chain of (cross-)certificates must be presented to the ICC in order to confirm the validity of each certificate. The chain of certificates ends at any key which is available in the ICC. The last possible key available in the ICC is the public key of the root CA, named PuK.RCA.AUT. Steps B and C have to be performed repeatedly until the correct PuK.CA.AUT is available in the ICC. This process is called multi-stage verification.

The multi-stage verification is necessary if and only (a) the CA is not the root CA (if it is the root CA, the key necessarily is present in the card), and (b) the appropriate public key of the CA is not present in the card.

Execution Flow

Table 36 — Selection of root key — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' SET for signature verification
P2	'B6' CRT for DST
Lc field	'xx'
Data field	'80' L ₈₀ AlgId '83' L ₈₃ CRT(PuK.RCA.AUT <or other key in chain>)

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.11.

The structure and coding of the key reference data is specified in 12.1.2.

The response data field is empty.

Table 37 — Selection of root key — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4.

8.4.1.3 Step C — Verify Certificate C_CV.CA.CS_AUT (conditional)

In multi-stage verification the certificate of a CA's public key C_CV.CA.CS_AUT is sent to the card with the VERIFY CERTIFICATE command. The ICC verifies this certificate with the public key selected in the previous step. If the verification fails, the ICC responds with an appropriate return code. On success, the ICC stores the PuK contained in the certificate and allows a further import of certificates.

The role of the certificate has to comply with its signer. The selection of PuK.RCA can always be accepted to verify a chained certificate. Any other key that was imported with certificate may be applied only if the role of its certificate matches the purpose of verification of the next incoming certificate.

Execution Flow**Table 38 — CA Certificate verification — command APDU**

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'00' response data field absent
P2	'AE' VERIFY CERTIFICATE
Lc field	'xx'
Data field	C_CV.CA.CS_AUT relevant data fields of the certificate

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

The structure of the certificate is specified in Subclause 14.15. The transmission of the certificate, however, is done without the tag '7F21' in compliance with the examples in ISO/IEC 7816-8:2004, Annex A.3. Only the relevant data objects from the value field of the certificate as described in 14.14 will be transmitted. These data objects may be transmitted with command chaining.

The response data field is empty.

Table 39 — CA Certificate verification — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4.

The ICC must verify evidence that the role for the PuK of the IFD either corresponds to signing another (subsequently transmitted) certificate or to the authentication mechanism. The role (refer to Clause 14) is part of the certificate when the PuK is imported. The role of the last PuK imported with VERIFY CERTIFICATE has to be stored in the ICC until the execution of the EXTERNAL AUTHENTICATE command in order to verification.

8.4.1.4 Step D — Selection of verification key PuK.CA_{IFD}AUT

The MSE command sets the key ID of the CA's public key to be used for the verification of the IFD's authentication certificate.

Execution Flow

Table 40 — Select key reference for certificate verification — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' SET for verification
P2	'B6' DST
Lc field	'xx'
Data field	'80' L ₈₀ AlgId '83' L ₈₃ CRT(PuK.CA _{IFD} .CS_AUT)

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.5.11.

The structure and coding of the key reference data is specified in Clause 12.1.2.

The response data field is empty.

Table 41 — Select key reference for certificate verification — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4.

8.4.1.5 Step E — Verify Certificate C_{CV}.IFD.AUT

The IFD sends a card verifiable (CV) certificate C_{CV}.IFD.AUT along with the VERIFY CERTIFICATE command. The ICC verifies the certificate using the public key PuK.CA_{IFD}.CS_AUT of the certification authority CA.

The certificate contains the IFD's PuK.IFD.AUT, which is stored temporarily by the ICC.

Execution Flow

Table 42 — IFD Certificate verification — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'00' response data field absent
P2	'AE' VERIFY CERTIFICATE
Lc field	'xx'
Data field	C _{CV} .IFD.AUT IFD's authentication certificate

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, Clause 5.8.

The structure of the certificate is specified in 14.14. The transmission of the certificate, however, is done without the tag '7F21' in compliance with the examples in ISO/IEC 7816-8:2004, Annex A.3. Only the relevant data objects from the value field of the certificate as described in Subclause 14.14 will be transmitted. These data objects may be transmitted with command chaining.

The response data field is empty.

Table 43 — IFD Certificate verification — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4.

The ICC stores the key in a temporary location for use in subsequent steps of the authentication, together with identification data such that the key can be recognized when specified in an MSE:SET APDU. The CHA has to be stored in order to identify the role of the key for further usage. See Clause 14.

The execution flow of the VERIFY CERTIFICATE is outlined in Subclause 14.12.

After Step E the IFD's public authentication key PuK.IFD.AUT can be trusted.

8.4.1.6 Step F — Skip reading chain certificates

If the public key PuK.CA_{ICC}.CS_AUT is available in the IFD or can be obtained by other means than reading it from the ICC, step G can be skipped.

8.4.1.7 Step G — Read C.CA_{ICC}.AUT (conditional)

The IFD obtains PuK.CA_{ICC}.CS_AUT by reading C.CA_{ICC}.CS_AUT from a file EF.C.CA_{ICC}.AUT in the ICC. Then it verifies the certificate C.CA_{ICC}.CS_AUT and extracts and stores the public key PuK.CA_{ICC}.CS_AUT temporarily for use in step H.

Execution Flow

Table 44 — Read ICC.CA certificate — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'83' Example: SFI = 03
P2	'00' offset = 0
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.2.3.

The structure and coding of C.CA_{ICC}.CS_AUT may be as described in Subclause 14.15, but it could also be some other format unknown to the ESIGN application. The actual format is application specific and out of the scope of this standard.

Table 45 — Read ICC.CA certificate — response

Response Parameter	Meaning
Data field	C_CV.CA.CS_AUT ICC's CA certificate
SW1-SW2	Refer to ISO/IEC 7816-4.

8.4.1.8 Step H — Read ICC's certificate C.ICC.AUT

The IFD reads the ICC's certificate to be used for later authentication.

The received certificate C.ICC.AUT contains the ICC's public key. The IFD verifies the certificate using the public key of the certification authority PuK.CA_{ICC}.CS_AUT. If C.ICC.AUT can be verified successfully, the PuK.ICC.AUT is stored in the IFD and can be trusted for subsequent authentication.

Execution Flow

Table 46 — Read C.ICC.AUT certificate — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'84' Example: SFI = 04
P2	'00' offset = 0
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2005, Clause 7.2.3.

Refer to Subclause 14.16 for more information concerning the structure and content of this certificate.

Table 47 — Read C.ICC.AUT certificate — response

Response Parameter	Meaning
Data field	data.EF.C.ICC.AUT ICC's CA certificate
SW1-SW2	Refer to ISO/IEC 7816-4.

8.4.1.9 Step I — Key selection

Before processing the INTERNAL AUTHENTICATE command the ICC's private authentication key PrK.ICC.AUT must be set by the MSE command.

Furthermore, the IFD's public key PuK.IFD.AUT needs to be selected for encryption in order to transport the ICC's key contribution K_{ICC} . K_{ICC} is a 32 byte random number generated by the ICC.

The MSE command updates the current security environment.