Specification

German
Health Professional Card
and
Security Module Card

Part 1: Commands, Algorithms and Functions of the COS Platform

Version 2.1.0

21.02.2006

BundesÄrzteKammer

Kassenärztliche Bundesvereinigung

BundesZahnÄrzteKammer

BundesPsychotherapeutenKammer

Kassenzahnärztliche Bundesvereinigung

Werbe- und Vertriebsgesellschaft Deutscher Apotheker mbH

Deutsche Krankenhaus-Gesellschaft

Zentralinstitut für die kassenärztliche Versorgung in der BRD

Editor: Bruno Struif, SIT

Revision History

Date	Version	Modifications
10.08.2005	V0.7	New version with enhacements: - CV certificates with 1024 and 1280 bits - MANAGE CHANNEL mandatory - Support of Key Usage Counter and reset with RESET (RETRY) COUNTER command
21.09.2005	V0.8	Changes: - Selection status after CREATE/ACTIVATE FILE - Le Handling - Clarification "security status evaluation counter" - Precisions, corrections and further information - RESET RETRY COUNTER extensions optional - CVCs with 1280 bits not yet mandatory - ENC and MAC computation for MUTUAL AUTHENTICATE
07.10.2005	V2.09	Alingment of version no.
07.11.2005	V2.099	Changes: - Addition of an Annex with Authentication Procedures (transfer of technical details of Part 2 to Part 1; no technical change) - Re-ordering of Annexes for alignment with Part 1 of eGK - Insertion of clause 14 for precising the Le and Lc handling
14.12.2005	V2.1	Changes: - Precision PIN management - CVC harmonization with eGK - Command handling after interrupt
21.02.2006	V2.1.0	Changes: - Some precisions for error cases - Harmonization with eGK V 1.1.0

Contents

1 Scope

This specification defines the minimum functionality of a COS platform usable for the German Health Professional Card (HPC) and the Security Module Card (SMC).

In particular, this specification defines

- commands and options
- algorithms and
- functions

on the basis of the ISO/IEC 7816 standards.

2 References

[ALGCAT]
Suitable Cryptographic Algorithms
Geeignete Algorithmen zur Erfüllung der Anforderungen nach §17 Abs. 1 bis 3 SigG vom 22. Mai 2001 in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV vom 22. November 2001, 30. März 2005, Bundesanzeiger Nr. 59, S. 4695-4696
See also www.bundesnetzagentur.de

[ANSI X9.19]
Financial Institution Retail: Message Authentication
1998

[ANSI X9.63]
Public Key Cryptography for the Finacial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography

[CWA14890-1]
Application Interface for SmartCards used as Secure Signature Creation Devices
Part 1 – Basic Requirements
March 8th 2004

[CWA14890-2]
Application Interface for SmartCards used as Secure Signature Creation Devices
Part 2 – Additional services
March 12th 2004

[DIN66291-1]
DIN V66291-1: 2000
Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG/SigV
Teil 1: Anwendungsschnittstelle

[ECC-2]
CEN TC224: European Citizen Card
Part 2: Logical Data Structures and Security Services
May 2005

[eGK-P1]
Die Spezifikation der elektronischen Gesundheitskarte
Teil 1 – Kommandos, Algorithmen und Funktionen der Betriebssystem-Plattform
V1.1.0, 07.02.2006

[ISO7816-1]
ISO/IEC 7816-1: 1996 (2nd edition)
Identification cards - Integrated circuit cards with contacts -
Part 1: Physical characteristics

[ISO7816-2]
ISO/IEC 7816-2: 1996 (2nd edition)
Identification cards - Integrated circuit cards with contacts -
Part 2: Dimensions and location of contacts
Amendment 1: Assignment of Contacts C4 and C8

[ISO7816-3]
ISO/IEC 7816-3: FCD 2004 (2nd edition)
Identification cards - Integrated circuit cards with contacts -
Part 3: Electrical interface and transmission protocols

[ISO7816-4]
ISO/IEC 7816-4: 2005 (2nd edition)
Identification cards - Integrated circuit cards -
Part 4: Organization, security and commands for interchange

[ISO7816-8]
ISO/IEC 7816-8: 2004 (2nd edition)
Identification cards - Integrated circuit cards -
Part 8: Commands for security operations

[ISO7816-9]
ISO/IEC 7816-9: 2004 (2nd edition)
Identification cards - Integrated circuit cards -
Part 9: Commands for card management

[ISO7816-12]
ISO/IEC 7816-12: FDIS 2004
Identification cards - Integrated circuit cards -
Part 12: Cards with contacts: USB electrical interface and operating procedures

[ISO7816-13]
ISO/IEC 7816-13: FCD 2006
Identification cards - Integrated circuit cards -
Part 13: Commands for application management in multi-application environment

[ISO8825]
ISO/IEC 8825-1: 1995
Information technology - ASN.1 encoding rules - Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

[ISO9564]
ISO 9564-1, Banking – Personal Identification Number management and Security, Part 1: PIN protection principles and techniques, 1999

[ISO9796-2]
ISO9796-2: 2002, Information technology – Security techniques – Digital signature schemes giving message recovery –
Part 2: Mechanisms using a hash function

[ISO10118]
ISO 10118-2, Information technology – Security techniques – Hash functions, Part 2: Hash functions using an n-bit block cipher algorithm, 2000

[ISO11770]
ISO/IEC 11770: 1996
Information technology - Security techniques - Key management
Part 3: Mechanisms using asymmetric techniques

[NIST-SHS]
NIST: FIPS Publication 180-2:
Secure Hash Standard
August 2002

[PKCS#1]
PKCS #1 RSA Cryptography Standard
V2.1: June 14, 2002

[PP-HPC]
Common Criteria Protection Profile – Health Professional Card (HPC)
BSI-PP-018, 05.12.2005

[PP-SMC]
Common Criteria Protection Profile – Security Module Card (SMC)

[Resolution190]
Beschluss Nr. 190 der Europäischen Union vom 18. Juni 2003 betreffend die technischen Merkmale der europäischen Krankenversicherungskarte

[RFC1510]
RFC 1510: May 1999
Public Key Cryptography for Initial Authentication in Kerberos

[RSA]
R. Rivest, A. Shamir, L. Adleman:
A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM, Vol. 21 No. 2, 1978

[SigG01]
Law Governing Framework Conditions for Electronic Signatures and Amending Other Regulations (Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften), Bundesgesetzblatt Nr. 22, 2001, S.876

[SigV01]
Ordinance on Electronic Signatures (Verordnung zur elektronischen Signatur – SigV), 2001, Bundesgesetzblatt Nr. 509, 2001, S. 3074

3 Abbreviations and notations

AID = Application Identifier
AlgID = Algorithm Identifier
AM = Access Mode
ARR = Access Rule Reference
AT = Authentication Template
ATR = Answer-to-Reset
AUT = Authentication
B = Byte
BCD = Binary Coded Decimal
BER = Basic Encoding Rules
C = Certificate
CA = Certification Authority
CAR = Certification Authority Reference
CBC = Cipher Block Chaining
CAMS = Card Application Management System
C2C = Card-to-Card
CC = Cryptographic Checksum
CCT = CC Template
CG = Cryptogram
CH = Cardholder
CHA = Certificate Holder Authorization
CHR = Certificate Holder Reference
CLA = Class byte of a command

COS = Card Operating System
CPI = Certificate Profile Identifier
CRT = Control Reference Template
CMS = Card Management System
CS = CertSign (= CertificateSigning)
CT = Confidentiality Template
CVC = Card Verifiable Certificate
CWA = CEN Workshop Agreement
D, DIR = Directory
DE = Data Element
DES = Data Encryption Standard
DO = Data Object
DF = Dedicated File
DI, Di = Baud rate adjustment factor
DSI = Digital Signature Input
DST = Digital Signature Template
ECB = Electronic Code Book
EF = Elementary File
eGK = elektronische Gesundheitskarte
ENC = Encipherment
EOF = End-of-File
FCP = File Control Parameter
FI, Fi = Clock rate conversion factor
FID = File Identifier
GP = Global Platform
HPC = Health Professional Card
HT = Hash Template
ICC = Integrated Circuit Card
ICCSN = ICC Serial Number
ID = Identifier
IFD = Interface Device
IFSC = Information Field Size Card
IFSD = Information Field Size Device
IV = Initial Value
KAT = Key Agreement Template
KE = Key Encipherment
KID = Key Identifier
Le = Length of expected data
LSB = Least Significant Byte(s)
MF = Master File
MSE = MANAGE SECURITY ENVIRONMENT
MSB = Most Significant Byte
OID = Object Identifier
PK,PuK = Public Key
PI = Padding Indicator
PIN = Personal Identification Number
PPS = Protocol Parameter Selection
PrK = Private Key
PSO = PERFORM SECURITY OPERATION
PV = Plain Value
QES = Qualified Electronic Signature
RC = Retry Counter
RD = Reference Data
RF = Radio Frequency
RFC = Request for Comment
RIPEMD = RACE Integrity Primitves Evaluation Message Digest
RND = Random Number
RSA = Algorithm of Rivest, Shamir, Adleman
S = Server
SC = Security Condition
SE = Security Environment
SEID = SE Identifier
SFID = Short EF Identifier
SHA = Secure Hash Algorithm
SIG = Signature
SK = Secret Key
SM = Secure Messaging
SMC = Security Module Card

SMK = SM key
SSC = Send Sequence Counter
TC = Trusted Channel
UQ = Usage Qualifier
VD = Verification Data
3DES = Triple DES

Notation:
Hexadecimal values are presented with quotes.

4 Commands

4.1 Conventions and Parameters

The following tables describe the commands and options to be supported by a COS for the HPC. The support of further commands and options is not excluded. The status words (status codes) to be supported are decribed in annex A.

Not all commands are currently used in HPC Part 2. However, since further applications shall be downloadable e.g. for use in hospitals, the subsequent commands shall be supported if not marked otherwise.
Hexadecimal values will be transmitted in "big endian" convention, i.e. the most significant byte (MSB) is transmitted first.

The CLA byte of the subsequent commands shall be encoded according to clause 5.1.1 of ISO/IEC 7816-4.

Table 1 - Commands for Selection

Commands for Selection					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'A4'	**SELECT** (Application or file)	- '02': Select EF under current DF - '04': Select by DF name	- '04': Return FCP - '0C': No response data	P1 = '02': File Id P1 = '04': AID	FCP, if P2 = '04' and Le present
'A4'	**SELECT** (Root, see conventions)	- '00' (native cards) - '04' (GP compliant cards)	- '04': Return FCP - '0C': No response data	P1 = '00': '3F00' P1 = '04': absent	FCP, if P2 = '04' and Le present
'70'	**MANAGE CHANNEL** see convention 4	- '0000' = Open a logical channel (min 4 channels to be supported), channel no. in response data field - '8000' = Closing a logical channel, channel no. in CLA		Absent	- If P1 = '00': Channel no. (1 B) - If P1 = '80': Absent

Conventions –

1. The root is in native cards the MF, in Global Platform compliant card the default selected application.
2. The behaviour of the card for P1 = '00' and File Id different from '3F00' is not specified here.
3. The command SELECT with P1 = '00' and P2 = '04' is only used in plaintext mode.
4. The command MANAGE CHANNEL shall only be used without SM.

Table 2 - Commands for Data Unit Handling

Commands for Data Unit Handling				
INS	**Name**	**P1 - P2**	**C-Data field**	**R-Data field**
'B0'	**READ BINARY**	- Short EFID and offset - Offset	Absent	Data; Le: see clause 14
'D6'	**UPDATE BINARY**	- Short EFID and offset - Offset	Data to be written	Absent

If Short EFID is used, then a successful completed command sets the denoted EF as current.

If the command with SFID is aborted due to a detected error, then the referenced file remains current, if the EF was already the current file. If another or a new file was addressed, then the situation remains undefined for the application system, i.e. the application system shall select the related EF in the next read or write command.

Table 3 - Commands for Record Handling

Commands for Record Handling					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'B2'	**READ RECORD**	- Record no.	- With/without Short EFID - Record no. in P1	Absent	Data without rec no. and length; Le: see clause 14
'DC'	**UPDATE RECORD**	- Record no.	- With/without Short EFID - Record no. in P1	Data to be written (the new length of a record with variable length is Lc)	Absent
'E2'	**APPEND RECORD**	- '00'	- With/without Short EFID	Record to be appended	Absent

For record oriented commands with Short EFID the conventions below Table 2 are also valid.

Table 4 - Commands for Data Object Handling

Commands for Data Object Handling					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'CA'	**GET DATA**	- '0040' – '00FF': BER-TLV tag in P2 - '4000' – 'FFFF': BER-TLV tag in P1-P2		Absent	Data without tag and length; Le: see clause 14
'DA'	**PUT DATA** see note 1	- '0040' – '00FF': BER-TLV tag in P2 - '4000' – 'FFFF': BER-TLV tag in P1-P2		Data to be written in the value field of the DO	Absent

NOTES –
1. If the DO exists, then it shall be replaced by the transmitted value, if allowed by the security conditions. The new length of the DO may be different from the existing DO.
2. The DOs are DF-specific.

Table 5 - Commands for Basic Security Handling

Commands for Basic Security Handling					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'88'	**INTERNAL AUTHENTICATE**	- '00'	- '00'	See Annex E	See Annex E
'84'	**GET CHALLENGE**	- '00'	- '00'	Absent (Le = '08')	8 Byte random number
'82'	**EXTERNAL AUTHENTICATE**	- '00'	- '00'	See Annex E	Absent
'82'	**MUTUAL AUTHENTICATE**	- '00'	- '00'	See Annex E	See Annex E
'20'	**VERIFY** see clause 7	- '00'	Verification data reference acc. to [ISO 7816-4], table 65	PIN in Format 2 PIN Block	Absent
'24'	**CHANGE REF. DATA** see clause 7	- '00'	see VERIFY	2 concatenated Format 2 PIN Blocks (for change of transport PIN see clause 7)	Absent
'2C'	**RESET RETRY COUNTER** see clause 7	- '00': Res. code || new PIN - '01': Resetting code - Further codings: see Annex F	see VERIFY	1 or 2 Format 2 PIN Blocks	Absent

'22'	**MANAGE SE** see MSE conventions	- b8-b5 acc. to [ISO 7816-4], table 78 - SET - RESTORE	- SEID (Restore) - 'A4': AT - 'B4': CCT - 'B6': DST - 'B8': CT - 'AA': HT	- Absent (Restore) - DO '80': AlgID - DO '83': KID.SK - DO '83': KID.PuK - DO '84': KID.PrK	Absent

General conventions:

1. The commands INTERNAL AUTHENTICATE, EXTERNAL AUTHENTICATE and MUTUAL AUTHENTICATE are used e.g. for authentication with Trusted Channel establishing and may contain enciphered data (see Annex E), but from the viewpoint of the IFD, the data are transparent.
2. The command GET CHALLENGE shall only be used without SM.

MSE conventions:

Option RESTORE:
- The command MSE with option RESTORE has to be sent always in plaintext mode.

Option SET:
- If the command data field contains a keyID and an AlgID, then the KeyID follows after the length field of the command data field. The AlgID - if present - denotes the algorithm for the key usage.
- If two keyIDs are set within the same MSE command, then an AlgID - if present - belongs to the secret or private key. Otherwise two separate MSE commands shall be used.
- If the Hash Template HT is referenced, then the data field contains the AlgID of the hash algorithm.

Table 6 - Commands for Security Operations

Commands for Security Operations					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'46'	**GENERATE ASYMMETRIC KEY PAIR** see conventions for key generation	- '82': Generate asym Key Pair and return PuK - '83': Read PuK - '86': Generate asym. Key Pair	- '00'	Absent	- If P1 = '82' or '83': PuK (Le present) - If P1 = '86': absent (Le absent)
'2A'	**PSO: COMPUTE DS**	- '9E'	- '9A'	- Absent (hash value already in the card via PSO:HASH command) - Digestinfo, if padding acc. to [PKCS#1] clause 9.2 shall be computed - Hash value, if padding acc. to ISO 9796 with RND shall be computed The length of the data in the data field (Digestinfo or hash value) shall not be longer than 40% of the length of the modulus of the signature key.	Signature
'2A'	**PSO: HASH** see conventions for hashing	- '90'	- 'A0'	- DO '90' (intermediate hash value = hash value of the data already hashed (x byte) || length of the counter (y byte)) indicating the number of bits already hashed) || DO '80' (final textblock)	Absent
'2A'	**PSO: VERIFY CERTIFICATE**	- '00'	- 'AE'	- DO '5F37' || DO '5F38' - Presentation of the above DOs in chained commands, if longer than 255 bytes - Certificates to be supported, see Annex B	Absent
'2A'	**PSO: ENCIPHER** see conventions for encipher/decipher	- '86'	- '80'	- Data to be enciphered (max. 254 byte)	- Enciphered data
'2A'	**PSO: DECIPHER** see conventions for encipher/decipher	- '80'	- '86'	- Data to be deciphered (PI || cryptogram of max. 254)	- Deciphered data

Conventions for key generation:

- The KeyId of the private key of the key pair to be generated shall be selected with the MSE command prior to the key generation
- With the key ID the key parameters are fixed in an issued card (key length and public exponent)
- The key parameters shall follow the requirements in [AlgCat], if relevant.
- The PuK data for an RSA Public key shall be coded according to [ISO7816-8] on the basis of an explicit or implicit "extended headerlist": '7F49'-L-'81'-L-'xx ...xx' || '82 0x .... ' = DO Public Key Data Objects (DO Modulus || DO Public Exponent, e.g. 65537)
- The above functionality shall be available not later than 2007.

Conventions for Hashing:

Table 7 – Values for last Round Computation in the Card (relevant for Command PSO: HASH)

Hash Algorithm	length of hash value of the data already hashed (x byte)	length of the counter (y byte) indicating the number of bits already hashed	block length of the hash algorithm in bytes	OID
SHA-1	20	8	64	{1 3 14 3 2 26}
RIPEMD160	20	8	64	
SHA-224	32	8	64	
SHA-256	32	8	64	{2 16 840 1 101 3 4 2 1}
SHA-384	64	16	128	
SHA-512	64	16	128	

If the message to be hashed is shorter than the block length of the hash algorithm, then the length of the DO with tag ‘90’ shall be set to zero (i.e. there is no intermediate hash value) followed by the DO with tag ‘80’ with the message to be hashed.

Conventions for ENCIPHER/DECIPHER:

The encipher/decipher algorithm is RSA (3DES optional).
PI = '00': If the padding indicator PI = '00' is used, the padding method and the maximal length of the command data field is defined by the relevant algorithm identifier (PI = '00' not used in the other parts).
PI = '81': The length of the command data shall be less than N-11 Byte, if the padding indicator PI = '81'. Table 8 shows the format for key encipherment input.

Table 8 – Format für Key Encipherment Input

PI	Key Encipherment Input	Specification
'81'	'02' || RND (all byte not equal zero., amount depends on key length) || '00' (Separator) || Data to be enciphered	[PKCS#1], Clause 7.1.2, "EME-PKCS1v1_5"

Table 9 - Commands for Card Management

Commands for Card Management					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'EA'	**LOAD APPLICATION** (see [ISO7816-13] and note 1)	- '00'	- '00'	- DO Command-to-perform, tag '52'	Response data or absent
'E0'	**CREATE FILE (DF)** see notes 2 and 3	- '38'	- '00' (no information given)	File control parameter	Absent
'E0'	**CREATE FILE (EF)** see note 2 and 3	- '00'	- '00' (no information given)	File control parameter	Absent
'E4'	**DELETE FILE**	- '00': Delete current DF (DF shall be selected before and there shall be no current EF to be ensured by the IFD) - '02': Delete EF under current DF	- '00' (no information given)	- P1 = '00': Absent - P1 = '02': EFID	Absent
'04'	**DEACTIVATE FILE** see note 4	- '00': Deactivate current DF (DF shall be selected before and there shall be no current EF to be ensured by the IFD) - '02': Deactivate denoted EF under current DF	- '00' (no information given)	- P1 = '00': Absent - P1 = '02': File Id	Absent
'44'	**ACTIVATE FILE** see note 4	- '00': Activate current DF (DF shall be selected before and there shall be no current EF to be ensured by the IFD) - '02': Activate denoted EF under current DF	- '00' (no information given)	- P1 = '00': Absent - P1 = '02': File Id	Absent

The selection status (file selected or not selected) after the processing of the commands CREATE FILE, ACTIVATE FILE and DEACTIVATE FILE is not fixed in this specification and remains COS specific, i.e. a software system assumes a non-selected state.

NOTES -
1. This command is not mandatory, but strongly recommended (usage specified in HPC Part 2; beside of wrapping commands like CREATE FILE, ACTIVATE FILE etc. it may also be used for loading "application images", i.e. file control blocks and data can be written in the memory without performing commands like CREATE FILE). For the LOAD APPLICATION command with its embedded commands only one access rule applies.
2. The Parameter P1 = '00' should be used only when an EF is created. However, according to ISO7816-9 the card is not obliged to reject the command, if the parameter P1 = '00' is used in connection with a file descriptor byte '38' transmitted in the data field of the command.
3. For HPCs with native platform this command is mandatory. The structure and content of the file control parameters remain COS-specific.
4. This command is not only relevant for card management.

4.2 Handling of Commands after Interrupt

If data in the persistent memory shall be changed by a command and the command is interrupted so that the data has been updated only partially, then internal recovery mechanisms shall ensure that either the state before command processing (roll-back) or the state after command processing (roll-forward) is achieved before processing any further command.

For the following commands a roll-back shall be performed:

- UPDATE BINARY
- UPDATE RECORD
- APPEND RECORD
- PUT DATA
- CHANGE REFERENCE DATA
- GENERATE ASYMMETRIC KEY PAIR
- LOAD APPLICATION
- CREATE FILE
- ACTIVATE FILE.

Roll-back is not valid for data like usage counter or error counter. For these data a roll-forward shall be performed.
If the interrupt occurs while transmitting response data, then no roll-back has to be performed. An interrupted GENERATE ASYMMETRIC KEY PAIR command may lead to a situation that the respect-tive key is not usable.

For the following commands a roll-forward operation (i.e. command completion) shall be performed, if the update process has been already initiated:

- DEACTIVATE RECORD
- DEACTIVATE FILE
- DELETE FILE.

5 File Organization

The file management shall be in accordance to [ISO7816-4] and support the following functionality:

- MF level and DF level (deeper DF levels optional)
- EFs with transparent structure
- EFs with linear fixed record structure
- EFs with linear variable record structure
- EFs with cyclic structure

- record length of 1 byte and up to 255 byte
- maximum number of records possible in a file: at least 254
- EFs with short EFID (usable only in the respective DF or application); according to [ISO7816-4], a SFID may be different from the 5 least significant bits of a FID (e.g. FID = '2F00', SFID: 30)
- DF with AID, see Table 1.

The following File Control Parameter (FCP, coding strongly recommended according to [ISO 7816-4], table 12), have to be supported:

- no of data bytes (EF) without structural information, see DO with tag '80'
- File descriptor byte, see DO with tag '82'
- File identifier, see DO with tag '83'
- AID, see DO with tag '84' (this DO may occur twice, e.g. a health care application may have a national and an international AID)
- Short EF Identifier, see DO with tag '88'
- Life cycle status byte, see DO with tag '8A'
- Security attribute referencing the expanded format, see DO with tag '8B' and table 25 of [ISO 7816-4]
- Security attribute template for data objects, see DO with tag 'A0'

6 Security Attributes and Access Rules

Since smartcards are personal security entities, the concept of security attributes is an essential feature of every card. Especially for electronic health cards, a differentiated and granular encoding scheme for access modes (AM) and security conditions (SC) is required.

The security management has to support the following functionality (coding strongly recommended according to [ISO 7816-4]):

- expanded format
- access mode byte for DFs
- access mode byte for EFs
- access mode byte for DOs
- access mode DOs with tag ‘80’ to ‘8F’
- security condition data objects, see table 23 of [ISO 7816-4]; support of security condition byte (tag ‘9E’), NOT-template (tag ‘A7’), and CRT with tag ‘B6’ indicating asymmetric SM is optional
- elementary file with access rule references (EF.ARR)

Cards not supporting an EF.ARR shall provide a mechanism, which is functionally equivalent.

An EF.ARR belonging to the Root is located under the Root. An EF.ARR belonging to a dedicated application is located under the respective DF.

Access rules in an EF.ARR can be modified or supplemented, if allowed by the security conditions. It may be necessary, to put access rules not modifiable in a separate EF.ARR und those modifiable in another EF.ARR, i.e. security environment, record number and file identifier shall be specifiable in the security attribute data objects referencing expanded format (see [ISO/IEC 7816-4], table 25).

A command shall only be processed, if the security conditions according to the related access rule are satisfied or if the command is allowd always due to an implicit convention. If no TC is established, then the following commands are allowed always in plaintext mode:

- SELECT
- MSE
- GET CHALLENGE
- PSO: HASH
- INT. / EXT. AUTHENTICATE with Public Keys imported via VERIFY CERTIFICATE.

In a Trusted Channel only the following commands sent in SM mode are allowed always:

- SELECT (EF)
- MSE Operation SET (i.e. a change of the SE is not allowed)
- PSO: HASH

If the card contains a dual interface chip (a contact-based interface according to [ISO7816-3] and a RF interface), then a COS shall be able to restrict the usage of an application to a specific interface (see [ISO7816-4]).

7 PIN Management

For user verification a HPC or SMC has to store PIN objects. The way of storing PIN objects is manufacturer specific. At the card interface the follwing attributes of a PIN object are visible:

1. **Reference value:** The reference value represents the user's secret. Only digits are used, i.e. a Personal Identification Number (PIN). During personalization of the card, a PIN is usually stored as transport PIN (the transport PIN convention is manufacturere specific. A PIN can be changed with a CHANGE REFERENCE DATA command. If allowed by the respective access rule, the setting of a new PIN is possible with the respective option of the command RESET RETRY COUNTER.

2. **PIN reference and PIN type:** A PIN reference is encoded in one byte. Global and DF-specific (local) PINs have to be distinguished according [ISO7816-4]. A PIN reference is stored during personalization. At least 3 PINs shall be supported coexistantly in a DF.

3. **Transmission format:** The transmission format determins the coding for the PIN in the data field of the commands VERIFY, CHANGE REFERENCE DATA and RESET RETRY COUNTER. It is fixed during personalization and shall not be modifiable afterwards. The mandatory transmission format is the Format 2PIN Block according to ISO 9564-1, see example for a 5-digit PIN:

C	L	P	P	P	P	P	F	F	F	F	F	F	F	F	F

   C = Control field, value 2
   L = Length of PIN in BCD
   P = PIN digit in BCD
   F = Filler with value ’F’

   The coding format has therefore always 8 byte. The correctness of the format has to be controlled, if the existing PIN is changed or a new PIN is set.

4. **Minumum length:** This attribute (set during personalization in a manufacturer specific way) is used by the commands CHANGE REFERENCE DATA and RESET RETRY COUNTER. The minimum length can be chosen in the intervall [4, 12].

5. **Maximum length:** The maximum length of a PIN is 12 digits. Any length shorter than 12 has to be controlled outside the card, e.g. by a card terminal.

6. **Initial value of the retry counter**: This attribute (set during personalization in a manufacturer specific way) is used by the commands VERIFY, CHANGE REFERENCE DATA und RESET RETRY COUNTER. The value can be chosen in the intervall [1, 15]. The usual initial value of the retry counter is 3.

7. **Retry counter:** This attribute (set during personalization in a manufacturer specific way) is used by the commands VERIFY, CHANGE REFERENCE DATA und RESET RETRY. The actual value is a value in the intervall [1, 15]. The retry counter is managed by a COS state:
   - if the value of the retry counter is zero, then the PIN is blocked (see resetting code)
   - if the value of the retry counter is not zero and the entered PIN is true, then the retry counter is set to its initial value and a security state is set: "PIN with PIN reference 'xx' successfully

presented"
- if the value of the retry counter is not zero and the entered PIN is false, then the retry counter is decremented and the remaining retries x have to be indicated with the Status-Code '63C*x'*
- a syntax error in the transmission format (Status-Code '6A80') shall have no impact on the retry counter.

8. **Resetting code:** This attribute (set during personalization in a manufacturer specific way) is used by the command RESET RETRY COUNTER. The transmission format is the same as for a PIN. The length of a resetting code to be supported is between 8 and 12 digits. The presence of the resetting code depends on the desired PIN attributes.

9. **Usage limitation of the resetting code**: This attribute (set during personalization in a manufacturer specific way) is used by the command RESET RETRY COUNTER. The value can be chosen in the intervall [1, 15].

If a CHANGE REFERENCE DATA command with a transport PIN is sent to the card, then a security status may be set. However, the application system should sent a VERIFY command before accessing PIN protected data or functions.

If logical channels are used then any presentaion of a global or local PIN is only relevant for the respective channel.

For the usage of the private key for qualified electronic signatures, a "security status evaluation counter" shall be supported, i.e. it shall be possible to configure that

- each time or
- after n times (n in the range 1 ... 254) or
- only once

before using the private signature key a user verification with PIN.QES is required during a session. The initial value of this "security status evaluation counter" will be fixed during personalization. The security status evaluation counter and its initial value is related to objects like a private key, but refers to the PIN management.

8 Security Environments

A SE is used for referencing cryptographic algorithms, mode of operations, protocols, procedures, keys and additional data needed for secure messaging and for security operations. Different SEs in the same application allow the coexistence of different sets of security conditions when using the same objects (e.g. keys and EFs) and to specify specific cryptographic algorithms only available or to be processed in the respective SE. Per channel, there is only the current SE (implicitly or explicitly selected) valid.

A Security Environment is selectable and identified by a SE identifier.

The SE management has to support the following functionality:

- SE #1 is the default SE
- at least three SEs in the same DF (support of MSE RESTORE)
- an explicit or implicit selected SE at the MF level remains valid for the MF also in case an application has been selected, i.e. there is a current SE at MF level and a current SE at DF level after DF selection.

Subject of SEs are different access rules e.g. for the same data object or key.

9 Secure Messaging

The SM management has to support the functionality as described in Annex C.

10 Security Status and Situation after DF Selection

The following security states have to be supported:

- status related to the successful presentation of at least one global and 2 DF-specific PINs
- status related to the successful presentation of min 3 global and 3 DF-specific keys (including CHAs (7 byte) with role identifier not equal 0).

A PIN-related security state may also be influenced by a usage counter (e.g. if a signature counter has the value 1 and a PSO: COMPUTE DS command is performed, then the respective security status has e.g. to be reset in order to enforce a new PIN presentation).

After successful DF selection (the SELECT command is performed as command without SM),

- the global security status is maintained
- selection state related to global keys or a SE at MF level remains valid
- specific states related to a previous selected DF are lost (exception: if the same application is reselected, then specific states may be maintained) and
- SM keys are no longer available.

11 Algorithms and Key References

The HPC shall support the following algorithms:

- RSA with 1024 bits for CV certificate management (support of longer keys, e.g. 1280 bits, will become mandatory in near future)
- RSA with length according to actual state of algorithm catalogue [ALGCAT]
- RSA digital signature input formats:
  - PKCS #1 (for signatures related to keys with X.509 certificates, commands PSO: COMPUTE DS and INTERNAL AUTHENTICATE)
  - ISO 9796-2 (with random no. for signatures related to keys with x.509 certificates and without random no. for CV certificate processing, commands PSO: COMPUTE DS and PSO: VERIFY VERTIFICATE)
- SHA-1 (usage in accordance with [ALGCAT] required, see also Table 8)
- 3DES (DES-3)
- CVC-based asymmetric authentication procedures with and without trusted channel establishment, see Annex E
- symmetric authentication procedures with trusted channel establishment, see Annex E
- challenge/response procedure for external authentication with a symmetric key, see Annex E
- asymmetric client/server authentication with X.509-certificates, see Annex E.

The support of elliptic curves for qualified electronic signatures is optional. However, a coexistence of HPCs with RSA and/or elliptic curves is envisaged for the next HPC generation.

With respect to hash algorithms for (qualified) electronic signatures, the card has to support the construction of the Digital Signature Input according to Tab. 8.

Table 10 – AlgIDs for hash functions and signature algorithms for PSO: COMPUTE DS

AlgID	Meaning	Support in HPC
'1x' (default)	SHA-1 (160 bit)	Algorithm and padding
'2x'	RIPEMD160	
'3x'	SHA-2 (224 bit)	
'4x'	SHA-2 (256 bit)	at least padding
'5x'	SHA-2 (384 bit)	
'6x'	SHA-2 (512 bit)	
'x0'	Hash Algorithmus x (selektierbar mit MSE SET HT)	at least SHA-1
'x1'	RSA with DSI according to ISO/IEC 9796-2 RND with hash alg x	DSI support mandatory, see note
'x2'	RSA with DSI according to PKCS #1 with hash alg x	DSI support mandatory, see note

NOTE - Currently only padding according to PKCS # 1 is used.
For each key, at least one purpose applies (e.g. key usage according alg. x in CRT DST). Special keys may be used for more than one purpose, if appropriate (e.g. usage with different padding schemes). At least 2 purposes shall be supportable.

Key referencing:

A cryptographic key may be referenced by a 1-byte key identifier or a key name. The key name may be 12 byte long. For PINs a 1-byte PIN reference is used.

12 Control Reference Templates

The following CRTs have to be supported:

- CRT for authentication, tag 'A4'
- CRT for cryptographic checksum, tag 'B4'
- CRT for hash computation, tag 'AA'
- CRT for digital signature, tag 'B6'
- CRT for confidentiality, tag 'B8'

Within a CRT, at least the following DOs shall be supported:

- Cryptographic mechanism reference (AlgID), tag '80'
- Key references, tags '83' and '84' (not relevant for CRT HT)
- Usage qualifier, tag '95'.

13 Key Usage Counter

The support of key usage counters for authentication keys is required. The usage counter shall be decremented by the COS when a command is performed, which uses the respective key (e.g. INTERNAL AUTHENTICATE). If the key usage counter has the value zero, then the related key is blocked.

The way a key usage counter is reset to its intial value is currently not fixed and will become subject of further ISO standardization. Therefore the implementation example in Annex F is only of informative nature.

14 Handling of Lc and Le

14.1 Short Length

Short length shall be handled according to [ISO 7816-4]:

- Lc ≠ '00': the command data field contains data of the indicated length
- Le ≠ '00': the response data field shall contain data of at most the indicated length
- Le = '00':
  a) Response data with length ≤ 256 byte
  the response data field contains ≤ 256 byte; if a record is read, the complete record data are returned; if a data object is retrieved with the GET DATA command, the complete value of the referenced DO is returned

  b) Data in a transparent file with length > 256 byte
  Data with length 256 byte are returned

  c) Response data with ≤ 256 byte, but with SM more than 256 byte
  In this case extended length shall be used.

14.2 Extended Length

The support of extended length has to be indicated in the card capabilities of the ATR. Furthermore, an EF.ATR has to be present with at least the data object shown in Table 9. It has 4 embedded DOs (Tag '02' = Integer value, length field 1 Byte with value '02' or '03', value field = max. number of bytes of the respective APDU).

Table 11 – Data object input/output buffer size

Tag	Länge	Wert
'E0'	'xx'	'02' -L-'xxxx' || '02' -L-'xxxx' || '02'-L-'xxxx' || '02'-L-'xxxx' = - DO max. length of command APDU without SM || - DO max. length of response APDU without SM || - DO max. length of command APDU with SM || - DO max. length of response APDU with SM

NOTE - The coding is compatible to [ECC-2].

With respect to the handling of the Lc and Le fields the same conventions as described in 14.1 applies, but instead of 256 byte the max. value of 65536 byte is used with the restrictions denoted by the data object shown in Table 9.

15 Technical Characteristics and Transmission Handling

An HPC is a normal size card (ID1 card) and shall support at least class AB (5V-3 V). The dimensions and location of contacts shall be in accordance to [ISO7816-2].

The transmission handling shall support the following functionality as specified in [ISO7816-3]:

- Transmission Protocol T=1
- NAD Byte: not interpreted (NAD shall be set to '00' by the IFD, but the card is not obliged to control this)
- S-Block ABORT: does not occur in normal situations; in case a chain is too long (I/O buffer size not observed), an ABORT may be sent by the card

- Protocol Parameter Selection (PPS), support of negotiable mode
- Clock rate conversion factor (Fi) and baud rate adjustment factor (Di) with FI/DI- and Fi/Di-values according to Table 10
- Information Field Size after PPS: IFSC = 254 byte, IFSD = 254 byte
- ATR coding shall comply to [ISO7816-3] taking into account requirements of this clause

Table 12 - FI/DI values, of which one of them has to be indicated in the ATR

Fi/Di	FI/DI (TA1)	Baudrate in kbps	MHz
372/12	18	115,2 / 161,3	3,5712 / 5
512/16	95	156,2	5
512/32	96	312,5	5

For backward compatibility cards shall support also the respective values shown in table 11, which may be chosen in a PPS procedure performed by a card terminal not yet able to support baudrates of 115 kbps and above:

Table 13 – Additional FI/DI values to be supported in a PPS procedure

TA1 in ATR	Fi/Di	FI/DI (TA1 in PPS)	Baudrate in kbps	MHz
TA1 = 18	372/2	12	19,2	3,5712 / 5
TA1 = 18	372/4	13	38,4	3,5712 / 5
TA1 = 95 or 96	512/2	92	19,5	5
TA1 = 95 or 96	512/4	93	39,1	5
TA1 = 95 or 96	512/8	94	78,1	5

16 Command Chaining

Command chaining (bit b5 in CLA) shall be supported by the following commands:

- PSO: VERIFY CERTIFICATE (in case the CV certificate is longer than the input buffer)
- LOAD APPLICATION.

If command chaining is used with SM, each command APDU and each response APDU is to be protected according to the rules in Annex C.

A command sequence shall be sent completely by the IFD, before sending a command in another channel.

17 Personalization, Card Management and Downloading

The commands for card management, application management and personalization process remain manufacturer specific. However the card management commands according to Table 9 shall be supported after card issuing, except LOAD APPLICATION, which is recommended.

The download mechanism shall support an authentication procedure and the establishment of a trusted channel (see secure messaging, Annex C). Furthermore, this mechanism shall be performed in such a way, that the HPC remains usable even in the case, that the download process was not successfully completed due to an interrupt (i.e. previously existing and unchanged DFs have to be fully functional even after interruption of download process).

The download management has to support the following functionality after card issuing:

- adding of an EF within an existing DF
- adding of a DF with its substructure
- key import
- key generation, see Table 6.

The COS platform shall allow also the deletion of DFs and EFs.

With respect to memory management, an existing application area (DF area) shall be dynamically extendable when creating an elementary file, if free physical memory is available. This means, that during personalization the application area for the defined applications shall not be fixed (recommended implementation: DO '81' absent indicates the dynamic memory usage). This does not exclude, that the COS platform may also support a fixed application area size.

18 Evaluation

An evaluation of the HPC and SMC on the basis of the final version of [PP-HPC] respective [PP-SMC] is required.

19 Conformance Tests

The description of conformance tests are not subject of this specification.

20 Requirements for SMCs

The specific functionality, which goes beyond that one for the HPC and only required for SMCs, is

- the production of secured commands and
- the processing of secured responses

for supporting a trusted channel between a SMC and a HPC or a SMC and an eHC. Therfore the support of the following additional commands is required for SMCs, whereby either the PSO commands or the ENVELOPE command are mandatory:

The SMC shall support extended length.

Table 12 – Additional Commands for SM support

Commands for Card Management					
INS	**Name**	**P1**	**P2**	**C-Data field**	**R-Data field**
'2A'	**PSO: COMPUTE CC**	- '8E'	'80'	Data for which the CC shall be computed	Cryptographic checksum
'2A'	**PSO: VERIFY CC**	- '00'	- 'A2'	'80'-L-PV || '8E'-L-CC	Absent
'C3'	**ENVELOPE** (special usage for the production of SM DOs for the command APDU)	- '00'	- '00'	- DO '52' with command to be secured || DO '7D' (SM template) with embedded DO 'BA' (Response descriptor), which contains '8700' (T-L of DO CG, if command data to be encrypted) || '8E00' (T-L of DO CC)	- DO '7D' (SM template) with DO '87' (CG, if command data to be encrypted) and DO '8E' (CC)
'C3'	**ENVELOPE** (special usage for the processing of SM	- '00'	- '00'	- DO '7D' (SM template) with embedded SM-DOs (see below) and DO 'BA' (Response descriptor) with '8000' (TL of DO PV), if encrypted data in	- absent - DO '7D' (SM template) with embedded DO '80'

	DOs of the response APDU)			secured response APDU SM-DOs: - '9902 xxxx' (DO SW) || '8E 0x xx..xx' (DO CC) or - '81'-L-'xx..xx' (DO with data, over which the CC has been computed) || '8E 0x xx..xx' (DO CC) or - '87'-L-'xx..xx' (DO CG) || '8E 0x xx..xx' (DO CC) || 'BA028000' (DO Response Descriptor with T-L of DO PV)	(PV of encrypted response data)

The usage of these commands is outlined in Annex E.

Annex A
(normative)

Status Codes

A.1 General requirements

The following tables identify the error conditions that shall be recognized by the COS for the HPC, and the associated status codes to be returned. The error codes and their meaning comply with ISO/IEC 7816-4.

Additional implementation specific error conditions and codes may be present, but are out of scope of this specification.

A.2 Status Codes

Table D.1 describes general error codes to be supported.

Table A.1 - General Error Codes

Error Condition	Status Code
Execution error	’64 00’
Memory error when reading or writing data	‘65 81’
Lc is not allowed for command variant	’67 00’
Lc is not consistent with length of command data	’67 00’
Lc or Le present while is has to be absent	’67 00’
Lc or Le absent while it has to be present	’67 00’
Logical channel not supported	’68 81’
Command chaining not supported (instead of '688A' also '6E00' may be used)	’68 84’
Security status not satisfied	’69 82’
Expected SM DO missing	‘69 87’
Incorrect SM DO	’69 88’
Incorrect parameters P1- P2	‘6A 86’
INS not supported	’6D 00’
Class not supported	‘6E 00’

Tables A.2 – A.27 describe command specific status codes to be supported.

Table A.2 - Status Codes for SELECT

SELECT	Status Code
File (DF or EF) to be selected not found	‘6A 82’
Selected file deactivated	‘62 83’

Table A.3 - Status Codes for MANAGE CHANNEL

MANAGE CHANNEL	Status Code
(Further) Logical channel not supported	‘68 81’

Table A.4 - Status Codes for READ BINARY

READ BINARY	Status Code
EOF reached before reading Le ≠ 00 bytes	’62 82’
File not transparent	‘69 81’
Command option without SFI and no current EF selected	’69 86’
File referenced by SFI not found	‘6A 82’
Offset greater or equal file size	‘6B 00’

Table A.5 - Status Codes for UPDATE BINARY

UPDATE BINARY	Status Code
File not transparent	‘69 81’
Command option without SFI and no current EF selected	’69 86’
File referenced by SFI not found	‘6A 82’
Offset + Lc > file size	‘6A 87’
Offset greater or equal file size	‘6B 00’

Table A.6 - Status Codes for READ RECORD

READ RECORD	Status Code
Selected record deactivated	’62 83’
File is transparent	‘69 81’
Command option without SFI and no current EF selected	’69 86’
File referenced by SFI not found	‘6A 82’
Record not found	‘6A 83’

Table A.7 - Status Codes for UPDATE RECORD

UPDATE RECORD	Status Code
Selected record deactivated	’62 83’
File is transparent	‘69 81’
Command option without SFI and no current EF selected	’69 86’
File referenced by SFI not found	‘6A 82’
Record not found	‘6A 83’
Not enough memory space	‘6A 84’

Table A.8 - Status Codes for APPEND RECORD

APPEND RECORD	Status Code
File is transparent or already contains the maximum number of records	‘69 81’
Command option without SFI and no current EF selected	’69 86’
File referenced by SFI not found	‘6A 82’
Not enough memory space	‘6A 84’

Table A.9 - Status Codes for GET DATA

GET DATA	Status Code
Data object not found	‘6A 88’

Table A.10 - Status Codes for PUT DATA

PUT DATA	Status Code
Not enough memory space	‘6A 84’

Table A.11 - Status Codes for INTERNAL AUTHENTICATE

INTERNAL AUTHENTICATE	Status Code
Referenced data (especially the key) not found, see note	‘6A 88’

NOTE- A wrong key reference may be already detected by a MSE command.

Table A.12 - Status Codes for EXTERNAL AUTHENTICATE

EXTERNAL AUTHENTICATE	Status Code
Referenced data (especially the key) not found, see note below Table 12	‘6A 88’

Table A.13 - Status Codes for MUTUAL AUTHENTICATE

MUTUAL AUTHENTICATE	Status Code
Referenced data (especially the key) not found, see note below Table 12	‘6A 88’

Table A.14 - Status Codes for VERIFY

VERIFY	Status Code
Authentication method blocked	’69 83’
PIN verification failed (PIN value incorrect), futher allowed retries in ‘X’	’63 CX’
Format error in format 2 PIN block in command data	‘6A 80’
Referenced data (especially the PIN) not found	‘6A 88’

Table A.15 - Status Codes for CHANGE REFERENCE DATA

CHANGE REFERENCE DATA	Status Code
PIN verification failed, futher allowed retries in ‘X’	’63 CX’
Authentication method blocked	’69 83’
Format error in format 2 PIN blocks in command data	‘6A 80’
Referenced data (especially the PIN) not found	‘6A 88’

Table A.16 - Status Codes for RESET RETRY COUNTER

RESET RETRY COUNTER	Status Code
Authentication method blocked	’69 83’
PIN verification failed, futher allowed retries in ‘X’	’63 CX’
Format error in format 2 PIN block(s) in command data	‘6A 80’
Referenced data (especially the PIN or the key) not found	‘6A 88’

Tabelle A.17 – Status Codes für MANAGE SE

MANAGE SE	Status Code
Referenced data (especially the key) not found, see note	‘6A 88’

NOTE - A key reference may be only set and evaluated afterwards by the command which uses the key.

Table A.18 - Status Codes for GENERATE ASYMMETRIC KEY PAIR

GENERATE ASYMMETRIC KEY PAIR	Status Code
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.19 - Status Codes for PSO: COMPUTE DS

PSO: COMPUTE DS	Status Code
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.20 - Status Codes for PSO: HASH

PSO: HASH	Status Code
Format error in command data	'6A 80'

Table A.21 - Status Codes for PSO: VERIFY CERTIFICATE

PSO: VERIFY CERTIFICATE	Status Code
Format error in command data	'6A 80'
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.22 - Status Codes for PSO: ENCIPHER

PSO: ENCIPHER	Status Code
In case of RSA: input out of range	'6A 80'
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.23 - Status Codes for PSO: DECIPHER

PSO: DECIPHER	Status Code
In case of RSA: input out of range	'6A 80'
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.24 - Status Codes for LOAD APPLICATION

LOAD APPLICATION	Status Code
Use return codes of the encapsulated commands	'xxxx'

Table A.25 - Status Codes for CREATE FILE

CREATE FILE	Status Code
Command incompatible with file structure	'69 81'
Not enough memory space	'6A 84'
File exists already	'6A 89'
DF name exists already	'6A 8A'

NOTE – '6981' or '6A89' or both may be used in case the file exists already.

Table A.26 - Status Codes for DELETE FILE

DELETE FILE	Status Code
File to delete not found	'6A 82'

Table A.27 - Status Codes for DEACTIVATE FILE

DEACTIVATE FILE	Status Code
File (EF) not found	'6A 82'

NOTE - The status code applies only to EF, because a DF has to be already selected.

Table A.28- Status Codes for ACTIVATE FILE

ACTIVATE FILE	Status Code
File (EF) not found, see note below Table 27	'6A 82'

Table A.29 - Status Codes for PSO: COMPUTE CC

PSO: COMPUTE CC	Status Code
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.30 - Status Codes for PSO: VERIFY CC

PSO: VERIFY CC	Status Code
Referenced data (especially the key) not found, see note below Table 12	'6A 88'

Table A.31 - Status Codes for ENVELOPE

ENVELOPE	Status Code
Incorrect secure messaging data	'69 88'

Annex B
(normative)

Card Verifiable Certificates (CV Certificates)

B.1 Principle structure

The principle structure of a card verifiable certificate (CV certificate) is shown in the subsequent table. The sequence of data elements can be described by a headerlist as defined in [ISO/IEC 7816-8]. This requires a fixed length of each data element.

Table B.1 – Certificate content and certificate headerlist

Certificate Content	Certificate Profile Identifier (1 byte)	Certification Authority Reference (8 byte)	Certificate Holder Reference (12 byte)	Certificate Holder Authorization (7 byte)	OID.PuK (x byte)	PuK (modulus tag '81', exponent tag '82') (x byte)
Headerlist Content	'5F29 01'	'42 08'	'5F20 0C'	'5F4C 07'	'06 0x'	'7F49 xx || 81 xx || 82 xx'

B.1.1 Certificate Profile Identifier

The "Certificate Profile Identifier (CPI)" has the purpose to denote the exact structure of a CV certificate. It can be considered as an identifier of a card internal headerlist describing the concatenation of the data elements including their length so that e.g. the Public Key (PuK) in a CV certificate can be found by the certificate verifying card.

B.1.2 Certification Authority Reference (Authority Key Identifier)

The „Certification Authority Reference (CAR)" has the purpose of identifying the certificate issuing Certification Authority (CA) with a distinguished name in such a way that the data element (DE) can be used as an authority key identifier for referencing the PK to be applied for the certificate verification. The CAR consists of

- the CA name (country code according to ISO 3166 (2 Bytes, DE = Deutschland) followed by an acronym of the CA (3 Bytes, ASCII characters)) and
- an extension for key referencing (3 Bytes).

Table B.2 – Structure of the Certification Authority Reference (Authority Key Identifier)

CA Name (5 byte)	Extension for key referencing (3 byte)

The extension has the following structure:

Table B.3 – Structure of the extension for key referencing

Service Indicator (1 BCD)	Discretionary Data (1 BCD)	Algorithm Reference (2 BCD)	Date (last two digits of key generation year) (2 BCD)

The Service Indicator has the value 1 = entity authentication according to the key usage in x.509v3 certificates.

The Discretionary Data may have a value at the discretion of the related CA.

The Algorithm Reference can be individually assigned by a CA for distinguishing different public key algorithms.
The Date consist of the last two digits of the year, in which the key pair for certificate signing was produced. If more than one key pair has been generated, it may be distinguished by using the discretionary data field.

B.1.3 Certificate Holder Reference (Subject Key Identifier)

The „Certificate Holder Reference (CHR)“ has the purpose to denote the certificate holder uniquely in such a way that the data element can be used as a subject key identifier for referencing the PK of the certificate holder. The two possible structures of CHR are outlined in the Tables B.4 and B.5.

Table B.4 – Structure of the Certificate Holder Reference, if certificate holder is a CA

Filler (4 byte)	CA Name (5 byte)	Extension for key referencing (3 byte)

A filler byte is coded ‘00’.

The „Extension for key referencing“ has the same structure as shown in tab. B.3. The field „date“ contains the last two digits of the year, in which the public key certified in the certificate (i.e. the PuK.CA.CS_AUT) is issued.

Table B.5 – Structure of Certificate Holder Reference, if the certificate holder is the card itself.

Filler (2 byte)	ICCSN.HPC (10 byte)

NOTE – The ICCSN has a length of 10 byte according to [Resoultion190]. The key reference has always the length of 12 byte.

B.1.4 Certificate Holder Authorization

The „Certificate Holder Authorization (CHA)“ has the purpose to denote the access rights of the card holder, e.g. the access rights of the card holder with respect to data stored in another card. The meaning of CHA can be compared with a role based group key when applying symmetrical algorithms.

The CHA consists of

- a prefix denoting the entity assigning the role id and
- the role identifier of the certificate holder.

NOTE: As role ID a profile identifier may be encoded.

Table B.6 – Structure of Certificate Holder Authorization

Prefix (6 byte)	Role/Profile ID (1 byte)

The prefix may consist of an AID (6 MSB) or a world wide unique identifier of the respective entity. Different groups or different entities of certificate holders shall not have the same CHA.

The subsequent tables show examples for role identifiers.

Table B.7a – CHA role/profile ID coding (certificate holder = CA)

CHA Role/Profile ID	Owner
'00'	CA (No authorization, used in higher level certificates or cross-certificates)

NOTE – '00' means: no access right to data in a target card (HPC or SMC or eHC)

Table B.7b – Examples of CHA role/profile ID coding (certificate holder = HPC/SMC)

CHA Role/Profile ID	Owner
'1A'	e.g. SMC ekiosk
'2A'	e.g. HPC Physician or HPC Dentist

NOTE – The concrete role ids used are specified in part 2 of the HPC specification.

Table B.7c – CHA role/profile ID coding (certificate holder = eGK)

CHA Role/Profile ID	Owner
'00'	eGK cardholder

NOTE – '00' means: no access right to data in a target card (HPC or SMC)

B.1.5 Object identifier for signature algorithm of the certificate holder

The Object Identifier has to be taken in compliance with [DIN V66291-1].

B.1.6 Public key of certificate holder

B.1.6.1 General construction

The Public Key in a certificate consists of a concatenation of parameters. These parameters, which have a context specific tag, belong to the DO PK (Tag '7F49', constructed) and have to be coded as an octet string. In the CV certificate verifying entity (i.e. in a HPC, eHC or SMC) the occurrence of such a parameter and its length can be described in an appropriate headerlist which is indicated by CPI.

B.1.6.2 Public key RSA

- Tag '81': Modulus
- Tag '82': Public exponent (value 65537 used)

B.1.7 Coding of the CV certificates

The following table shows the coding of the CV certificates (see [CEN-WG16]).

Table B.8 – CPI values and CV field values

CPI (1 B)	**CAR** (8 B)	**CHR** (12 B)	**CHA** (7 B)	**OID**	**PK**	**Remark**
'03'	...	...	Prefix (6 byte) || '00'	'2B240304020201' (7 byte)	Modulus (128 byte) || Exponent (4 byte)	CVC for a CA, issued by the RCA (signed key is used for verifying of further keys in a CVC chain; signed is the hash value of the CVC content)
'04'	...	...	Prefix (6 byte) || 'xx'	'2B2403050203' (6 byte)	Modulus (128 byte) || Exponent (4 byte)	CVC for an eGK, HPC or SMC, issued by a CA (signed key is used for authentication with or without TC establishment; see note and Annex E

NOTE - For distinguishing between the authentication procedure with and without TC establishment, the related private key will be used with 2 different key identifiers (a COS not able to support multi-referece key objects may store the same key twice).

Table B.9 – Object Identifier

OID Coding	OID Number	OID Name	OID Registration Authority
'2B240304020201'	{1 3 36 3 4 2 2 1}	sigS_ISO9796-2Withsha1 (= signature scheme with RSA signature and DSI according to ISO9796-2 and SHA-1	TeleTrusT
'2B2403050203'	{1 3 36 3 5 2 3}	authS_ISO9796-2Withrsa_mutual = authentication scheme with RSA signature and DSI according to ISO/IEC 9796-2 and SHA-1 for a mutual authentication with or without establishment of a Trusted Channel	TeleTrusT
TeleTrusT-OID Registration Authority: www.teletrust.de/anwend.asp?Id=30200&Sprache=D_&HomePG=0			

NOTE – In [ISO9796-2], Annex A.4 is relevant.

B.2 Structure and content of a CV certificate file

A CV certificate EF contains a constructed certificate data object with tag '7F21' (RSA certificate with message recovery), see Table B.10. The total length of the file content is in case of CPI = '03' 210 byte, in case of CPI = '04' 209 byte, see Table B.10 and B.11.

Table B.10 – Structure and content of an EF containing a CV certificate with CPI = '03'

Tag	L	Value		
'7F21'	'81CE'	CV certificate (206 byte)		
		Tag	**L**	**Value**
		'5F37'	'8180'	SIG.CA (128 byte)
				Digital Signature Input for SIG.CA ('6A' …'BC'):
				'6A' = Padding acccording to ISO 9796-2
				'03' = CPI
				'xx..xx' = CAR (8 byte)
				'xx..xx' = CHR (12 byte)
				'xx..xx' = CHA (7 byte)
				'xx..xx' = OID (7 byte)
				'xx..xx' = PK part 1 (first part of modulus, 71 byte)
				'xx..xx' = Hash (20 byte, Hash Input: DEs CPI … PK, see Table B.8)
				'BC' = Trailer
		'5F38'	'3D'	'xx..xx' = PK remainder (rest of modulus followed by exponent '00010001', 61 byte)
		'42'	'08'	'xx..xx' = CAR (8 byte)

Table B.11 – Structure and content of an EF containing a CV certificate with CPI = '04'

Tag	L	Value		
'7F21'	'81CD'	CV certificate (205 byte)		
		Tag	**L**	**Value**
		'5F37'	'8180'	SIG.CA (128 byte)
				Digital Signature Input for SIG.CA ('6A' …'BC'):
				'6A' = Padding acccording to ISO 9796-2
				'04' = CPI
				'xx..xx' = CAR (8 byte)
				'xx..xx' = CHR (12 byte)
				'xx..xx' = CHA (7 byte)
				'xx..xx' = OID (6 byte)
				'xx..xx' = PK part 1 (first part of modulus, 72 byte)
				'xx..xx' = Hash (20 byte, Hash Input: DEs CPI … PK, see Table B.8)
				'BC' = Trailer
		'5F38'	'3C'	'xx..xx' = PK remainder (rest of modulus followed by exponent '00010001', 60 byte)
		'42'	'08'	'xx..xx' = CAR (8 byte)

B.3 CVC-Handling

For CVC handling the following additional conventions apply:

- The verification of a CVC chain starts always with selection of the Root PuK (KeyReference 8 byte).
- A CVC chain consists usually of 2 CVCs. If a cross CVC is used, then 3 CVCs have to be verified.
- The card has to remember the PuK and its reference when verifiying a CVC with PuK usage "CertSign".
- The card has to remember the PuK, its reference and the CHA when verifiying a CVC with PuK usage "Authentication".
- After successful authentication a security status is set that the related CHA has been successfully presented (in the HPC not used in access rules, but for the SMC relevant).

Annex C
(normative)

Secure Messaging

NOTE – The processing of secured commands is only relevant for HPCs. The production of secured commands and the processing of secured responses are only relevant for SMCs, see Annex E.

C.1 SM-DOs

Table C.1 shows the DOs used within the scope of the health care application (these are a partial set of the SM-DOs described in [ISO/IEC 7816-4]).

Table C.1- SM Data objects

Tag	Meaning
'81'	Plain Value, if INS even (to be protected by CC)
'B3'	Plain Value, if INS odd (to be protected by CC)
'97'	Le (to be protected by CC)
'99'	Status-Info (to be protected by CC)
'8E'	Cryptographic Checksum
'87'	PI || Cryptogram, if INS even (to be protected by CC)
'85'	Cryptogram, if INS odd (to be protected by CC)

For cryptograms the padding indicator PI (see DO with tag '87') is always set to '01', i.e. padding acc. to [ISO/IEC 7816-4] ('80 ...00').

C.2 Commands and Responses with SM

After an authentication procedure is completed and a trusted channel is established, all commands and responses shall be transferred in the SM mode. If session keys are established for a certain logical channel and the card receives a command without SM for this logical channel, then the session keys for that logical channel are no longer available. Furthermore, the security status with respect to the authentication procedure with SM key transport/agreement shall no longer usable.
Since the command header should be integrated into the CC calculation, the bits b4 and b3 of the CLA byte shall be set to 1. For simplification, the examples are outlined with channel #0. Thus there is the following structure for commands and responses:

Command:

'0C'	INS	P1-P2	Lc	T_{PV}	L_{PV}	PV	T_{Le}	L_{Le}	Le	T_{CC}	Lcc	CC	Le'

The DOs PV and Le are conditional, i.e. those DOs are only present, when the command without SM contains a data field and an Le field respectively. The part of the cryptographic checksum (CC) to be transmitted is for the HPC 8 byte.

Response with data:

T_{PV}	L_{PV}	PV	T_{CC}	Lcc	CC	SW1-SW2

Response without data:

T_{SW}	'02'	SW1-SW2	T_{CC}	Lcc	CC	SW1-SW2

The DO PV is conditional, i.e. the DO PV has to be present, if the plaintext response APDU contains data.
DO SW with the status information (tag '99') has to be present, if the plaintext response APDU contains no data or an error has been detected. In other cases the DO SW may be absent.

If commands with command or response data are performed with SM, then the data in the data field shall be transferred as a cryptogram, if required by the respective security conditions. Examples of those commands are

- READ BINARY
- UPDATE BINARY
- VERIFY
- CHANGE RD
- RESET RC.

Thus there is the following structure for those commands and their responses:

Command without cryptogram (e.g. READ BINARY):

'0C'	INS	P1-P2	Lc	T_{Le}	L_{Le}	Le	T_{CC}	Lcc	CC	Le'

Response with cryptogram:

T_{CG}	L_{CG}	PI, CG	T_{CC}	Lcc	CC	SW1-SW2

Command with cryptogram (e.g. VERIFY):

'0C'	INS	P1-P2	Lc	T_{CG}	L_{CG}	PI,CG	T_{CC}	Lcc	CC	Le'

Response without cryptogram:

T_{SW}	'02'	SW1-SW2	T_{CC}	Lcc	CC	SW1-SW2

The status bytes of the command response shall be identical to the status bytes protected by CC.

C.3 Treatment of SM-Errors

When the HPC recognizes an SM error while interpreting a command, then the status bytes shall be returned without SM. In [ISO/IEC 7816-4] the following status bytes are defined to indicate SM errors:

- '6987': Expected SM data objects missing
- '6988': SM data objects incorrect

After an SM error has been detected, the session keys of the respective channel shall be erased. Furthermore, the security status with respect to the authentication procedure with SM key transport/-agreement shall no longer usable.

C.4 Padding for checksum calculation

The padding mechanism acc. to [ISO/IEC 7816-4] ('80 ...00') is applied.

C.5 DES-Mode, Initial Value and Send Sequence Counter

C.5.1 Cryptograms

Cryptograms are built with two key triple DES (DES-3) in CBC-Mode with the Null vector as initial value. The padding of the plain value shall be performed according to ISO/IEC 7816-4, see clause 6.2.3.1.

C.5.2 Cryptographic Checksums

Cryptographic checksums are built acc. to [ISO/IEC 7816-4] as follows (the basic mechanism is to build a retail MAC acc. to ANSI X9.19 with DES):

- Initial stage: The initial check block y_0 is E(Ka, SSC).
- Sequential Stage: The check blocks y_1, .. , y_n are calculated using Ka.
- Final Stage: The cryptographic checksum is calculated from the last check block y_n as follows: E(Ka, D(Kb, y_n)).

Here E() means encryption with DES, respectively D() decryption with DES.

The send sequence counter SSC must be increased (+1) each time before a MAC is calculated, i.e. if the starting value is x, in the next command the value of SSC is x+1. The SSC value of the first response is then x+2.

The starting value for the SSC is

for HPC: SSC = RND.HPC (4 least significant bytes) || RND.SMC (4 least significant bytes)
for SMC: SSC = RND.yyy (4 least significant bytes) || RND.SMC (4 least significant bytes)

with yyy = card, which has produced the second random no. (HPC or eGK).

C.6 Use of DES

The following figure shows the application of keys in DES-3 (see also [ISO 11568-2]).

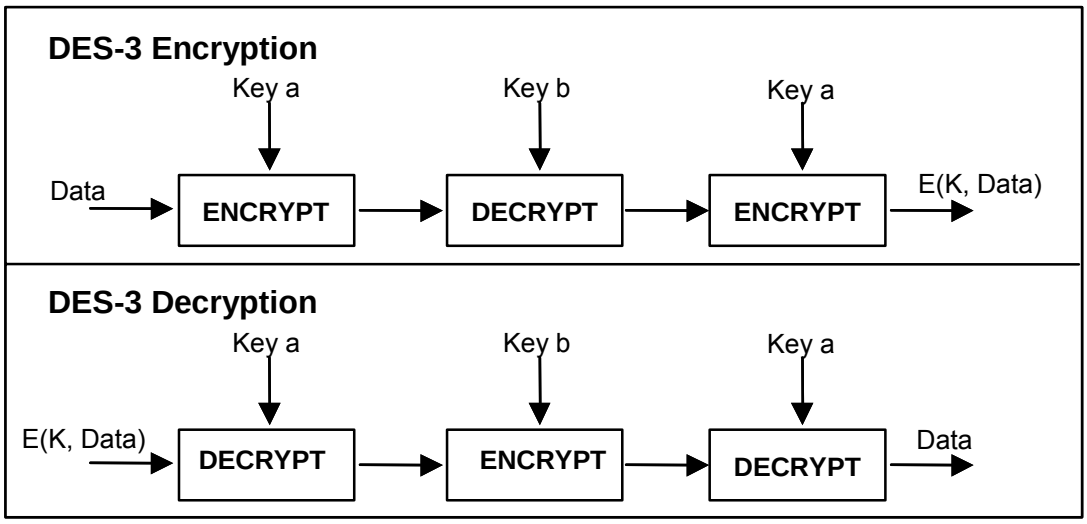

Figure C.3 – DES-3-Encryption/Decryption

The retail MAC is calculated as depicted in figure C.4.

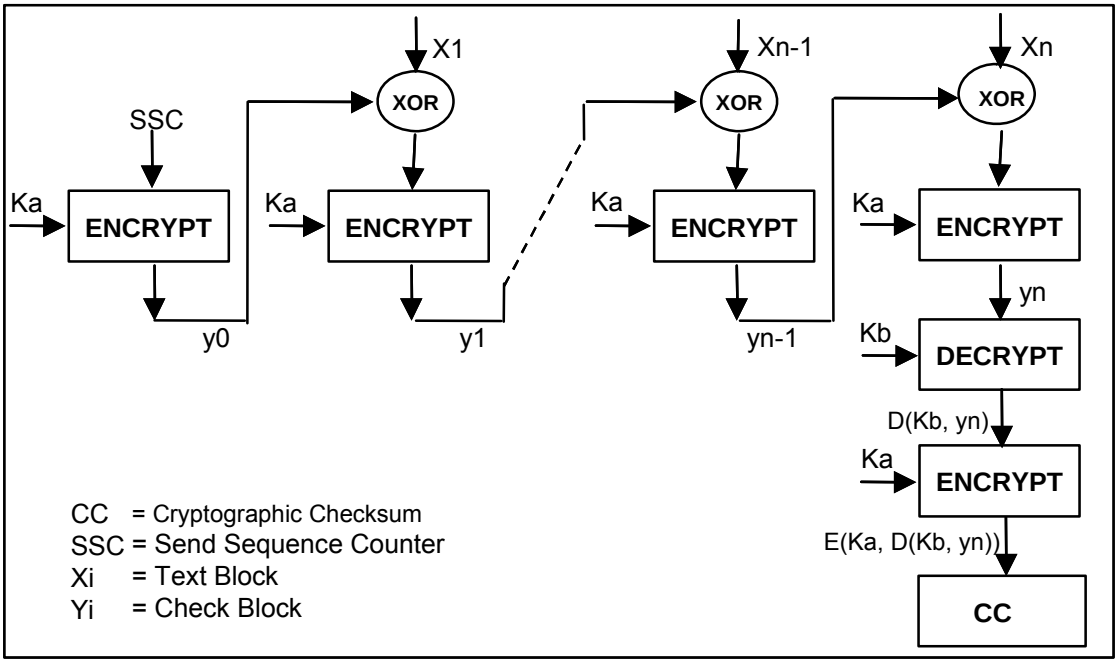

Figure C.4 – Calculation of the retail MAC

C.7 SM Key Referencing

After TC establisment, the SM keys are implicitly selected, i.e. a MSE command is not necessary.

Annex D
(normative)
Production of secured Commands and Processing of secured Responses

NOTE – This annex is only relevant for SMCs.

D.1 General

For the production of a secured HPC command and the processing of a secured HPC response, the SMC shall support either

- PSO commands (COMPUTE CC, VERIFY CC, ENCIPHER, DECIPHER) or
- the ENVELOPE command or
- both concepts.

D.2 PSO Method

It is assumed, that an authentication procedure between the HPC and a security module card SMC in the IFD has been successfully completed. The authentication procedure is based e.g. on card verifiable certificates and has an inherit key transport or key agreement mechanism so that after this procedure

- a symmetric SM key for the computation of cryptographic checksums SK.CC,
- a symmetric SM key for the computation of cryptograms SK.CG and
- a send sequence counter SSC with its initial value is available in the HPC and the SMC.

All commands to the HPC are sent in SM mode with bit b4 = 1 and b3 = 1 in the CLA byte, i.e. the command header will be integrated in the CC computation.
All commands sent to the SMC are PSO related commands, not in SM mode but using SM-DOs and the secure messaging keys set with an MSE command. The send sequence counter SSC is incremented each time before usage.

The general construction principle for secured command production and secured response processing is shown in Figure D.1 and D.2.

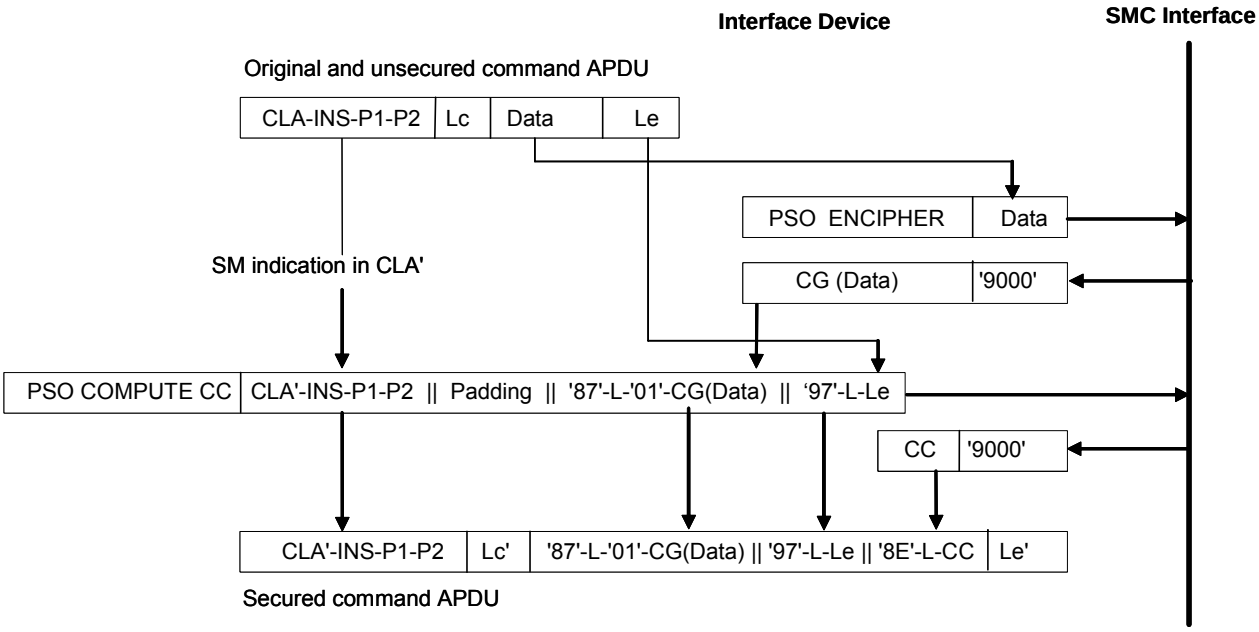

Figure D.1 – Example of secured command production

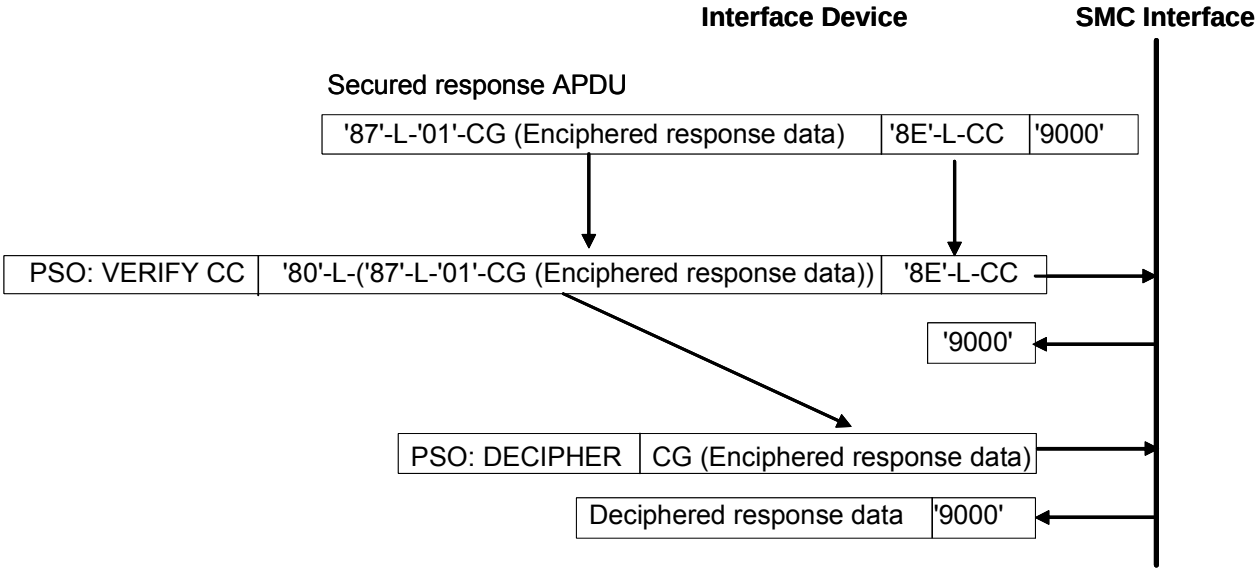

Figure D.2 – Example of secured response processing

The subsequent example presents the computation of a digital signature (DS) whereby the usage of the signature key requires the successful presentation of reference data (password).

Notation:
CLA' = Header with SM indication (b4 = 1, b3 = 1)

Step 1: Verification of reference data (password)

Command to SMC: MSE SET <CT, ’83’-'01'-’F1’> -- In this example SK.CG has the key reference 'F1'
SMC response: OK

Command to SMC: MSE SET <CCT, ’83’-'01'-’F2’> -- In this example SK.CC has the key reference 'F2'
SMC response: OK

Command to SMC: PSO ENCIPHER <RD>
SMC response: <CG(RD)>

Command to SMC: PSO COMPUTE CC <CLA'-INS-P1-P2 - Padding - '87'-L-PI-CG(RD) - '97'-'01'-Le>
SMC response: <CC>

Now the IFD is able to construct the secured VERIFY command.

Command to HPC: VERIFY <'87'-L-'01'-CG(RD) - '97'-'01'-Le - '8E'-L-CC>
HPC response : <'99'-'02'-SW - '8E'-L-CC>

Command to SMC: PSO VERIFY CC <'80'-L-('99'-'02'-SW) - '8E'-L-CC>
SMC response: OK

Step 2: Computation of a hash value

Command to SMC: PSO COMPUTE CC <CLA'-INS-P1-P2 - Padding - '81'-L-('90'-L-Intermediate Hash - '80'-L-Last block) - '97'-'01'-Le - Padding>
SMC response: <CC>

Command to HPC: PSO HASH <'81'-L-('90'-L- Intermediate Hash - '80'-L-Last block)> - '8E'-L-CC>
HPC response : < '99'-'02'-SW - '8E'-L-CC>

Command to SMC: PSO VERIFY CC <'80'-L-('99'-'02'-SW) - '8E'-L-CC>
SMC response: OK

Step 3: Computation of a digital signature

Command to SMC: PSO COMPUTE CC <CLA'-INS-P1-P2 - Padding - '97'-'01'-'00'>
SMC response: <CC>

Command to HPC: PSO: COMPUTE DS <'97'-'01'-'00' - '8E'-L-CC >
HPC response : <'81'-L-DS - '8E'-L-CC>

Command to SMC: PSO VERIFY CC <'80'-L('81'-L-DS) - '8E'-L-CC>
SMC response: OK

D.3 ENVELOPE Method

D.3.1 Production of a secured command

For production of a secured command to be sent to a HPC or an eGK, an ENVELOPE command with odd instruction code is send in normal mode to the SMC. In the data field, the unsecured command in the DO Command-to-perform (tag '52') and an SM template (tag '7D') is present containg the Response Descriptor (tag 'BA') which denotes, what shall be returned: the SM data object CC and – if

needed – also the SM data object CG, encapsulated in an SM Template. The SM template enforces the usage of the SM keys.

Table D.1 – ENVELOPE command

CLA	As defined in ISO/IEC 7816-4
INS	‘C3’ = ENVELOPE
P1-P2	‘0000’
Lc	‘xx’ = Length of subsequent data field
Data field	- If the command to be secured contains data to be transmitted in DO PV (Case 1): '52'-L-command to be secured || '7D'-L-('BA' -L- ['8E'-'00']) - If the command to be secured contains data to be transmitted in a DO CG (Case 2): '52'-L-command to be secured || '7D'-L-('BA' -L- ['87'-'00' || '8E'-'00'])
Le	'00'

Table D.2 – ENVELOPE response

Data field	- Case 1: '7D'-L-('8E' -'0x'-CC) - Case 2: '7D'-L-('87' -L- '01'-CG || '8E' -'0x'-CC)
SW1-SW2	‘9000’or specific status bytes

For the CC computation, the value field of the DO ‘Command-to-perform’ (tag ‘52’) shall be taken applying the rules for CC computation for SM-protected commands as defined in [ISO7816-4], whereby the command header shall be always integrated in the CC.

D.2.2 Processing of secured responses using the ENVELOPE command

The HPC or eHC will return secured responses, whereby 3 cases occur:

- response with DO Processing status (tag '99')
- response with DO Plain value (tag '81')
- response with DO Cryptogram (tag '87').

All secured responses are protected by a cryptographic checksum CC.

NOTE – The cases addressed here are application cases and should not be mixed up with transmission cases as described in ISO/IEC 7816-3.

The CC must be verified. If a cryptogram is present, then the plain value has to be returned by the SMC after successful verification of the CC.

Table D.3 – ENVELOPE command

CLA	As defined in ISO/IEC 7816-4
INS	‘C3’ = ENVELOPE
P1-P2	‘0000’
Lc	‘xx’ = Length of subsequent data field
Data field	- Case 1: '7D'-L-('99'-'02'- SW1-SW2 || '8E'-'08'-CC) - Case 2: '7D'-L-('81'-L- Data || '8E'-'0x'-CC) - Case 3: '7D'-L-('87' –L- '01'-CG || '8E'-'0x'-CC || 'BA' –L- ['80'-'00'])
Le	Case 1, 2: Absent Case 3: '00'

Table D.4 – ENVELOPE response

Data field	- Case 1: Absent - Case 2: Absent - Case 3: '7D'- L-('80'-L-Data)
SW1-SW2	‘9000’or specific status bytes

Annex E
(normative)
Authentication Procedures

E1. Notation

Command/Resp. data	< >
Concatenation	||
h(x)	Hash calculation with SHA1
ENC [key, data]	Cryptogram
MAC [key, data]	Message Authentication Code
KD.xxx	Key derivation Data, choosable by xxx
PRND	Pseudo Random Number
RND	Random number (8 byte)
SIG	Signature
DS [key, DSI]	Signature calculation according to OID in CVC OID
DS-1	Signature verification according to OID in CVC OID
ICCSN8.xxx	8 Least significant Byte of ICCSN of card xxx

E.2 Asymmetric Authentication Procedure without SM Key Agreement

Figure E.1 shows the asymmetric authentication procedure without SM key agreement between an eGK and a HPC. The same procedure is also relevant for eGK and SMC interactions.
Before performing the authentication procedure in the HPC the private key PrK.HPC.AUT has to be selected with the key reference where the purpose of the key belonging to this authentication procedure. SHA-1 is the hash algorithm to be used in conjunction with the RSA algorithm (key length at present: 1024 bit). The public key of the eGK PuK.eGK.AUT has to be imported with the CVC mechanism prior to the authentication procedure.

The authentication procedure is performed by two separated command sequences, i.e. in a first step the eGK is authenticated and in a second step the HPC has to be authenticated by the eGK to prove its access rights.

A random number RND.HPC respectively RND.eGK is to be fetched immediately before the command in which it is used, i.e. between GET CHALLENGE and EXTERNAL AUTHENTICATE no other command is allowed.

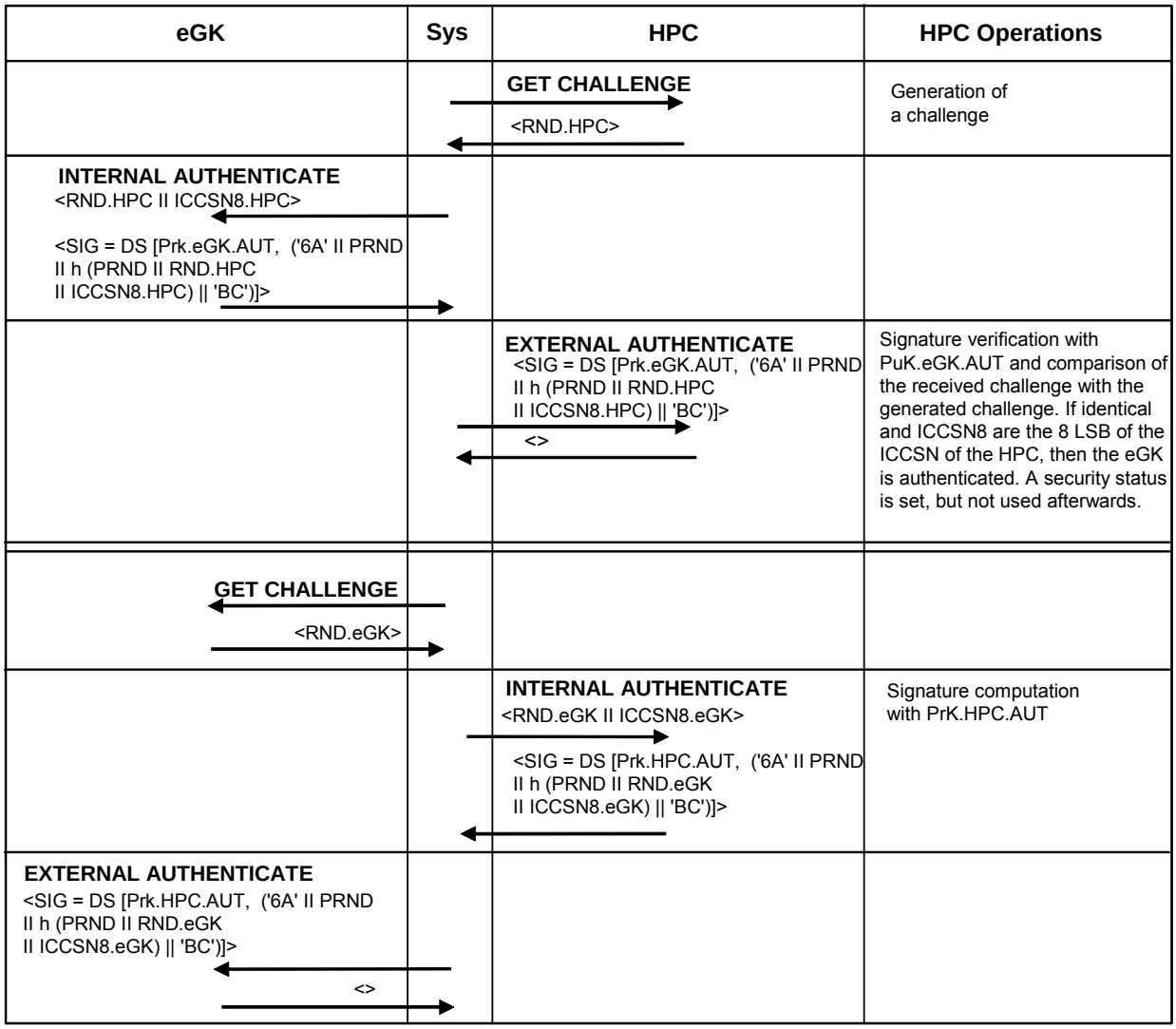

Figure E.1 - Asymmetric Authentication Procedure without SM Key Agreement

NOTE - The octet string '6A … BC' has always the same length in byte as the modulus of the RSA key.

E.3 Asymmetric Authentication Procedure with SM Key Agreement

Figure 2 shows the asymmetric authentication procedure with SM key agreement between an eGK and a SMC.
Before performing the authentication procedure in the SMC the private key PrK.SMC.AUT has to be selected with the key reference where the purpose of the key belonging to this authentication procedure. SHA-1 is the hash algorithm to be used in conjunction with the RSA algorithm (key length at present: 1024 bit). The public key of the eGK PuK.eGK.AUT has to be imported with the CVC mechanism prior to the authentication procedure.

The command sequence shown in Figure E.2 shall not be interrupted by any other command.

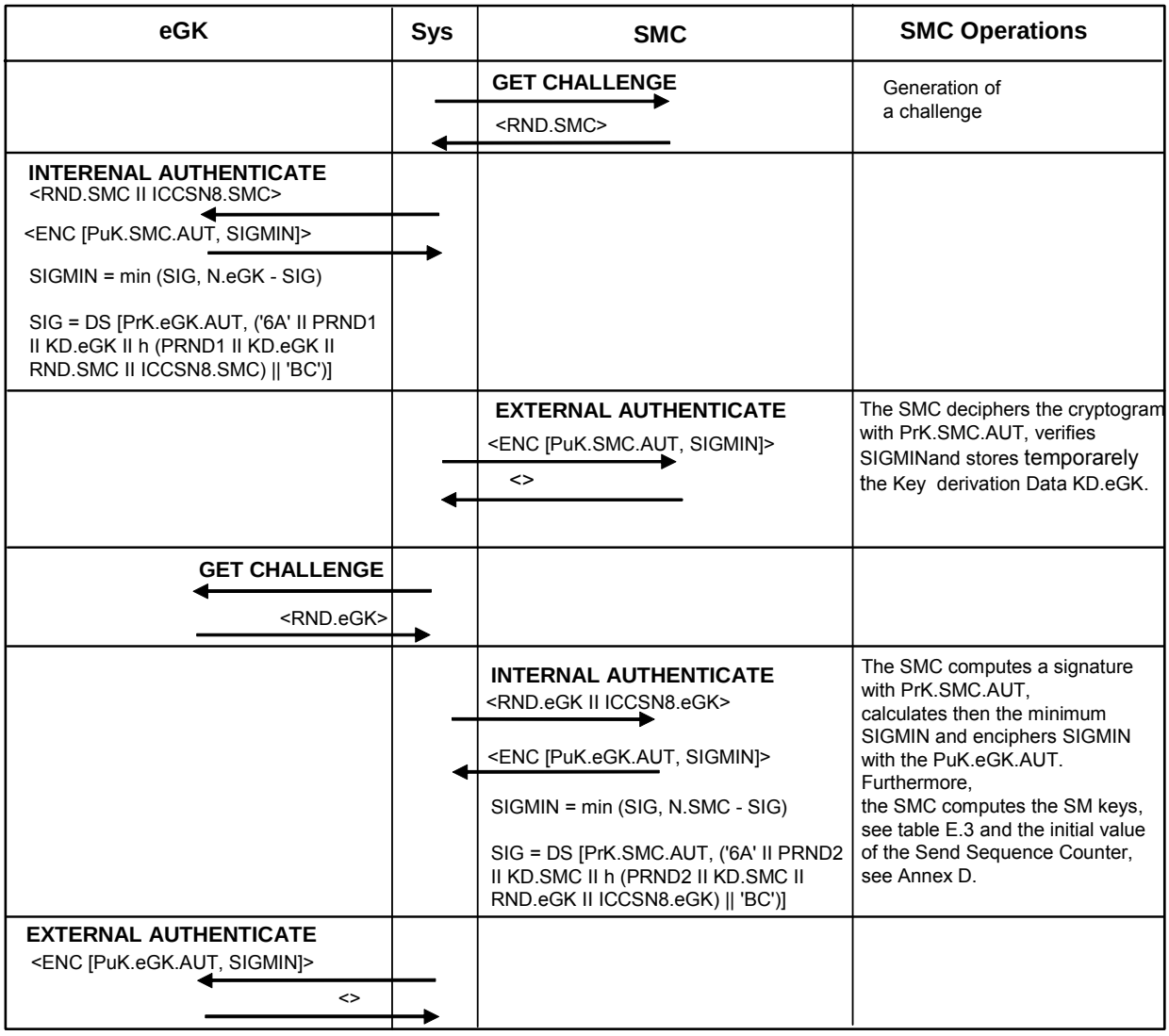

Figure E.2 - Asymmetric Authentication Procedure with SM Key Agreement

NOTE - The octet string '6A … BC' has always the same length in byte as the modulus of the RSA key.

The authentication procedures to be performed by HPC, SMC and eGK are executed always in the order shown in Table E.1.

Table E.1 - Authentication Sequences

Card	Authentication sequence as seen at the interface of the card mentioned in column 1
HPC	Partner: eGK or SMC 1. EXT. AUTHENTICATE 2. INT. AUTHENTICATE
eGK	Partner: HPC or SMC 1. INT. AUTHENTICATE 2. EXT. AUTHENTICATE
SMC	Partner: eGK 1. EXT. AUTHENTICATE 2. INT. AUTHENTICATE Partner: HPC 1. INT. AUTHENTICATE 2. EXT. AUTHENTICATE

The SM Keys are computed as shown in Figure E.3.

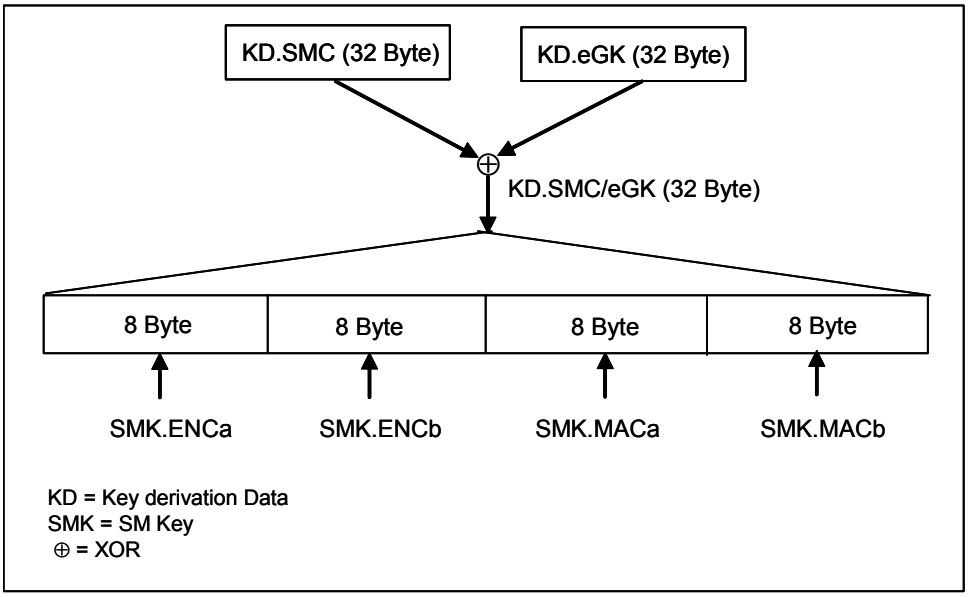

Figure E.3 – SM Key Agreement according to [DIN 66291-4] and [CWA 14890-1] for asymmetric Authentication Procedures

E.4 Symmetric Authentication Procedure with SM Key Agreement

Figure 2 shows the symmetric authentication procedure with SM key agreement between an HPC and a server as described in [CWA 14890-1].
The server uses

- a MasterKey MK, from which all individual SK.HPC are derived applying the ICCSN of the HPC or
- n Group Keys GK or
- m IndividualKeys IK, whereby m is the number of issued HPCs (i.e. no key derivation is needed, only key selection).

Before performing the authentication procedure in the HPC the symmetric key SK.HPC.AUT has to be selected. Since only 3DES is used, no algorithm reference is needed.

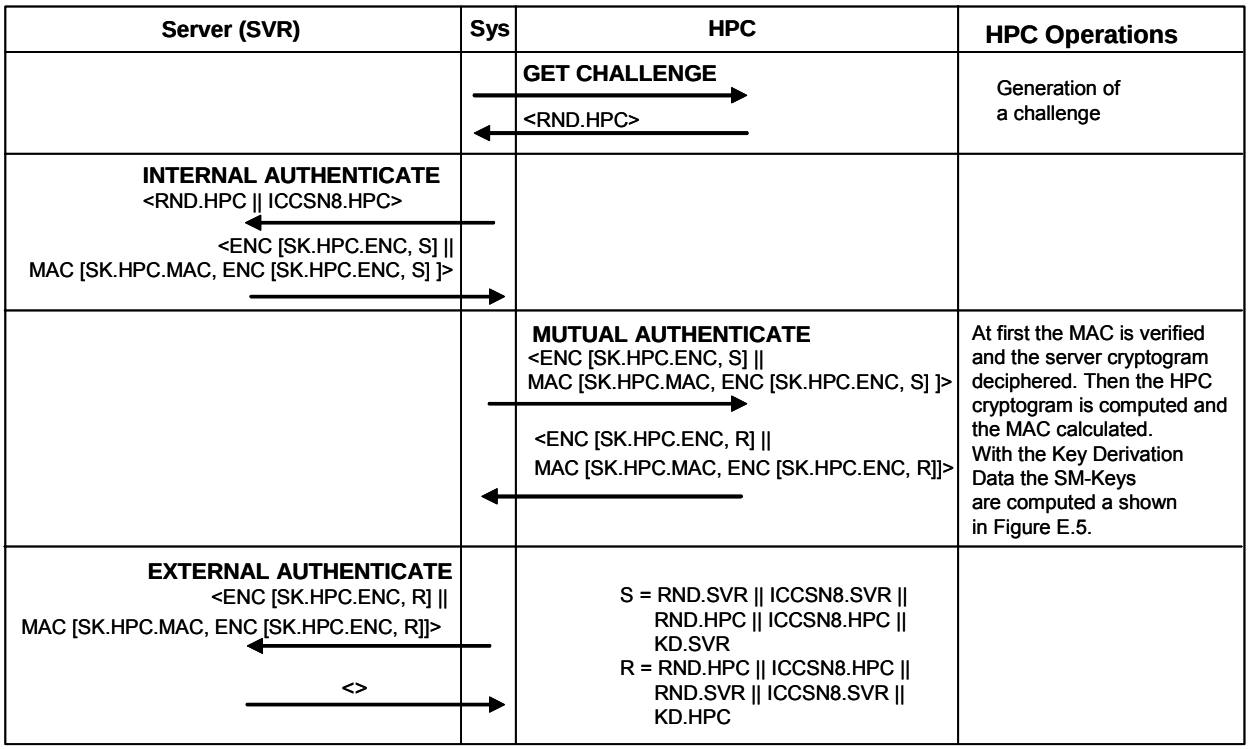

Figure E.4 - Symmetric Authentication Procedure with SM Key Agreement

Rules for ENC and MAC computation for MUTUAL AUTHENTICATE:

- For the computation of the ENC part in the command/response data field, the IV for CBC 3DES encryption is zero. No padding is applied.
- For the computation of the MAC part in the command/response data field, the initial check block Yo is zero. Padding is mandatory based on [ISO7816-4] ('80...').

The SM Keys are computed as shown in Figure E.5.

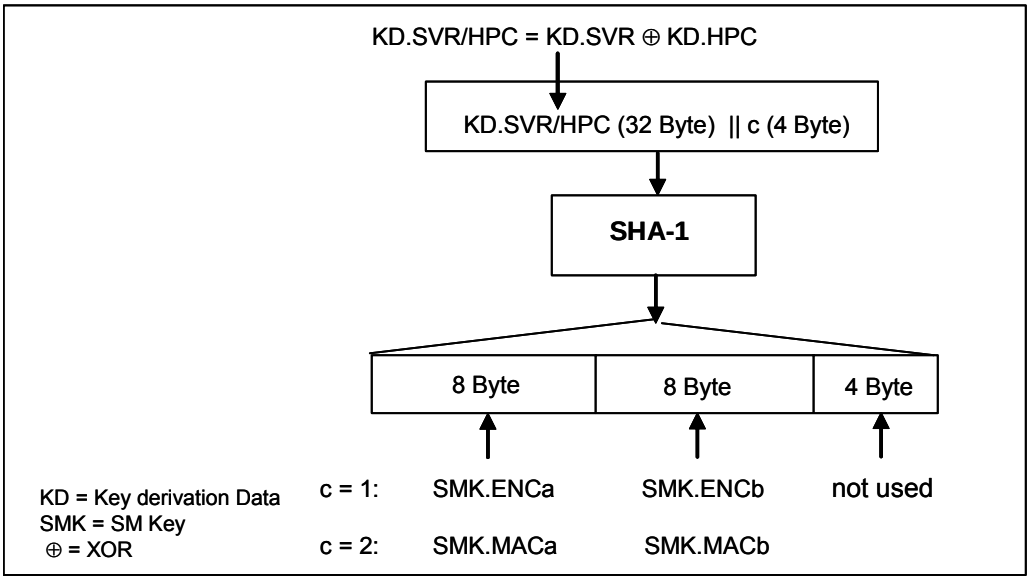

Figure E.5 – SM Key Derivation according to ANSI X9.63 for symmetric Authentication Procedures

E.5 Challenge/Response Procedure with a symmetric Key

The authentication procedure consists of enciphering and deciphering of a challenge (8 byte, no padding) with a 3DES key in ECB mode, see Figure E.6. The algorithm reference is implicit.

HPC	Sys	Server	HPC Operations
GET CHALLENGE <RND.HPC>			Generation of a challenge
	INTERNAL AUTHENTICATE <RND.HPC> <ENC [SK.HPC.ENC, RND.HPC]>		
EXTERNAL AUTHENTICATE <ENC [SK.HPC.AUT, RND.HPC]> <>			The HPC deciphers the cryptogram with the key SK.HPC.ENC and compares the received challenge with the generated challenge. If identical, the outside entity is authenticated and a respective security status is set in the HPC.

Figure E.6 - Challenge/Response Procedure with a symmetric Key

E.6 Client/Server Authentication with X.509 Certificate

When performing a Client/Server Authentication with X.509 Certifcate, the HPC computes a signature using the Private Key PrK.HP.AUT of the cardholders.Padding is applied according to PKCS#1 V1.5 and in the command data field the digest info or similar authentication data are delivered to the card, see Table E.2. The algorithm reference is implicit.

Table E.2 - Digital Signature Input for Client/Server-Authentication

DSI	Input-Elements
'00' || '01' || PS || '00' || T	PS = Padding String of octets with 'FF' T = Data in the data field of the INTERNAL AUTHENTICATE command (the length of the data field shall not exceed 40% of the length of the modulus of the signature key, see [DIN66291-4]). The formatited octet string has k Octets, whereby k is the length in octets of the modulus of the signature key.

E.7 Optional Features

E.7.1 Storing SM Keys

When performing a CVC based authentication procedure between HPC and SMC, the SM keys may be stored persistent in a COS dependant way under the key reference SMK.SMC (ICCSN8.SMC). These keys are intended to be used for SM key negotiation for saving time (symmetric authentication is faster than CVC based authentication procedures). The authentication procedure to be applied is the one described in clause E.4, but instead of the server the SMC is involved.

E.7.2 Storing Public Keys of CAs

As part of the functionality of the PSO: VERIFY CERTIFCATE command, public keys of a CA may be persistently stored in the HPC. This feature allows e.g. the import of a new RCA key, if the RCA uses a new key for CVC signing.

.

Annex F
(informative)

Reset of a Key Usage Counter

NOTE – The functionality is not yet required for the first issuing of the HPC. However, it may become mandatory, if problems occur with the unlimited usability of the private key for C2C authentications. It is intended to submit a new work item proposal to ISO addressing the issue of resetting a key usage counter.

In this annex an example is presented, how to reset a usage counter of a key using a functional extended version of the RESET RETRY COUNTER (the outcome of an ISO standardization process addressing this issue is not foreseeable):

- In case the command is related to PIN or biometric reference data, the retry counter is addressed
- In case the command is related to a key, the usage counter is addressed.

This command is needed, if e.g. the capability of performing C2C-authentication procedures shall be bound to a resettable usage counter to avoid unlimited usage of the C2C authentication mechanism.

Table F.1 – RESET RETRY COUNTER command-response pair

CLA INS P1 P2	As defined in 5.1.1 of ISO/IEC 7816-4 '2C' = RESET RETRY COUNTER see table F.2 - If P1 = '00' – '03': see Table 65 of [ISO7816-4] - If P1 = '8x': see Table F.3
Lc field	Absent for encoding Nc = 0, present for encoding Nc > 0
Data field	- If P1 = '00' – '03': see Table 76 of [ISO7816-4] - If P1 = '8x': Absent
Le field	Absent for encoding Ne = 0

NOTE – The key of which the key usage counter is to be reset may be selected with a separate MSE command.

Data field	Absent
SW1-SW2	See Table A.1 and specific status codes in Table A.16 In addition: '6984' = Reference data not usable

Table F.2 – Coding of P1

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	-	-	-	-	-	-	-	**Indication of attribute list in bits b4 – b1**
0 1	-	-	-	-	-	-	-	- Usage according to Table 76 of [ISO7816-4] - Usage according to this table
-	**x**	**x**	**x**	**x**	**x**	**-**	**-**	**RFU (default 0)**
-	**-**	**-**	**-**	**-**	**-**	**x**	**x**	**Key selection**
-	-	-	-	-	-	0	0	No information given, key implicitly known
-	-	-	-	-	-	0	1	P2 shall be interpreted as shown in table F.3
-	-	-	-	-	-	1	0	P2 contains a key reference according to table 65 in [ISO7816-4]
-	-	-	-	-	-	1	1	RFU

Table F.3 – Coding of key referencing with usage and CRT indication

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	**x**	**x**	**x**	**-**	**-**	**-**	**-**	**Coding of relvant service**
0	0	0	0	-	-	-	-	Verification (DST, CCT), Encipherment (CT), External authenticate (AT), Key agreement (KAT)
0	1	1	1	-	-	-	-	Computation (DST, CCT), Decipherment (CT), Internal authenticate (AT)
0	1	1	0	-	-	-	-	SM in response data fields (CCT, CT, DST)
0	1	0	1	-	-	-	-	SM in command data fields (CCT, CT, DST)
-	**-**	**-**	**-**	**x**	**x**	**x**	**x**	**Coding of relevant CRT**
-	-	-	-	0	0	0	0	AT
-	-	-	-	0	0	0	1	CCT
-	-	-	-	0	0	1	0	DST
-	-	-	-	0	0	1	1	CT
-	-	-	-	0	1	0	0	KAT
- Any other value is reserved for future use by ISO/IEC JTC 1/SC 17								