Functional Specification
# of the
OpenPGP application
on
ISO Smart Card Operating Systems

Version 2.1.1

Author: Achim Pietig

December 10th

Author:

Achim Pietig

Lippstädter Weg 14
32756 Detmold
Germany
Email: openpgp@pietig.com

Many thanks to all people for advice, error correction and hints to this specification.

The author does not assume responsibility nor give a guarantee for the correctness and/or completeness of the features and functions described in this document.

The author is unable to accept any legal responsibility or liability for incorrect and/or incomplete details and their consequences.

Furthermore, the author reserves the right to revise these specifications for technical reasons and make amendments and/or updates to the same.

History

V1.0 to V1.1:

- Change of access rights for command GENERATE ASYMMETRIC KEY PAIR and P1=81 (reading of public key) to always.
- Adjustment of the literature.
- New Data Objects for private use, with different access conditions. This optional feature is announced in Extended capabilities.
- New Data Objects for key generation date/time.
- Data Object 'PW Status Bytes' (C4) mandatory for GET DATA as single object.

V1.1 to V2.0:

- Correction of AID description (RID of FSFE).
- Adjustment of the literature.
- Alignment with changes and innovations in ISO 7816 and EN 14890.
- Change in VERIFY command and password management.
- Enhancement of Extended capabilities.
- Enhancement of Algorithm attributes.
- Reset functionality (life cycle management).
- Definition of key import according to ISO 7816-8.
- Additional Hash algorithms and different behaviour of PSO:CDS command.
- Improvements for cards without MF.
- New Data Objects:
    - Cardholder certificate (ISO 7816-6)
    - Extended header list for key import, supporting
        - Cardholder private key template (ISO 7816-6/-8)
        - Cardholder private key (ISO 7816-6/-8)
    - Historical bytes (ISO 7816-4/-6)
    - Algorithm attributes for PUT DATA
    - Resetting Code for PUT DATA
    - SM keys for PUT DATA
- Deletion of DOs 'FF', 'E0'-'E2'.
- Support for Secure Messaging.
- Support for Command Chaining.
- Support for different algorithm and key length (see PUT DATA for Algorithm attributes).
- Introduction of a Resetting Code for RESET RETRY COUNTER.
- Simplification in password management.
- Several editorial clarifications.

V2.01 to 2.1:

- Support for AES algorithm for PSO:DEC.
- New Data Objects:
    - AES-Key for PSO:DEC
- Change access conditions for TERMINATE.
- Update list of Status Bytes
- INS for TERMINATE and ACTIVATE corrected

Version 3.0 with major improvements, conformance with latest standards and ELC support will follow soon...

TABLE OF CONTENTS

1 Introduction

This functional specification describes the OpenPGP application based on the functionality of ISO smart card operating systems. In principle it defines the interface of the application between card and terminal, in this context the OpenPGP software with a standard card reader on PC/SC basis.

The solution takes care of:

- use of international standards,
- avoiding of patents,
- free usage under GNU General Public License,
- independence from specific smart card operating systems (second source),
- easy enhancement for future functionality,
- international use.

Consequently this specification does not deal with the description of the global commands and data fields of the card, the security functions generally provided by the card, any features that apply to more than one application, such as transmission protocols, nor with the description of the general mechanical and electrical characteristics of the card.

In particular, the specification provides a detailed description of the data objects directly related to the applications and their respective content formats. Contents of the application data are only prescribed if they represent a constant factor of the application.

Besides the definitions in this specification a card may support any other protocols, commands, data and variants. However, the OpenPGP application (e. g. GnuPG) should use only the defined features in this specification to be compatible to different implementations.

The encoding values mentioned in the specification are stated in hexadecimal form, unless otherwise indicated.

1.1 Definition of Abbreviations

AC	Access Condition
AID	Application IDentifier
ATR	Answer To Reset
AUT	AUThentication
BCD	Binary Coded Decimal
CLA	CLAss byte
CRT	Control Reference Template
DEC	DECipher
dec.	Decimal
DF	Dedicated File
DIR	DIRectory
DO	Data Object
DSI	Digital Signature Input
ECDSA	Elliptic Curve Digital Signature Algorithm
EF	Elementary Dile
FCI	File Control Information
FCP	File Control Parameter
FID	File IDentifier
INS	INStruction byte
LCS	Life Cycle Status
MF	Master File
OS	Operating System
PK	Public Key
PW	PassWord
RC	Resetting Code
RFU	Reserved for Future Use
RSA	Rivest-Shamir-Adleman
SE	Security Environment
SIG	SIGnature
SK	Secret Key
SM	Secure Messaging
SSC	Send Sequence Counter
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format 8 (compatible with 7-bit US-ASCII for all characters < 80)

2 General Requirements

The OpenPGP application is designed to run under several ISO-compatible card operating systems. The application can be developed on various chips and from different manufacturers. For all implementations, the following requirements should be fulfilled.

Card ->

- The OpenPGP application does evaluate Historical bytes for ‘Card capabilities’. For that reason the card shall provide a 'DO Historical bytes'.
- As single transmission protocol any ISO protocol is allowed.
    - T=1 is preferred for cards with contacts (chaining support, extended length).
- The card may support different transmission protocols.
- High speed modes are requested (maximum as possible for the chip).
- Extended length (Lc and Le) fields are recommended.
    - The card shall announce this feature in ‘Card capabilities’.
    - If Extended length is not supported, the card shall support command chaining and/or GET RESPONSE for large command/response data.

Reader (informative) ->

- A common driver (CCID, PC/SC or CT-API) shall be supported.
    - The driver should be available for several platforms (e.g. Win32, Linux, Macintosh)
- T=1 and T=0 shall be supported for cards with contacts
- High-Speed protocols should be supported.
- Extended length shall be supported.

2.1 Limitations in This Version

This version of the OpenPGP application has some restrictions in the terminal application and also in the card. The main reason is that actual cards and card readers do not support all requirements.

Terminal application:

- ECDSA and DSA are not supported (only RSA algorithm is used for all functions)

Card:

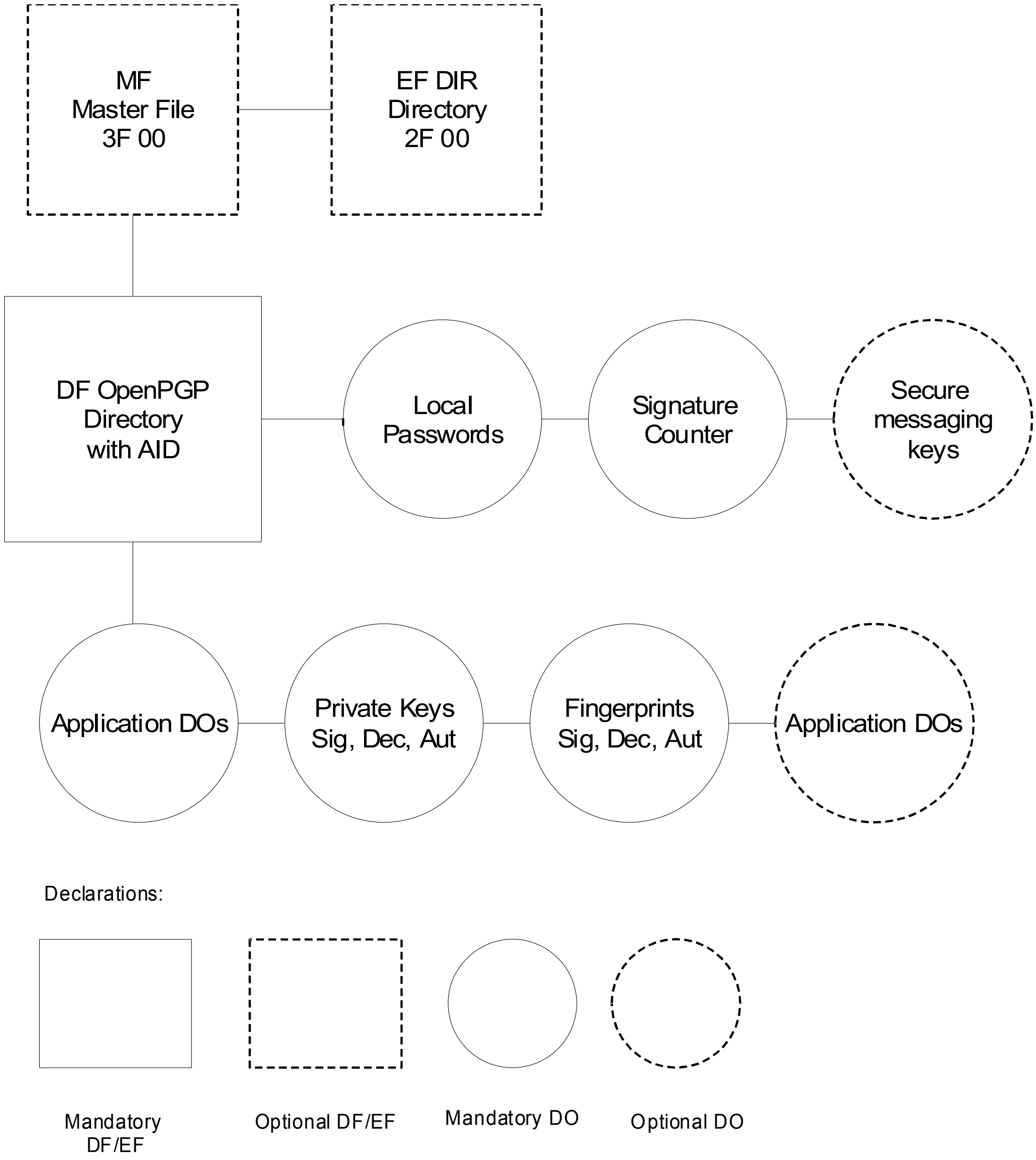
- High-Speed protocol may not be supported (terminal assumes standard values of ISO in that case)
- RSA is the only defined algorithm (minimum 1024 bit). EN 14890 recommends ECDSA as the preferred algorithm for signature cards in the future. RSA should be used as a fall-back algorithm. In a couple of years most smart cards will be available for ECDSA and it is highly recommended to add this algorithm to the OpenPGP specification (RFC 4880).

Card reader:

- High-Speed protocol may not be supported (terminal assumes standard values of ISO in that case).

3 Directory Structure

The following diagram gives an overview of the directory and data objects which are relevant to the OpenPGP application. Security related data (e.g. keys, passwords) are stored in accordance to the used OS (files, data objects or other).



4 Directory and Data Objects of the OpenPGP Application

The DF_OpenPGP directory and the data objects contained therein constitute the OpenPGP application. On the card several other applications may exist in specific Dedicated Files (DF). Because the application supports cards without Master File (MF), no data outside the OpenPGP application are addressed.

4.1 Data Files and Objects in the MF or Other DFs

The OpenPGP application uses its own set of data, including keys and passwords. No files/data in the MF or other DFs are needed for the application. However the operating system may store common data, like passwords, sharable in the card and use them for several applications.

4.1.1 EF_DIR

This optional file under the MF (file identifier: ’2F00’) may contain one or several application templates and/or application identifiers as defined in ISO/IEC 7816-4. The data file is not requested and evaluated by the OpenPGP application, but may be used to declare the application to 3rd parties. The following entries should be added in that case:

- Application Identifier (tag ‘4F’), only the significant values should be used (6 bytes = D27600012401)
- Application label (tag ‘50’), the application label should contain the following UTF-8 encoded text: OpenPGP

Example:
An entry in EF_DIR is an application template (Tag 61) in most cases. The template is stored in a record or is appended to a previous template in case of a binary structure of the EF_DIR. An entry has the content:

61 11 4F 06 D27600012401 50 07 'OpenPGP'

4.1.2 DF_OpenPGP

The directory of the OpenPGP application is stored anywhere in the card. It has no fixed File Identifier (FID), so it is easy to integrate the application in any existing context. The FID (if needed) can be chosen by the card manufacturer or any other party. In the directory all data objects of the application are accessable. The OpenPGP application can be selected with a SELECT FILE command directly after a Reset of a card. The SELECT FILE command may return FCPs (File Control Parameter), but the OpenPGP application in the terminal will not evaluate them.

4.1.2.1 Application Identifier (AID)

The OpenPGP application is selectable by a unique application identifier (see SELECT FILE). The AID has a length of 16 bytes (dec.) and is coded in the following way. The AID is unique for each card and it is recommended to integrate this value in certificates, e.g. for client/server authentication. The RID in the AID is registered by FSF Europe e.V.

	RID	**PIX**				
Coding	D2 76 00 01 24	01	xx xx	xx xx	xx xx xx xx	00 00
Length (dec.)	5	1	2	2	4	2
Name	RID of FSFE	Application	Version	Manufacturer	Serial number	RFU

RID	Registered application provider identifier (unique identification of FSFE), ISO 7816-5
PIX	Proprietary application identifier extension (defined for OpenPGP application)
Application	Indication of the application (OpenPGP)
Version	Version number of the application
Manufacturer	Unique code for the manufacturer of the application (card)
Serial number	Unique serial number
RFU	Reserved for Future Use

Application

This value (1 byte binary) specifies the application. With this definition it is possible to design different applications under control of FSF Europe e.V. in the future. The following values are defined:

00	Reserved
1	OpenPGP application (standard)
2	SmartChess
...	
FF	Reserved

Version

The version number (2 bytes, BCD) gives information about the current status of the application. With this value it is possible to announce updates to the outside world. The version number is defined as follows:

Byte 1	Byte 2	
Main version	Secondary version	(values from 00 – 99)

Example: A version

1.0	is coded	01 00
2.1		02 01
11.7		11 07

Manufacturer

To identify a card in open networks (e.g. key servers) and for the purpose of Log-In in local or open networks or to a single computer, it is necessary to have unique application numbers (related to a specific card). For that reason, every card manufacturer or personaliser - who makes the card/application ready to run - has a unique address. This manufacturer identification is controlled by FSF Europe e.V. and given to every interested manufacturer for free. Only registered manufactures are allowed to produce applications compatible with an OpenPGP application. The system works similar to MAC addresses on network cards. The 2 bytes are coded binary and the values 0000 and FFFF are reserved for test purposes.

Serial Number

Each OpenPGP application on a card from a manufacturer/personaliser has a unique serial number. The manufacturer/personaliser is responsible that no cards with duplicate numbers will occur in the outside world (like MAC addresses in networks). The number is 4 byte long (binary) and has the format MSB .. .. LSB (Most Significant Bit ... Least Significant Bit). It should start with 00 00 00 01 for the first card with an OpenPGP application of a manufacturer and normally is incremented automatically by him. However gaps in the range of numbers are allowed.

4.2 User Verification in the OpenPGP Application

The OpenPGP application uses two local passwords for user verification, called PW (PW1 with 6 characters minimum and PW3 with 8 characters minimum). PW1 is also called user-password and PW3 admin-password. PW1 (user) is the password used for signing and decryption operations, PW3 (admin) is the security officer's password. PW1 is needed for everyday use of the card, while PW3 is used to manage the card.

The format of the PWs is UTF-8 (case sensitive), the maximum length supported by the card for each PW is declared in the ´PW maximum length´ DO. Only the relevant bytes are used in the PW commands, no fillers or paddings are added. The storage of the PWs is dependent on the card OS, global PWs (used by other applications as well) may be used but mapped to the application as local. PW1 is used as access condition for the command PSO:CDS, PSO:DEC, INT-AUT, GET DATA and PUT DATA. The command PSO:CDS uses PW1 in a different mode then the other commands. The OpenPGP application uses PW3 as access condition for the commands RESET RETRY COUNTER, PUT DATA and GENERATE ASYMMETRIC KEY PAIR. All PWs use an error counter with an initial value of 3. This error counter is readable with GET DATA. After correctly verifying the PW, the access status of the corresponding PW remains valid up to a RESET of the card, a SELECT FILE to a different DF or an internal resetting by specific commands.

If the card is delivered without personalisation and/or PW letter, then a default content is assumed: PW1 = “123456” (6 bytes, 313233343536); PW3 = “12345678” (8 bytes, 3132333435363738). It is highly recommended that the card holder changes these values.

4.2.1 Resetting Code

If, for example, the card is issued by an authority or company, the users will get a complete personalised card with keys and password. The user should be able to work with the card, but is not permitted to change card data like keys and DOs under control of the issuer. He must know his user-password (PW1), but is not aware of the admin-password (PW3). To reset PW1 in the case of a blocked error counter, a special Resetting Code (RC) is introduced. The issuer should give the RC to the user together with his password. The Resetting Code has the same format as the password and is stored in a DO 'RC'. The maximum length is announced in PW Status bytes, the minimum length is 8 bytes. The Resetting Code can be used within the command RESET RETRY COUNTER instead of the admin-password (PW3). It is only valid for resetting PW1. By default DO 'RC' is empty and the related error counter is zero, so it cannot be used. The Resetting Code has an error counter with an initial value of 3. This error counter is readable with GET DATA. The DO 'RC' can be set to any value with a PUT DATA command after correct verification of the admin-password (PW3), the error counter then is set to 3.

4.3 Data Objects (DO)

To keep the interface to terminals simple and for the reason to integrate the OpenPGP application in different card OS, all relevant data elements for the application are stored as data objects. Terminals can run the application only with the SELECT FILE, GET DATA, PUT DATA and cryptographic commands. Changing of any file identifier, short file identifier, file type or file structure has no influence on the terminal interface. DOs are stored in a TLV (Tag, Length, Value) format, whenever possible definitions of ISO (e.g. 7816-6) are used.

4.3.1 DOs for GET DATA

The following DOs shall be supported by the GET DATA command. They can be accessed at least in the OpenPGP DF of the card. Simple DOs (S) return only the value with GET DATA. Constructed DOs (C, marked yellow) are returned including their tag and length. In constructed DOs additional DOs may occur (not defined here) but are not evaluated by the OpenPGP application in the terminal. The DOs in cursive letters cannot accessed directly with GET DATA as single DO, the OpenPGP application in the terminal uses the non-cursive DOs (mostly constructed) only. Some of the DOs are optional (marked green), the occurrence is announced in Extended capabilities. DOs may appear several times in the table, if they are defined as single DO and occur in constructed DOs also. The order of DOs in a constructed DO may vary.

Tag	Format	Description
0101	S	Optional DO for private use, max. 254 (dec.) bytes (binary, proprietary), can be used to store any information.
0102	S	Optional DO for private use, max. 254 (dec.) bytes (binary, proprietary), can be used to store any information.
0103	S	Optional DO for private use, max. 254 (dec.) bytes (binary, proprietary), can be used to store any information.
0104	S	Optional DO for private use, max. 254 (dec.) bytes (binary, proprietary), can be used to store any information.
4F	S	Application identifier (AID), 16 (dec.) bytes (ISO 7816-4)
5E	S	Login data, max. 254 (dec.) bytes (binary, proprietary) This DO can be used to store any information used for the Log-In process in a client/server authentication (e.g. user name of a network).
5F50	S	Uniform resource locator (URL, as defined in RFC 1738), up to 254 (dec.) bytes. The URL should contain a Link to a set of public keys in OpenPGP format, related to the card.
5F52	S	Historical bytes, up to 15 bytes (dec.), according to ISO 7816-4. Card capabilities shall be included.
65	C	Cardholder Related Data (Tag)
5B	*S*	*Name (up to 39 bytes (dec.), according to ISO/IEC 7501-1)*
5F2D	*S*	*Language preferences, max. 8 bytes (according to ISO 639)*
5F35	*S*	*Sex, 1 byte (according to ISO 5218)*
6E	C	Application Related Data (Tag)
4F	S	Application identifier (AID), 16 (dec.) bytes (ISO 7816-4)
5F52	S	Historical bytes, up to 15 bytes (dec.), according to ISO 7816-4. Card capabilities shall be included.
73	*C*	*Discretionary data objects (Tag)*
C0	*S*	*Extended capabilities* *10 bytes (dec.), Flag list*
C1	*S*	*Algorithm attributes signature* *1 Byte Algorithm ID, according to RFC 4880* *further bytes depending on algorithm (e.g. length modulus and length exponent)*
C2	*S*	*Algorithm attributes decryption*
C3	*S*	*Algorithm attributes authentication*

Tag	Format	Description
C4	S	PW Status Bytes (7 bytes, binary) 1st byte: 00 = PW1 (no. 81) only valid for one PSO:CDS command 01 = PW1 valid for several PSO:CDS commands 2nd byte: max. length of PW1 (user) 3rd byte: max. length of Resetting Code (RC) for PW1 4th byte: max. length of PW3 (admin) Byte 5, 6, 7 (first byte for PW1, second byte for Resetting Code, third byte for PW3): Error counter of PW1, RC and PW3. If 00 then the corresponding PW/RC is blocked. Incorrect usage decrements the counter, correct verification sets to default value = 03.
C5	*S*	*Fingerprints (60 bytes (dec.), binary, 20 bytes (dec.) each for Sig, Dec, Aut in that order), zero bytes indicate a not defined private key*
C6	*S*	*List of CA-Fingerprints (60 bytes (dec.), binary, 20 bytes (dec.) each) of “Ultimately Trusted Keys”. Zero bytes indicate a free entry. May be used to verify Public Keys from servers.*
CD	*S*	*List of generation dates/times of public key pairs, 12 bytes (dec.) binary. 4 bytes, Big Endian each for Sig, Dec and Aut. Each value shall be seconds since Jan 1, 1970. Default value is 00000000 (not specified).*
7A	C	Security support template (Tag)
93	*S*	*Digital signature counter (counts usage of Compute Digital Signature command), 3 bytes binary, ISO 7816-4*
7F 21	C	Cardholder certificate This DO is designed to store a certificate (e.g. X.509) for the AUT-key in the card. It can be used to identify the card in a client-server authentication, where specific non-OpenPGP-certificates are needed. The maximum length of this DO is announced in Extended capabilities. The content should be TLV-constructed, but is out of scope of this specification. It may be necessary to use GET RESPONSE to read the content with GET DATA.
C4	S	PW Status Bytes (7 bytes, binary) 1st byte: 00 = PW1 (no. 81) only valid for one PSO:CDS command 01 = PW1 valid for several PSO:CDS commands 2nd byte: max. length of PW1 (user) 3rd byte: max. length of Resetting Code (RC) for PW1 4th byte: max. length of PW3 (admin) Byte 5, 6, 7 (first byte for PW1, second byte for Resetting Code, third byte for PW3): Error counter of PW1, RC and PW3. If 00, then the corresponding PW/RC is blocked. Incorrect usage decrements the counter, correct verification sets to default value = 03.

4.3.2 DOs for PUT DATA

The following DOs are supported by the PUT DATA command. They can be accessed at least in the OpenPGP DF of the card.

Tag	Format	Description
0101	S	Optional DO for private use, max. 254 (dec.) bytes (binary)
0102	S	Optional DO for private use, max. 254 (dec.) bytes (binary)
0103	S	Optional DO for private use, max. 254 (dec.) bytes (binary)
0104	S	Optional DO for private use, max. 254 (dec.) bytes (binary)
4D	C	Extended Header list (used for key import), uses: 7F48 (Cardholder private key template) 5F48 (Cardholder private key)
5B	S	Name
5E	S	Login data
5F2D	S	Language preferences
5F35	S	Sex
5F50	S	Uniform resource locator (URL)
7F 21	C	Cardholder certificate This DO is designed to store a certificate (e.g. X.509) for the AUT-key in the card. It can be used to identify the card in a client-server authentication, where specific non-OpenPGP-certificates are needed. The maximum length of this DO is announced in Extended capabilities. The content should be TLV-constructed, but is out of scope of this specification. It may be necessary to use command chaining for writing the DO with PUT DATA.
C1	S	Optional DO (announced in Extended capabilities). Algorithm attributes signature
C2	S	Optional DO (announced in Extended capabilities). Algorithm attributes decryption
C3	S	Optional DO (announced in Extended capabilities). Algorithm attributes authentication
C4	S	Optional DO (announced in Extended capabilities). 1st PW Status Byte (1 byte binary): 00 = PW1 (No. 81) only valid for one PSO:CDS command 01 = PW1 valid for several PSO:CDS commands
C7	S	Fingerprint (binary, 20 bytes dec.) for signature key, format according to RFC 4880
C8	S	Fingerprint (binary, 20 bytes dec.) for decryption key
C9	S	Fingerprint (binary, 20 bytes dec.) for authentication key
CA	S	1st CA-Fingerprint in list (binary, 20 bytes dec.)
CB	S	2nd CA-Fingerprint in list (binary, 20 bytes dec.)
CC	S	3rd CA-Fingerprint in list (binary, 20 bytes dec.)
CE	S	Generation date/time of signature key (4 bytes Big Endian, format according to RFC 4880
CF	S	Generation date/time of decryption key (4 bytes Big Endian)
D0	S	Generation date/time of authentication key (4 bytes Big Endian)

Tag	Format	Description
D1	S	Optional DO (announced in Extended capabilities) for SM. SM-Key-ENC for cryptogram (24 bytes dec. in case of Triple-DES, 16 bytes in case of AES)
D2	S	Optional DO (announced in Extended capabilities) for SM. SM-Key-MAC for cryptographic checksum (24 bytes dec. in case of Triple-DES, 16 bytes in case of AES)
D3	S	Resetting Code, 0 or 8 – xx bytes (dec.), binary
D5	S	Optional DO for PSO:DEC with AES (32 bytes dec. in case of AES256, 16 bytes dec. in case of AES128). The OpenPGP application should support AES128 and -256 for DECIPHER, there is no further announcement.
F4	C	Optional DO (announced in Extended capabilities) for SM. Container for both SM keys (ENC and MAC) with Tags D1 and D2. Useful for updating or deleting both keys simultaneous.

4.3.3 DOs in Detail

The following chapter describes some DOs in detail, especially the proprietary DOs.

4.3.3.1 Private Use

These optional DOs can be used by the card holder, administrator or any application for proprietary data (e.g. password list). The difference between the DOs are the access conditions. The presence of this DOs is announced in Extended capabilities.

4.3.3.2 Name

This interindustry data element consists of up to 39 bytes, each byte is a character from ISO 8859-1 (Latin 1) alphabet (identical to 7-bit-US-ASCII for characters < 80). The data element consists of surname (e.g. family name and given name(s)) and forename(s) (including name suffix, e.g., Jr. and number). Each item is separated by a ´<´ filler character (3C), the family- and fore-name(s) are separated by two ´<<´ filler characters.

4.3.3.3 Language Preferences

This data element consists of 1 to 4 pairs of bytes (e.g. 2 bytes or 6 bytes) with coding according to ISO 639-1, ASCII lower case (e.g. de = german; en = english; nl = dutch; fr = french). At least one entry (2 bytes) should be present, the first entry has highest priority. The information can be used by the terminal for the user interface (e.g. language of text).

4.3.3.4 Sex

This data element of 1 byte (binary) represents the ´Sex´ of a person according to ISO 5218. The following values are defined for the OpenPGP application:

Male	31
Female	32
Not announced	39

The terminal can use the information for the user interface.

4.3.3.5 Extended Capabilities

The Extended capabilities consists of 10 bytes (dec.) with the following meaning:

The first byte is a table that indicates additional features to the terminal. A set bit (1) means that the function is available, a value equalling zero means that the function is not available. Bits can be set simultaneous.

Coding of byte 1 of Extended capabilities:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	-	-	-	-	-	-	-	Secure Messaging supported
-	1	-	-	-	-	-	-	Support for GET CHALLENGE The maximum supported length of a challenge can be found in special bytes of Extended capabilities.
-	-	1	-	-	-	-	-	Support for Key Import
-	-	-	1	-	-	-	-	PW1 Status byte changeable (DO C4 available for PUT DATA)
-	-	-	-	1	-	-	-	Support for Private use DOs (0101-0104)
-	-	-	-	-	1	-	-	Algorithm attributes changeable with PUT DATA
-	-	-	-	-	-	0	0	RFU

The other bytes are coded as follows:

Byte	Length	Value
02	01	Secure Messaging Algorithm 00 = Triple-DES (168 bit. dec.) 01 = AES (128 bit, dec.) If SM is not supported (see 1st byte), the coding is 00
03 - 04	02	Maximum length of a challenge supported by the command GET CHALLENGE (unsigned integer, Most Significant Bit ... Least Significant Bit) If GET CHALLENGE is not supported (see 1st byte), the coding is 0000
05 - 06	02	Maximum length of Cardholder Certificate (DO 7F21). Coded as unsigned integer (Most Significant Bit ... Least Significant Bit). If the length is > FF and the card does not support Extended length, then command chaining shall be used for PUT DATA. The same is relevant if the card supports Extended length, but the 'Maximum length of command data' is smaller than the length of the DO.
07 - 08	02	Maximum length of command data In future ISO specifications this information is available in an EF_ATR under the MF. Because the definition is not finished yet and OpenPGP supports cards without MF too, the information is given here. The unsigned integer value (Most Significant Bit ... Least Significant Bit) defines the maximum length of the command data field (including SM) in any command, for Short or Extended length (Lc/Le). If a card supports Extended length (see card capabilities), a maximum length of FF is assumed für Short length. If command data exceeds the maximum length (e.g. PSO:DECRYPT), then the command shall be send with command chaining.
09 - 0A	02	Maximum length of response data The unsigned integer value (Most Significant Bit ... Least Significant Bit) defines the maximum length of the response field (including SM) in any command, for Short or Extended length (Lc/Le). If a card supports Extended length (see card capabilities), a maximum length of 256 (dec.) is assumed for Short length. If a response exceeds the maximum length (e.g. GET DATA), then the card should answer with status bytes 61xx and remaining data can be read with GET RESPONSE.

4.3.3.6 Algorithm Attributes

This DO announces information related to the supported algorithm of the card. The terminal shall use this information for the key import functionality (if available). The formats are used by the key generation in the card and are also related to the output format of the corresponding command.

The content of the DO is optionally changeable (announced in Extended capabilities) with PUT DATA. This is useful if the card supports several algorithms or different key length. The attributes can be changed independent for each key, so it is possible for example to use different key length for signing and decrypting. A card should reject unsupported values in the DO. The supported values are manufacturer specific, please ask your card vendor for the correct parameters if the card supports changing of the attributes.

If the attributes of an existing key are changed and do no longer match with the stored key, the card should delete this key internally or prevent it from usage.

RSA:

Byte	Length	Value
01	01	Algorithm ID (RFC 4880) 01 = RSA (Encrypt or Sign)
02 - 03	02	Length of modulus n in bit (e.g. 1024 bit decimal = 0400), binary
04 - 05	02	Length of public exponent e in bit (e.g. 32 bit decimal = 0020), binary
06	01	Format of private key 00 = standard (e, p, q) 01 = standard with modulus (n) 02 = crt (Chinese Remainder Theorem) 03 = crt with modulus (n)

This version defines only the content for the RSA algorithm. Future cards may support ECDSA as recommended algorithm, but it is not supported by OpenPGP yet.

4.3.3.7 Private Key Template

If the card supports key import (see Extended Capabilities), the terms of the corresponding private key are coded in the following way. The function does not matter how the key is stored in the card internally. The function does not set the value of the corresponding fingerprint. The key import uses a PUT DATA command with odd INS (DB) and an Extended header list (DO 4D) as described in ISO 7816-8.

The DOs in the Extended Header list start with a Control Reference Template (CRT) of the referenced key.

Digital signature:	B6 00 (Tag, Length)
Confidentiality:	B8 00
Authentication:	A4 00

The next DO consists of a Cardholder private key template (7F48), that describes the input and the length of the content of the following DO. The last DO (Cardholder private key, 5F48) represents a concatenation of the key data elements according to the definitions in DO 7F48.

This version defines the input for PUT DATA for RSA keys only. The order of the DOs is mandatory (the card may support any order), xx means a length field (1, 2 or 3 bytes). Unnecessary DOs in between DO 7F48 should be stripped off (see format of private in Algorithm attributes). Descriptions are in cursive letters, coded data are marked yellow.

4D	xx	*Extended Header list*					
		B6 or B8 or A4	00	*Control Reference Template to indicate the private key*			
		7F48	xx	*Cardholder private key template*			
				91	xx	*Public exponent: e*	*key format: standard and crt*
				92	xx	*Prime1: p*	*standard and crt*
				93	xx	*Prime2: q*	*standard and crt*
				94	xx	*PQ: 1/q mod p*	*crt*
				95	xx	*DP1: d mod (p - 1)*	*crt*
				96	xx	*DQ1: d mod (q - 1)*	*crt*
				97	xx	*Modulus: n*	*optional for standard and crt*
		5F48	xx	Concatenation of key data as defined in DO 7F48			

Several chips for smart cards support only one coding for the Public exponent (e.g. 65537 dec.). The card may reject an imported key, if e does not match the internal requirements. But the card shall accept 65537 dec. (10001) as value for e.

Some chips are not able to calculate n from p and q. In that case the modulus can be integrated in the import. This feature is announced in 'Algorithm attributes'.

The length of the key data shall match the values given in the DO 'Algorithm attributes' (C1 – C3). E.g., if the Modulus n has a length of 1024 bit (dec.), then p and q have a fixed length of 512 bits each.

In case of a signature key (CRT B6), the card internally resets the signature counter to zero.

4.3.4 Length Field of DOs

According to ISO 7816-4 the length field in TLV-structures has the following format:

Number of bytes	First byte	Second byte	Third byte	Value (dec.)
1	00 – 7F	-	-	0 – 127
2	81	00 - FF	-	0 – 255
3	82	0000 - FFFF		0 - 65535

5 Security Architecture

All commands and data of a smart card are under control of the security of the card operating system. ISO defines mechanisms, attributes (e.g. in FCP) and environments for security purposes. Because this features are quite complex and may differ from card to card (depending on mask developer), the OpenPGP application does not evaluate security related items of a card. So this chapter is informative for the card developer and defines the access conditions for all commands and data objects of the application in a common way. The described security features are mandatory for the card, but the coding or the way of implementation is up to the card developer, manufacturer or personaliser. Private keys and passwords cannot be read from the card with any command or function. Commands and data have access conditions to be fulfilled. The following tables show all access conditions for the OpenPGP application. READ is a synonym for all functions and commands of the operations system that present data to the external world, WRITE is a synonym for all functions and commands that change data in the Eeprom of the chip. If constructed DOs are processed, the access conditions of each single DO shall be fulfilled.

Access conditions for relevant commands:

Command	Access condition	Description
SELECT FILE	Always	
GET DATA	Various	Depending on Data Objects
VERIFY	Always	
CHANGE REFERENCE DATA	Always (version >= 2.0) *VERIFY of corresponding PW (up to version 1.1)*	
RESET RETRY COUNTER	VERIFY of PW3 or Resetting Code	
PUT DATA	Various	Depending on Data Objects
GENERATE ASYMMETRIC KEY PAIR	Always (read pub-key) VERIFY of PW3 (generate key-pair)	
PSO: COMPUTE DIGITAL SIGNATURE	VERIFY of PW1	With PW no. 81
PSO: DECIPHER	VERIFY of PW1	With PW no. 82
INTERNAL AUTHENTICATE	VERIFY of PW1	With PW no. 82
GET CHALLENGE	Always	
TERMINATE DF	VERIFY of PW3 or Always if blocked	
ACTIVATE FILE	Always	
Other commands	Various	e.g. commands for personalisation

Access conditions for Data Objects:

Data Object	READ	WRITE	Description
Private use (0101)	Always	Verify PW1	With PW no. 82
Private use (0102)	Always	Verify PW3	
Private use (0103)	Verify PW1	Verify PW1	With PW no. 82
Private use (0104)	Verify PW3	Verify PW3	
AID (4F)	Always	Never	Writing possible only during personalisation (manufacturer)
Login data (5E)	Always	Verify PW3	
Name (5B)	Always	Verify PW3	
Language preference (5F2D)	Always	Verify PW3	
Sex (5F35)	Always	Verify PW3	
URL (5F50)	Always	Verify PW3	
Historical bytes (5F52)	Always	Never	Writing possible only during personalisation
Card holder private key (5F48)	Never	Verify PW3	Relevant for all private keys in the application (signature, decryption, authentication)
Cardholder certificate (7F21)	Always	Verify PW3	
DS-Counter (93)	Always	Never	Internal Reset during key generation
Extended capabilities (C0)	Always	Never	Writing possible only during personalisation
Algorithm attributes (C1 – C3)	Always	Verify PW3	
PW1 Status bytes (C4)	Always	Verify PW3	Only 1st byte can be changed, other bytes only during personalisation
Fingerprints (C5, C7 – C9)	Always	Verify PW3	
CA-Fingerprints (C6, CA – CC)	Always	Verify PW3	
Generation date/time of key pairs (CD – D0)	Always	Verify PW3	
SM-Key-ENC (D1)	Never	Verify PW3	
SM-Key-MAC (D2)	Never	Verify PW3	
Resetting Code (D3)	Never	Verify PW3	
SM-Key-Container (F4)	Never	Verify PW3	

In case that the card supports Secure Messaging, a correct command in SM mode is equivalent with reaching the access condition for PW3 (VERIFY PW3).

6 Historical Bytes

Historical bytes are part of the ATR (Answer To Reset) from a card, the ´format byte (T0)´ indicates the presence of Historical bytes in bits 1- 4, according to ISO 7816-3. An alternative way of reading Historical bytes is a special DO with the Tag '5F52'. Because not all cards provide Historical bytes with an appropriate content, the OpenPGP application uses this DO. In the Historical bytes the presence of a DO Card capabilities is relevant. It is coded according to ISO 7816-4.

The DO 'Historical bytes' (5F52) can be read with the GET DATA command from within the OpenPGP card application. It is relevant for the application and may differ from Historical bytes in the ATR.

The first Historical byte is the "category indicator byte". If the category indicator byte is set to '00', '10' or '80', then the format is according is ISO. Any other value indicates a proprietary format. The OpenPGP application assumes a category indicator byte set to '00'.

If the first Historical byte is set to '00', then the remaining Historical bytes consist of optional consecutive COMPACT-TLV data objects followed by a mandatory status indicator (the last three bytes, not in TLV).

The COMPACT-TLV format has a Tag in the first nibble of a byte (bit 5-8) and a length in the second nibble (bit 1-4). For the OpenPGP application only a TL with 73 is relevant. It announces a DO 'Card capabilities' with 3 bytes. Other DOs are allowed.

The last 3 bytes of the Historical bytes in this format are a status indicator byte and two processing status bytes SW1/SW2 (normally 9000).

The status indicator byte is evaluated by the OpenPGP application as follows:

00 =	No information given Card does not offer life cycle management, commands TERMINATE DF and ACTIVATE FILE are not supported
03 =	Initialisation state OpenPGP application can be resetted to default values with an ACTIVATE FILE command
05 =	Operational state (activated) Card supports life cycle management, commands TERMINATE DF and ACTIVATE FILE are available

6.1 Card Capabilities

This interindustry data element consists of three software function tables (1 byte each) according to ISO 7816-4. The first software function table indicates selection methods supported by the card. The second software function table is the "data coding byte". The third software function table indicates the ability to chain commands, to extend Lc and Le fields and to handle logical channels. A set Bit (1) means that the function is available (unless otherwise specified), a value equal zero means that the function is not available. Bits can be set simultaneous. For the OpenPGP application only the third table (byte 3) is relevant (functions in cursive will not be evaluated in this version).

Command chaining, length fields and logical channels (third byte):

b8	B7	b6	b5	B4	b3	b2	b2	Meaning
1	-	-	-	-	-	-	-	Command chaining
-	1	-	-	-	-	-	-	Extended Lc and Le fields
-	-	-	x	x	-	-	-	*Logical channel number assignment*
-	-	x	-	-	y	z	t	*Maximum number of logical channels*

7 Commands

The OpenPGP application is based on the functionality of ISO 7816-4, -8 and -9. Thus the standard OS commands are available to the 'external environment', it depends on the current OS, how the code for the commands is stored.

7.1 Usage of ISO Standard Commands

The following table shows all standard commands of an ISO operating system, which are used by the OpenPGP application. Only the given subsets (P1/P2) of a command shall be implemented, however the card may provide other functions.

Command	INS	P1	P2	Comment
SELECT FILE	A4	04	00	AID = 1-16 Byte (partial AID is recommended) P2 = 00 for first or only occurrence
GET DATA	CA	xx	xx	Fully supported for defined DOs
VERIFY	20	00	81/82/83	Local PW1 or PW3
CHANGE REFERENCE DATA	24	00 (01)	81/83	Change of PW1 or PW3 P1=01 is supported only up to version 1.1
RESET RETRY COUNTER	2C	00/02	81	Resets the retry counter of PW1 and sets a new value for PW1. In the command data the new PW1 is present.
PUT DATA	DA or DB	xx	xx	Fully supported for defined DOs
GENERATE ASYMMETRIC KEY PAIR	47	80/81	00	P1=80: Generation of internal private key, public key in response (DO 7F49) P1=81: Reading of actual public key Relevant key is addressed by a CRT in the command data.
PERFORM SECURITY OPERATION (PSO)	2A	xx	xx	As defined in the next lines.
COMPUTE DIGITAL SIGNATURE	2A	9E	9A	Input are plain data (e.g. hash code), length must match the algorithm and key length of the card, digital signature in response
DECIPHER	2A	80	86	Input: Padding indicator byte (always 00) and encrypted data, length of encrypted data must

Command	INS	P1	P2	Comment
				match algorithm and key length Response: Plain data
INTERNAL AUTHENTICATE	88	00	00	Authentication input related to algorithm
GET RESPONSE	C0	00	00	Used under T=0 and for retrieving long DOs with GET DATA under any protocol
GET CHALLENGE	84	00	00	Fully supported (Le defines length of random number), optional command. If supported the card shall provide any length according to simple Le or extended Le
TERMINATE DF	E6	00	00	The command puts the application into the termination state. After termination only SELECT FILE and ACTIVATE FILE are available
ACTIVATE FILE	44	00	00	In the OpenPGP application the termination state is equivalent to the initialisation state. An ACTIVATE FILE command in this stage resets all DOs to default values and sets the application into the operational state. The usage of this command in operational state (not terminated) has no effect to any data

Additional commands for production, personalization and other applications are out of scope of this specification.

7.2 Commands in Detail

The following section describes some of the commands in more detail. In all examples short Lc/Le is used. If the card provides extended Lc/Le than the terminal may extend the fields to a length of 2 or 3 bytes.

7.2.1 SELECT FILE

With this command the OpenPGP application in the terminal selects the corresponding application on the card. Only the significant bytes of the AID are presented in the command data. Possible response data (FCI) are not evaluated by the application.

Command:

CLA	00
INS	A4
P1	04
P2	00
Lc	06
Data field	D2 76 00 01 24 01
Le	00

Response:

Data field	FCI or empty
SW1-SW2	9000 or specific status bytes

7.2.2 VERIFY

With this command one of the passwords of the application is verified. PW1 has two modes and can be used with 2 different numbers (in P2). PW 1 with P2 = 81 sets the access conditions for a PSO:CDS command only. Depending on the PW1 Status byte (see Extended capabilities) this access condition is only valid for one PSO:CDS command or remains valid for several attempts. PW1 with P2 = 82 sets the access condition for several other functions.

Command:

CLA	00 / 0C
INS	20
P1	00
P2	81 (PW1) or 82 (PW1) or 83 (PW3)
Lc	xx (min. 06 for PW1, min. 08 for PW3, max. see DO ´C4´)
Data field	Corresponding PW
Le	Empty (means not present in command)

Response:

Data field	None
SW1-SW2	9000 or specific status bytes

7.2.3 CHANGE REFERENCE DATA

With this command the passwords of the application can be changed. In compliance with EN 14890 only one P1 variation from ISO 7816-4 is defined for signature cards. For that reason the coding of the command was changed in this version. The command can be accessed without restrictions, the actual PW is part of the command data and will be verified first by the card. The length of the existing password 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_{new} = Lc – $L_{old.}$

Command:

CLA	00 / 0C / 10 / 1C
INS	24
P1	00
P2	81 (PW1) or 83 (PW3)
Lc	xx
Data Field	Actual PW + New PW Length of new PW: min. 06 for PW1, min. 08 for PW3 For max. length see DO ´C4´
Le	Empty (means not present in command)

Response:

Data field	None
SW1-SW2	9000 or specific status bytes

7.2.4 RESET RETRY COUNTER

With this command the error counter and the value of PW1 can be resetted, that means the new value is stored and the error counter is set to the default value (3). RESET RETRY COUNTER can be used after correct presentation of PW3 (P1 = 02) or by presenting the Resetting Code (DO D3) in the command data (P1 = 00). Usage of secure messaging is equivalent to PW3. The length of the Resetting Code 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_{new} = Lc – $L_{RC.}$

Command:

CLA	00 / 0C / 10 / 1C
INS	2C
P1	00 / 02
P2	81 (PW1)
Lc	xx (min. 06 for PW1, max. see DO ´C4´)
Data field	New PW (P1 = 02) Resetting Code + New PW (P1 = 00)
Le	Empty (means not present in command)

Response:

Data field	None
SW1-SW2	9000 or specific status bytes

7.2.5 GET DATA

With this command DOs can be read from the card. The Tag (simple or constructed) is given in P1/P2 (e.g. 5F50 for URL or 006E for Application Related Data). For simple DOs only the value is in the response field (e.g. 5F50 = URL returns a byte string representing the URL without leading Tag/Length). For constructed DOs all values returned are encapsulated with Tag/Length (e.g. 0065 = Cardholder Related Data returns the concatenation of following DOs (L = Length): 5B L Name 5F2D L Language Preferences 5F35 L Sex). If the DO is longer than the maximum supported length for a response of a card, then the card answers with status bytes 61xx and return only the first part of the data, xx indicate the remaining data in the card. The data may be truncated at any position and shall be concatenated later in the terminal. The terminal can read the missing data with a following GET RESPONSE command and Le = 00 (or 0000 for extended Le). This can be repeated several times (another status byte 61xx). The reading of data is complete if any command (GET DATA or GET RESPONSE) answers with status bytes 9000.

Command:

CLA	00 / 0C
INS	CA
P1	xx (00 if Tag has a length of only one byte)
P2	xx
Lc	Empty
Data Field	None
Le	00

Response:

Data field	Addressed data or DOs (maybe partially)
SW1-SW2	9000 or 61xx or specific status bytes

7.2.6 PUT DATA

With this command DOs can be written to the card. The Tag is given in P1/P2 (e.g. 5F50 for URL or 005B for Name). For simple DOs only the value is in the data field (without leading Tag/Length). The command can only be used after correct presentation of PW3 (except DO 0101 and DO 0103 after correct verification of PW1 with No. 82).

For key import in compliance with ISO 7816-8 an Extended header list is supported with an odd INS of the command. In that case P1/P2 has the value 3FFF (references the current DF for the DOs in the Extended header list).

An empty data field (Lc = 0) deletes the content of a DO with variable length (e.g. URL), but not the DO itself.

Command:

CLA	00 / 0C / 10 / 1C
INS	DA / DB
P1	xx = 00 if Tag has a length of one byte only = 3F in case of odd INS
P2	xx (FF in case of odd INS)
Lc	xx
Data Field	Addressed data or Extended header list
Le	Empty

Response:

Data field	None
SW1-SW2	9000 or specific status bytes

7.2.7 GET RESPONSE

This command is needed under T=0 for some command cases according to ISO 7816-3 and under any protocol (e.g. T=1) for receiving long data that cannot be transmitted in one response.

After receiving status bytes with 61 xx, the terminal should send a GET RESPONSE command with xx or 00 in the Le field.

Command:

CLA	00 / 0C
INS	C0
P1	00
P2	00
Lc	Empty
Data field	Empty
Le	00 (xx given in previous SW2)

Response:

Data field	Data
SW1-SW2	9000 or 61xx or specific status bytes

7.2.8 PSO: COMPUTE DIGITAL SIGNATURE

The command for digital signature computation is shown in the table below. The hash value (ECDSA) or the DigestInfo is delivered in the data field of the command. Signature key as well as signature algorithm and the related Digital-Signature-Input formats are implicitly selected. The command is only possible after correct presentation of PW1 with No. 81. The command internally checks the PW1 Status byte DO (first byte), if the value is 00, then the access condition (No. 81 of PW1) is reset and the PW has to be verified again for the following command.

Command:

CLA	00
INS	2A
P1	9E
P2	9A
Lc	Length of subsequent data field
Data field	Data to be integrated in the DSI: hash value or DigestInfo
Le	00

Response:

Data field	Digital signature
SW1-SW2	9000 or specific status bytes

The DSI format for RSA:

According to PKCS #1 the DSI is generated internally by the card and has the following structure:

Description	Length	Value
Start byte	1	00
Block type	1	01
Padding string (PS)	N-3-L	FF ... FF
Separator	1	00
Data field	L	DigestInfo

In compliance with PKSC #1, the card checks that the DigestInfo in the command data field is not longer than 40% of the length of the modulus of the signature key, otherwise the command is rejected.

DSI for ECDSA:
The DSI consists of the hash value which was calculated (20, 32 or 64 bytes, dec.). If the DSI is longer than the hash value (e.g. if q is longer than 160 bits, dec.), then the DSI is filled with leading zero bits by the card.

DSI for DSA:
The DSI consists of the hash value calculated (20, 32 or 64 bytes, dec.). Maybe supported by cards with version up to V1.1, DSA is not supported by cards with versions >= 2.0.

7.2.8.1 Hash Algorithms

The following hash algorithms are supported by RFC 4880 and can be used as input in the DSI. However the card may not check the integrity of a DSI. Cards with Version < 2.0 support RIPEMD-160 and SHA-1 only and may check it, so other hash algorithms cannot be used.

Hash algorithm	***Length of hash-code (dec.)***
SHA-1	20
RIPEMD-160	20
SHA-224	28
SHA-256	32
SHA-384	48
SHA-512	64

7.2.8.2 DigestInfo for RSA

The following tables show the contents of defined DigestInfos for RSA. The card may not check the correctness of the DigestInfo.

SHA-1:

Tag	***Length***	***Value***	***Description***
30	21		Tag/Length of Sequence
30	09		Tag/Length of Sequence
06	05	2B0E03021A	OID of SHA-1 {1 3 14 3 2 26}
05	00	-	TLV coding of ZERO
04	14	xx...xx	hash-code

RIPEMD-160:

Tag	*Length*	*Value*	*Description*
30	21		Tag/Length of Sequence
30	09		Tag/Length of Sequence
06	05	2B24030201	OID of RIPEMD-160 {1 3 36 3 2 1}
05	00	-	TLV coding of ZERO
04	14	xx...xx	hash-code

SHA-224:

Tag	*Length*	*Value*	*Description*
30	2D		Tag/Length of Sequence
30	0D		Tag/Length of Sequence
06	09	608648016503040204	OID of the SHA-224 {2 16 840 1 101 3 4 2 4}
05	00	-	TLV coding of ZERO
04	1C	xx...xx	hash-code

SHA-256:

Tag	*Length*	*Value*	*Description*
30	31		Tag/Length of Sequence
30	0D		Tag/Length of Sequence
06	09	608648016503040201	OID of the SHA-256 {2 16 840 1 101 3 4 2 1}
05	00	-	TLV coding of ZERO
04	20	xx...xx	hash-code

SHA-384:

Tag	*Length*	*Value*	*Description*
30	41		Tag/Length of Sequence
30	0D		Tag/Length of Sequence
06	09	608648016503040202	OID of the SHA-384 {2 16 840 1 101 3 4 2 2}
05	00	-	TLV coding of ZERO
04	30	xx...xx	hash-code

SHA-512:

Tag	*Length*	*Value*	*Description*
30	51		Tag/Length of Sequence
30	0D		Tag/Length of Sequence
06	09	608648016503040203	OID of the SHA-512 {2 16 840 1 101 3 4 2 3}
05	00	-	TLV coding of ZERO
04	40	xx...xx	hash-code

7.2.9 PSO: DECIPHER

The command is used by the application as key decipherment service. The command can be used after correct presentation of PW1 (with No. 82) only. For confidential document exchange, the following scheme is applied:

- The key transport is organised by enciphering the content encryption key with the receivers public key.
- The document enciphering is done with a symmetrical algorithm (e.g. AES).

The card is not involved in the enciphering of the document. The software computes the content encryption key, enciphers the document and finally enciphers the content encryption key by using the receivers public key. The card performs a key decryption applying the private key for decryption in a DECIPHER command to the cryptogram contained in the data field of the command.

By option the card supports the decryption of a plain text with an AES-key stored in a special DO (D5). This is useful if no certificate or public key exists and the external world has a common secret with the card. The functionality can be used e. g. for storing an encrypted transport key or seed on a flash card/stick or hard disc, only the relevant card is able to decrypt the secret if the AES-key is deleted in the outside world after encryption and storing into the card.

In case of the RSA algorithm the command input (except padding indicator byte) shall be formatted according to PKCS#1 before encryption:

Description	Length	Value
Start byte	1	00
Block type	1	02
Padding string (PS)	N-3-L	Non-zero-random-bytes
Separator	1	00
Data field	L	Message

PS is a byte string consisting of randomly generated nonzero bytes. The length of PS must be at least 8 bytes. The formatted string must consist of N bytes where N is the length of the modulus of the private key for decryption. The Padding indicator byte and the encrypted message is given to the command in the command data. The card decrypts all bytes after the padding indicator byte, checks the conformance of correct PKCS#1 padding and returns the plain text (length = message) in the response.

In case of the AES algorithm the command input (except padding indicator byte) shall be exact 16 bytes. A key shall be present in DO (D5), the Padding indicator byte (02) and the encrypted message is given to the command in the command data. The card decrypts all bytes after the padding indicator byte and returns the plain text (16 bytes) in the response.

Command:

CLA	00
INS	2A
P1	80 = Return plain value
P2	86 = Enciphered data present in the data field
Lc	xx = Length of subsequent data field
Data field	Padding indicator byte (00) for RSA or (02) for AES followed by cryptogram
Le	00

Response:

Data field	Plain data
SW1-SW2	9000 or specific status bytes

Input in case of ECDSA:
In this version of the OpenPGP application ECDSA decryption is not defined and will be added in a later version.

7.2.10 INTERNAL AUTHENTICATE

The INTERNAL AUTHENTICATE command can be used for Client/Server authentication. The usage is up to the terminal, the card only provides this command for asymmetric algorithms. The input data shall be an Authentication Input (AI) compliant with PKCS#1, the card does an internal PKCS#1-padding and calculates a signature with the corresponding secret key for authentication. The command can be used after correct presentation of PW1 (with No. 82) only.

In compliance with PKSC #1, the card checks that the AI in the command data field is not longer than 40% of the length of the modulus of the signature key, otherwise the command is rejected.

Command:

CLA	00
INS	88 = INTERNAL AUTHENTICATE
P1	00
P2	00
Lc	xx = Length of subsequent data field
Data Field	Authentication Input (AI) for RSA: Lc <= 0,4 * N, e.g.: Lc <= 51 for 1024 bit modulus Lc <= 102 for 2048 bit modulus for ECDSA: Hash value (values are decimal)
Le	00

Response:

Data field	Signature
SW1-SW2	9000 or specific status bytes

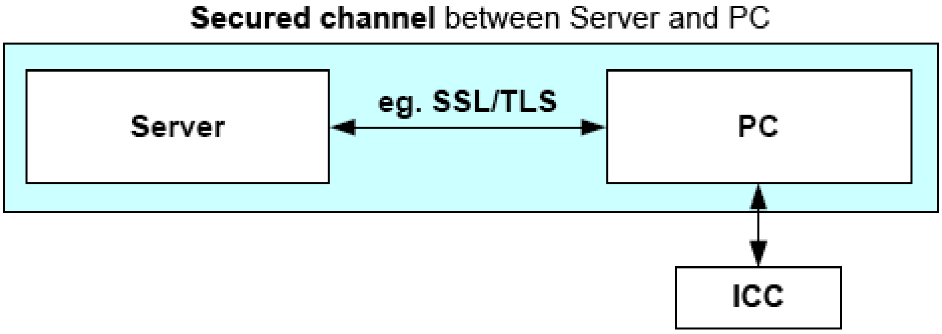
PKCS#1-Padding for Authentication Input used with RSA:

Description	Length	Value
Start byte	1	00
Block type	1	01
Padding string (PS)	N-3-L	FF...FF
Separator	1	00
Data field	L	Authentication Input (AI)

The resulting input for the signature in case of RSA has the length N. The card calculates the signature with the private key for authentication: sign(SK_{Aut})[00 | 01 | PS | 00 | AI] and returns the result as authentication data in the response.

7.2.10.1 Client/Server Authentication

This specification covers only the case where the card performs a digital signature computation applying the private key for authentication in an INTERNAL AUTHENTICATE command to the authentication input contained in the data field of the command after formatting the input. The mechanism can be used, for example, with Secure Shell (SSH), PK Kerberos, SSL/TLS or WTLS. All these protocols base on the same cryptographic algorithms. In particular they are all using PKCS #1 padding format in the case of RSA.



In the above example (taken from EN 14890-2), client/server authentication establishes a secured channel between a remote server and a PC. The ICC will be used as a cryptographic toolbox in order to provide the cryptographic functionality to the PC. If a process needs a certificate from the card that is not provided by OpenPGP (e.g. x.509), then it can be stored in the DO 'Cardholder certificate'.

7.2.11 GENERATE ASYMMETRIC KEY PAIR

This command either initiates the generation and storing of an asymmetric key pair, i.e., a public key and a private key in the card, or returns the public key of an asymmetric key pair previously generated in the card or imported. The card should use 65537 dec. (1001) as default value for e, but other values are accepted by the outside world. In case of key pair generation the command does not set the values of the corresponding fingerprint.

After receiving the public key the terminal has to calculate the fingerprint and store it in the relevant DO. The generation of a key pair for digital signature resets the digital signature counter to zero (000000). The command can only be used after correct presentation of PW3 for the generation of a key pair. Reading of a public key is always possible.

Command:

CLA	00 / 0C
INS	47
P1	80 = Generation of key pair 81 = Reading of actual public key template
P2	00
Lc	02
Data field	CRT for relevant function
Le	00

Response:

Data field	Public key as a set of data objects
SW1-SW2	9000 or specific status bytes

Defined CRTs (Control Reference Template) for the command (generation of key pair or reading of public key).

Digital signature:	B6 00 (Tag, Length)
Confidentiality:	B8 00
Authentication:	A4 00

Defined DOs for response (xx = Length):

7F49 xx

Set of public key data objects for RSA
81 xx Modulus (a number denoted as n coded on x bytes)
82 xx Public exponent (a number denoted as v, e.g. 65537 dec.)

Set of public key data objects for ECDSA (not supported by OpenPGP yet)
81 xx Prime (a number denoted as p coded on z bytes)
82 xx First coefficient (a number denoted as a coded on z bytes)
83 xx Second coefficient (a number denoted as b coded on z bytes)
84 xx Generator (a point denoted as PB on the curve, coded on 2z or z+1 bytes)
85 xx Order (a prime number denoted as q, order of the generator PB, coded on z bytes)
86 xx Public key (a point denoted as PP on the curve, equal to x times PB where x is the private key, coded on 2z or z+1 bytes)
87 xx Co-factor

Set of public key data objects for DSA (only available in cards with version < 2.0)
81 xx First prime (a number denoted as p coded on y bytes)
82 xx Second prime (a number denoted as q dividing p-1)
83 xx Basis (a number denoted as g of order q coded on y bytes)
84 xx Public key (a number denoted as y equal to g to the power x mod p where x is the private key coded on y bytes)

7.2.12 GET CHALLENGE

This optional command (announced in Extended Capabilities) generates a random number of the length given in Le. It is a service to the terminal application because smart cards often generate high sophisticated random numbers by certified hardware. Several smart card implementations have limitations for the length of the random number, so the maximum length is announced in Extended capabilities.

The command is mandatory if the cards supports Secure Messaging.

Command:

CLA	00
INS	84
P1	00
P2	00
Lc	Empty
Data field	Empty
Le	xx (01-FF for Short Le, 0001-FFFF for Extended Le)

Response:

Data field	Challenge with length xx
SW1-SW2	9000 or specific status bytes

7.2.13 TERMINATE DF

This optional command (announced in Life Cycle Status indicator in Historical bytes) sets the Life Cycle Status indicator to 03 and puts the card into initialisation state. The behaviour of the application is similar to the termination state, no commands can be used except SELECT FILE, which return specific status bytes (6285). After a selection an ACTICATE FILE command is possible. That command is designed to renew a card in case of blocked passwords or other problems. The command is allowed after correct verification of PW3 (Admin-PW) or secure messaging (SM) if PW3 is disabled.

If PW3 is blocked or not accessable (e. g. memory failure) and SM is disabled, then TERMINATE can be used without access conditions. This prevents the card from any status that makes it unusable.

Command:

CLA	00 / 0C
INS	E6
P1	00
P2	00
Lc	Empty
Data field	Empty
Le	Empty

Response:

Data field	Empty
SW1-SW2	9000 or specific status bytes

7.2.14 ACTIVATE FILE

This optional command (announced in Life Cycle Status indicator in Historical bytes) can be used to reset all values (DOs, Keys, PWs etc.) to their default values (after production/ initialisation). The command has effect only, if the life cycle status is in initialisation state (indicator byte set to 03). If the command is used in operational state, it will end with status bytes 9000, but nothing in the application will be changed (dummy command).

Command:

CLA	00
INS	44
P1	00
P2	00
Lc	Empty
Data field	Empty
Le	Empty

Response:

Data field	Empty
SW1-SW2	9000 or specific status bytes

7.3 Command Usage under Different I/O Protocols

For the OpenPGP application the T=1 protocol (ISO 7816-3) is recommended for cards with contacts. However other protocols (one or more) in a card are possible too. The OpenPGP application is designed to run under every protocol (e.g. T=0, contactless) that is provided by the card reader. If contactless protocols (e.g. ISO 14443) are implemented, card and reader should support Secure Messaging (e.g. VERIFY) to avoid data tracking 'over the air'.

7.4 Class Byte Definitions

For the OpenPGP application all standard commands are used with a class byte (CLA) coding according to ISO. The following values are defined.

CLA	Description
00	CLA without SM (last or only command of a chain)
0C	CLA with SM and header authentication (last or only command of a chain)
10	CLA without SM (command is not the last command of a chain)
1C	CLA with SM and header authentication (command is not the last command of a chain)

7.5 Secure Messaging (will be changed in next version!)

The OpenPGP application defines secure messaging with a static keys in this version in compliance with EN 14890. Command data are encrypted first and then protected by a cryptogram. SM is optional and announced in Extended capabilities. Only a sub-set of the commands may support SM (see CLA-Definition in the command section). SM is designed as a replacement of the Admin-PW (PW3) and can be used to change data objects online (helpful for company cards, admin = company; user = employee). It is highly recommended to use SM for security related commands (e.g. VERIFY) if the OpenPGP application runs in contactless mode, because all data on the interface can be traced easily within a range of several meters.

SM is defined for the algorithms Triple-DES (168 bit key) or AES (128 bit key). The supported algorithm is announced in Extended capabilities, the related keys are stores in the application DOs 'SM Key-ENC' and 'SM Key-MAC'. By default the content of the DOs is set to zero and SM will not work. The keys can be set with a PUT DATA command after correct presentation of the Admin-PW (PW3). Form that point on all related commands can be used with SM. A command with correct SM has the same access condition to data in the application than a VERIFY with PW3.

If existing keys are replaced with PUT DATA, the card shall use the new values for the calculation of the response. The deletion of SM-keys should be done simultaneous with the DO 'F4', otherwise the functionality of SM is undefined. A key is deleted by setting all bits to zero.

EN 14890 uses a SSC (Send Sequence Counter) as first data block for MACing. This value is derived during a device authentication process. There is no definition in the EN how the SSC value is generated in case of static keys and/or without authentication.

For the OpenPGP application SSC is defined as follows:
A command GET CHALLENGE with 8 bytes for 3DES or 16 bytes for AES sets the SSC value in the card with the returned random. The card should reject a SM-command if SSC is not set properly. SSC is used for MACing only, any encryption is done without SSC. In compliance with EN 14890 'SSC + 1' is used for the next command APDU and 'SSC + 2' is used for the response. Several SM-commands can be send by using the automatically incremented SSC, but it is allowed to set a new SSC with a GET CHALLENGE command at any time.

Rules for SM usage:

- For a command with even INS, any command data is encrypted and capsulated in a Tag 87 with padding indicator (01).
- For a command with odd INS, any command data is encrypted and capsulated in a Tag 85 without padding indicator.
- Any response data are encrypted and capsulated in a Tag 87 with padding indicator (01).
- Commands with response (Le field not empty) have a protected Le-field (Tag 97) in the command data.
- Commands with response have a protected processing status (Tag 99) in the response.

- Commands without response (e.g. Case 3) have no processing status and/or MAC in the response field (SW1SW2 only).
- All SM-commands have at least an unencrypted MAC (Tag 8E) in the command data field.
- Encryption is done in CBC-Mode without SSC and with ICV = 00...00.
- MACing is done with SSC and ICV = 00...00, for 3DES Retail-MAC-Mode and for AES CMAC-Mode is used.

7.6 Logical Channels

The OpenPGP application does not use logical channels in this version. Channel number zero is assumed for all commands.

7.7 Command Chaining

If command data are too long for a single command (e.g. PSO:DECRYPT with RSA 2048 and support of Short length only) a card may provide command chaining. The feature is announced in card capabilities in the Historical bytes.

If chaining is used, the CLA of a command shall be set to the appropriate value and the command data consist of the first block of the complete command data. The card stores the data block internally and wait for a follow-up command with the same INS/P1/P2. If sufficient the next data block is concatenated by the card, up to an equal command with no chaining bit in the CLA. Then the command is executed with the whole data from all previous commands in this chain.

The length of a data field in a command with chaining bit in CLA should be equal to the maximum input buffer of the card announced in Extended capabilities.

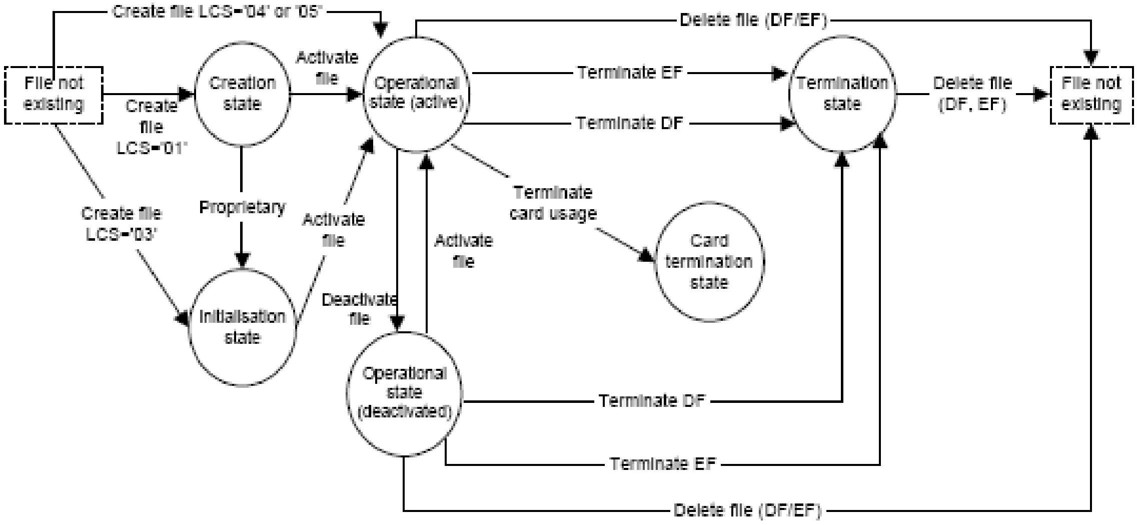
Chaining can be combined with Secure Messaging, the complete length of a single command shall not exceed the maximum input buffer of the card announced in Extended capabilities.

Commands that may support chaining have an appropriate coding in their CLA definition.

7.8 Life Cycle Management

Many users of an OpenPGP card asked me if it would be possible to RESET a card, if the Admin-PW was forgotten and the error counter of PW3 is blocked. ISO 7816 offers no command directly for that purpose, but it can be done with a behaviour of the card called Life Cycle Status (LCS). A card can have a life cycle from 'creation state' to 'initialisation state' to 'operational state' (that can be activated or deactivated) up to 'termination state'. ISO defines two types of LCS: primary state, that allows only transitions in one direction and is irreversible and secondary state, that is application specific and reversible.

The following figure from ISO 7816-9 demonstrates it:



The solution for the OpenPGP application uses an application specific secondary state, that has some additions to the defined behaviour in ISO. In the phase from initialisation to operational state a card normally gets its data. The ISO command ACTIVATE FILE (related to a DF) can set this state. So the solution for the OpenPGP application is to use the ACTIVATE FILE command to reset all values in the card to their default values (user DOs empty, PWs to default values, keys/fingerprints empty etc.).

ISO defines no direct way from operational to initialisation state. The solution for the secondary state in the OpenPGP application is a transition of the termination state. After termination, the OpenPGP application falls back to initialisation state. Then the card can be resetted (re-newed) by an ACTIVATE FILE command.

Because this behaviour is proprietary and cannot be implemented on all existing platforms, it is optional and announced in the Life Cycle Status indicator in the Historical bytes.

7.9 Status Bytes

After a command the chip returns a pair of status bytes (return code). All codings of ISO 7816-4 are valid for the card and may occur in a specific context.

The following table shows possible coding for status bytes:

SW1	SW2	Description
61	xx	Command correct, xx bytes available in response (normally used under T=0 or for commands under any protocol with long response data that cannot be transmitted in one response)
62	85	Selected file in termination state
65	81	Memory failure
67	00	Wrong length (Lc and/or Le)
68	81	Logical channel not supported
68	82	Secure messaging not supported
68	83	Last command of the chain expected
68	84	Command chaining not supported
69	82	Security status not satisfied PW wrong PW not checked (command not allowed) Secure messaging incorrect (checksum and/or cryptogram)
69	83	Authentication method blocked PW blocked (error counter zero)
69	85	Condition of use not satisfied
69	87	Expected SM data objects missing (e.g. SM-key, SSC)
69	88	SM data objects incorrect (e. g. wrong TLV-structure in command data)
6A	80	Incorrect parameters in the data field
6A	82	File or application not found
6A	88	Referenced data not found
6B	00	Wrong parameters P1-P2
6D	00	Instruction (INS) not supported
6E	00	Class (CLA) not supported
90	00	Command correct

8 Literature

European Standard (2009):
EN 14890-1, Application Interface for smart cards used as secure signature creation devices, Part 1: Basic services

European Standard (2009):
EN 14890-2, Application Interface for smart cards used as secure signature creation devices, Part 2: Additional services

ISO/IEC (2006):
ISO/IEC 7816-3, Identification cards - Integrated circuit cards - Part 3: Cards with contacts: Electrical interface and transmission protocols

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

ISO/IEC (2004):
ISO/IEC 7816-6, Identification cards - Integrated circuit cards - Part 6: Interindustry data elements for interchange

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

ISO/IEC (2004):
ISO/IEC 7816-9, Identification cards - Integrated circuit cards, Part 9: Commands for card management

RSA Laboratories (2002):
PKCS #1 v2.1: RSA Encryption Standard

The Internet Society (2007):
RFC 4880: OpenPGP Message Format

9 Flow Charts

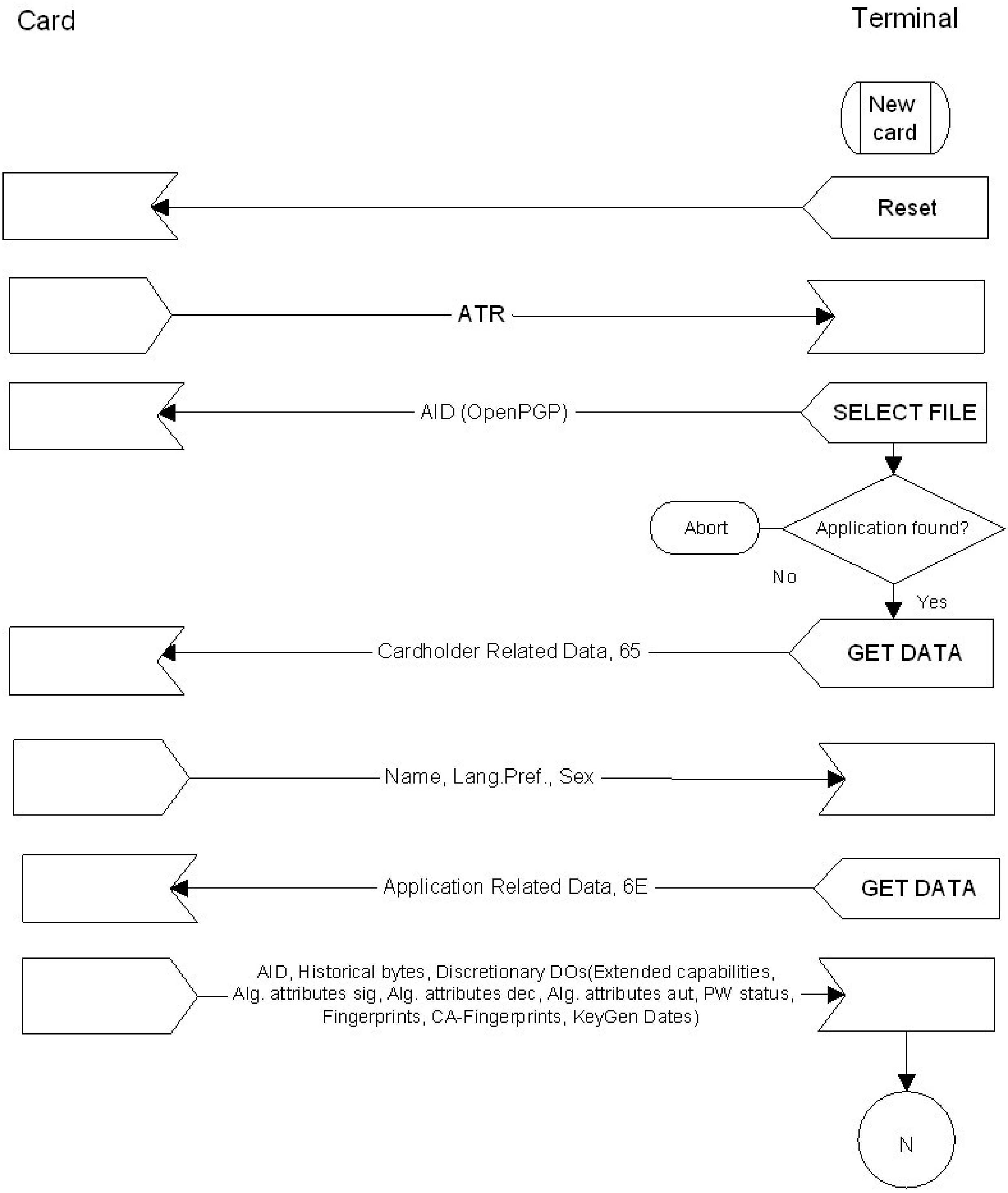
The communication scenarios illustrate some possibilities for the use of the OpenPGP application. Only a few functions are described, there are several additional functions available.

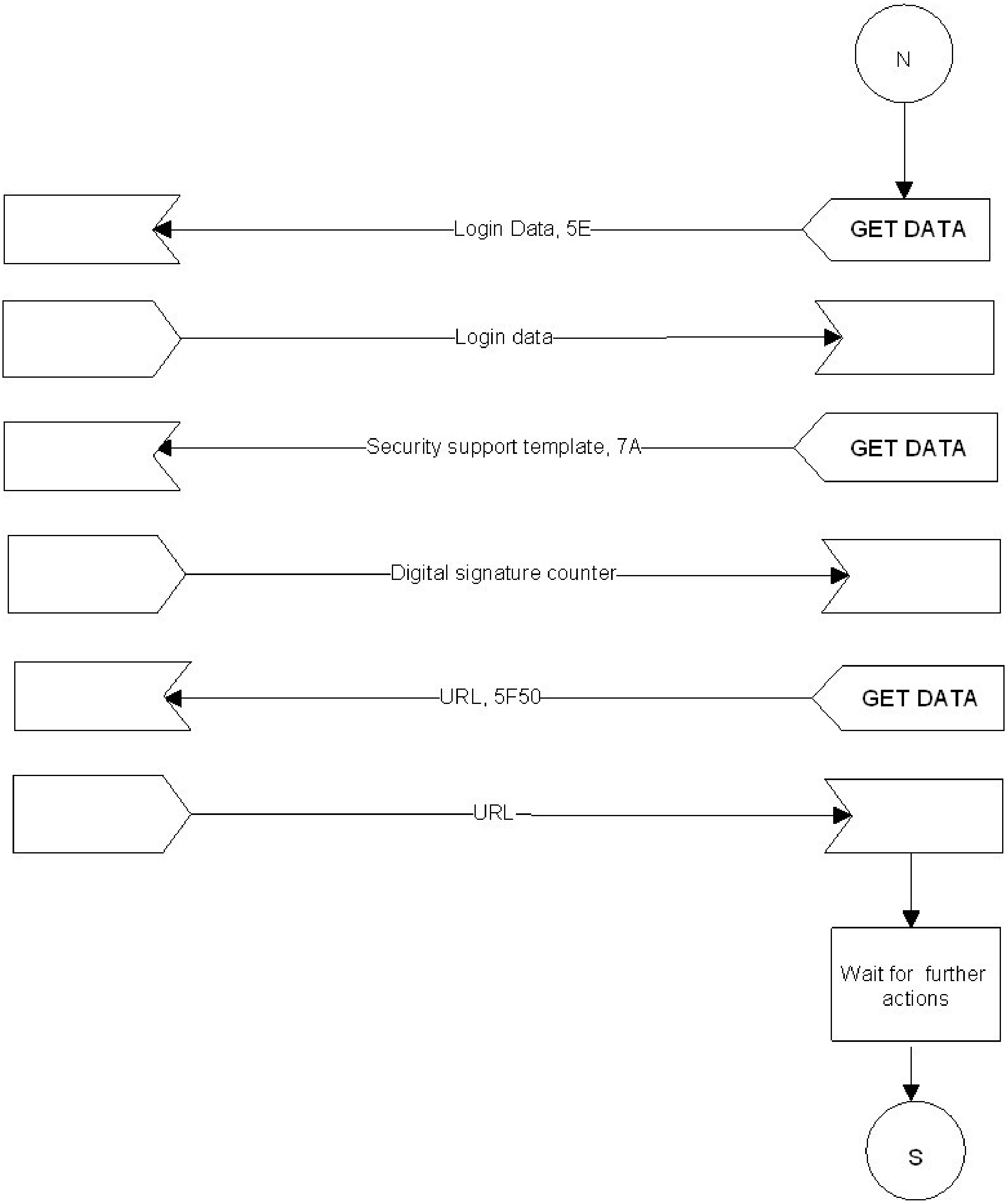
In principle, the application sequences to be realised apply to the application structure described in the specification. The realisation of the application sequences is generally made possible by the global commands provided to the card by the operating system, taking account of the security structure.

With respect to the sequences, only those application data are considered that are relevant at the interface between card and terminal. Standard return codes, header information and error events are not included for reasons of clarity. The scenarios are intended to clarify the essential mechanisms of the application and are used to facilitate a better understanding of the entire specification. They are not intended to serve as the only basis for the realisation of terminal programs.

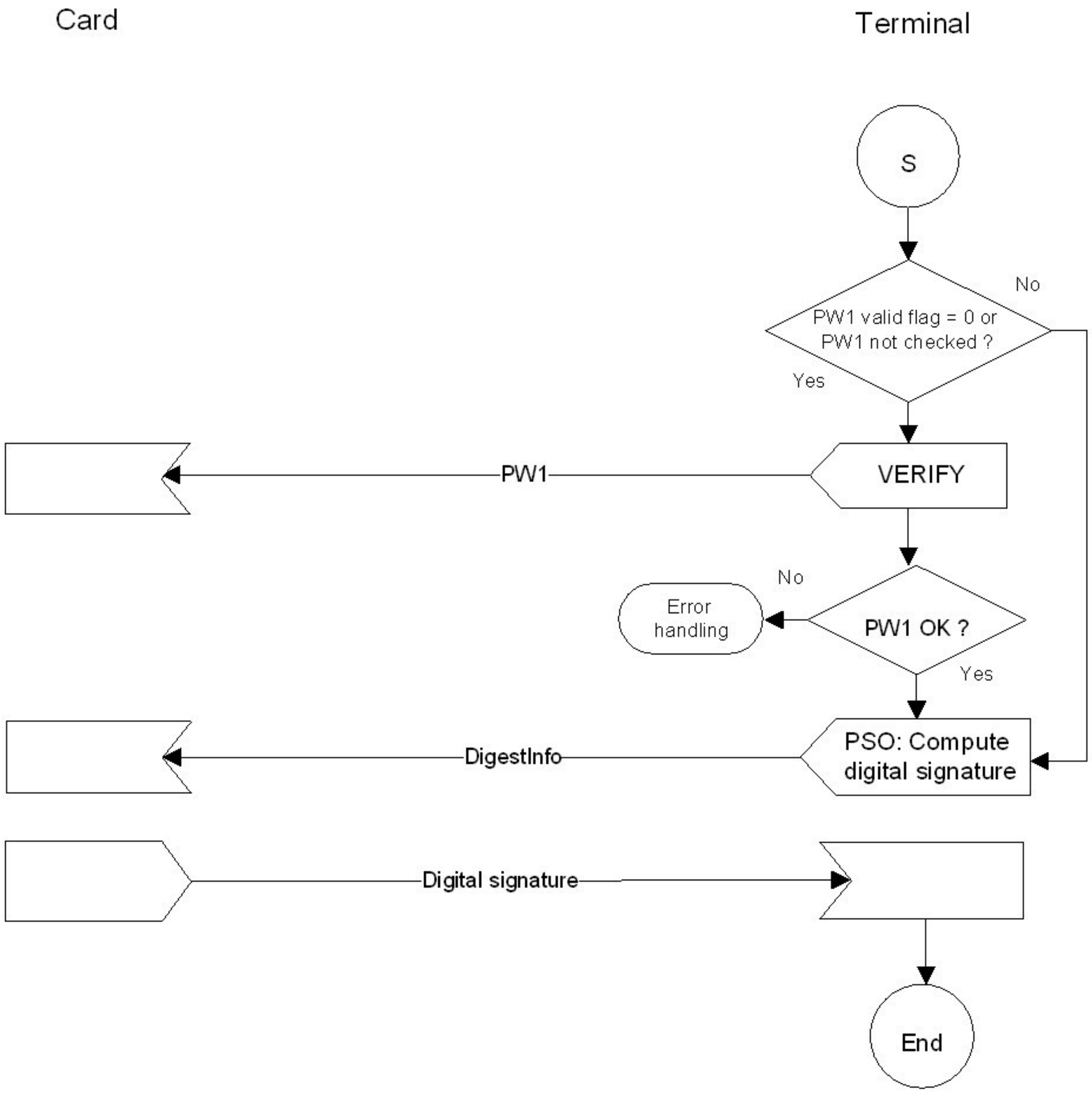
As long as the security guidelines required by the applications are observed, the modification of the following scenarios is possible.

9.1 Application Selection

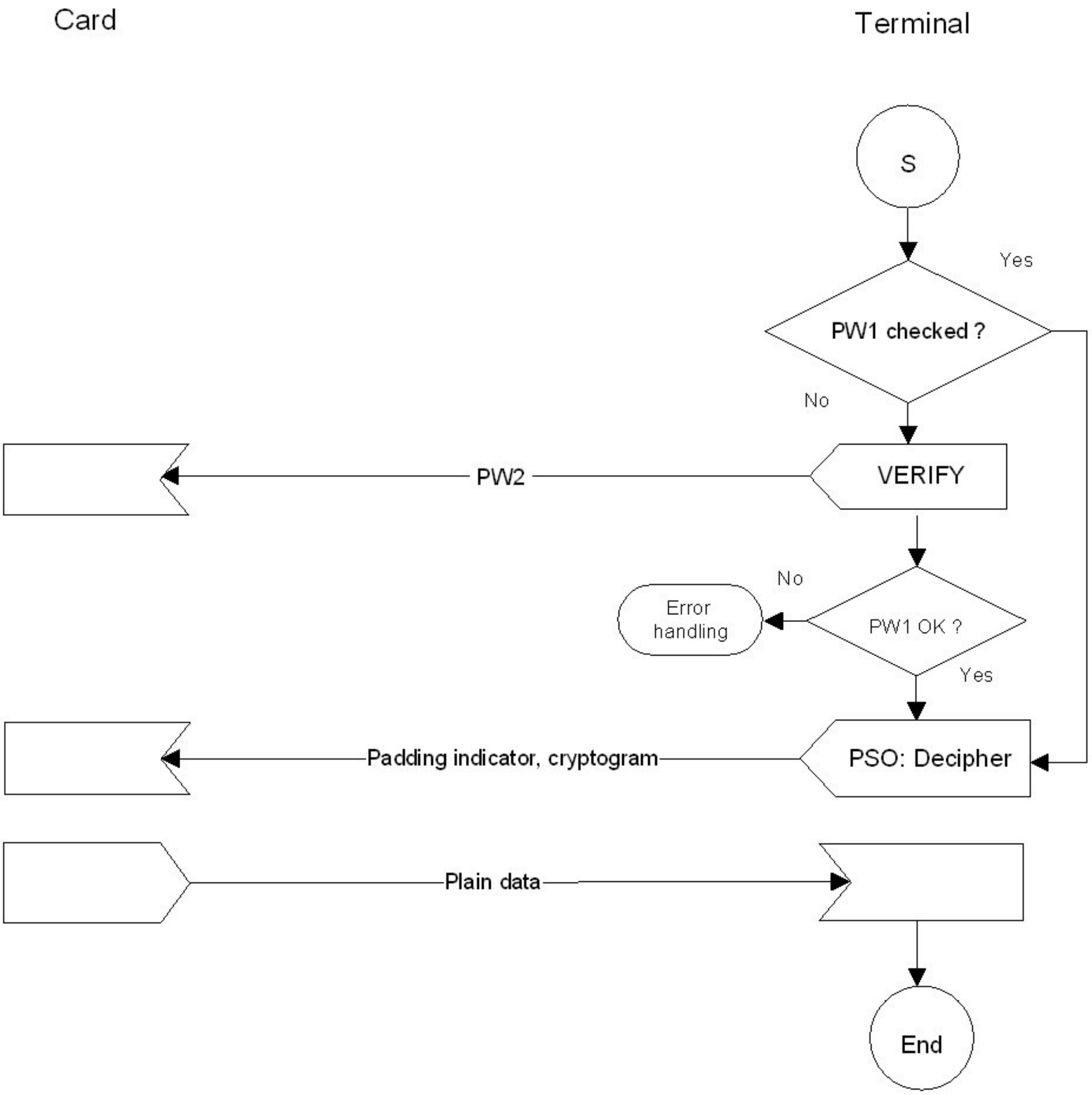

Card
Terminal
New card
Reset
ATR
AID (OpenPGP)
SELECT FILE
Application found?
Abort
No
Yes
GET DATA
Cardholder Related Data, 65
Name, Lang.Pref., Sex
Application Related Data, 6E
GET DATA
AID, Historical bytes, Discretionary DOs(Extended capabilities, Alg. attributes sig, Alg. attributes dec, Alg. attributes aut, PW status, Fingerprints, CA-Fingerprints, KeyGen Dates)
N


N
GET DATA
Login Data, 5E
Login data
Security support template, 7A
GET DATA
Digital signature counter
URL, 5F50
GET DATA
URL
Wait for further actions
S

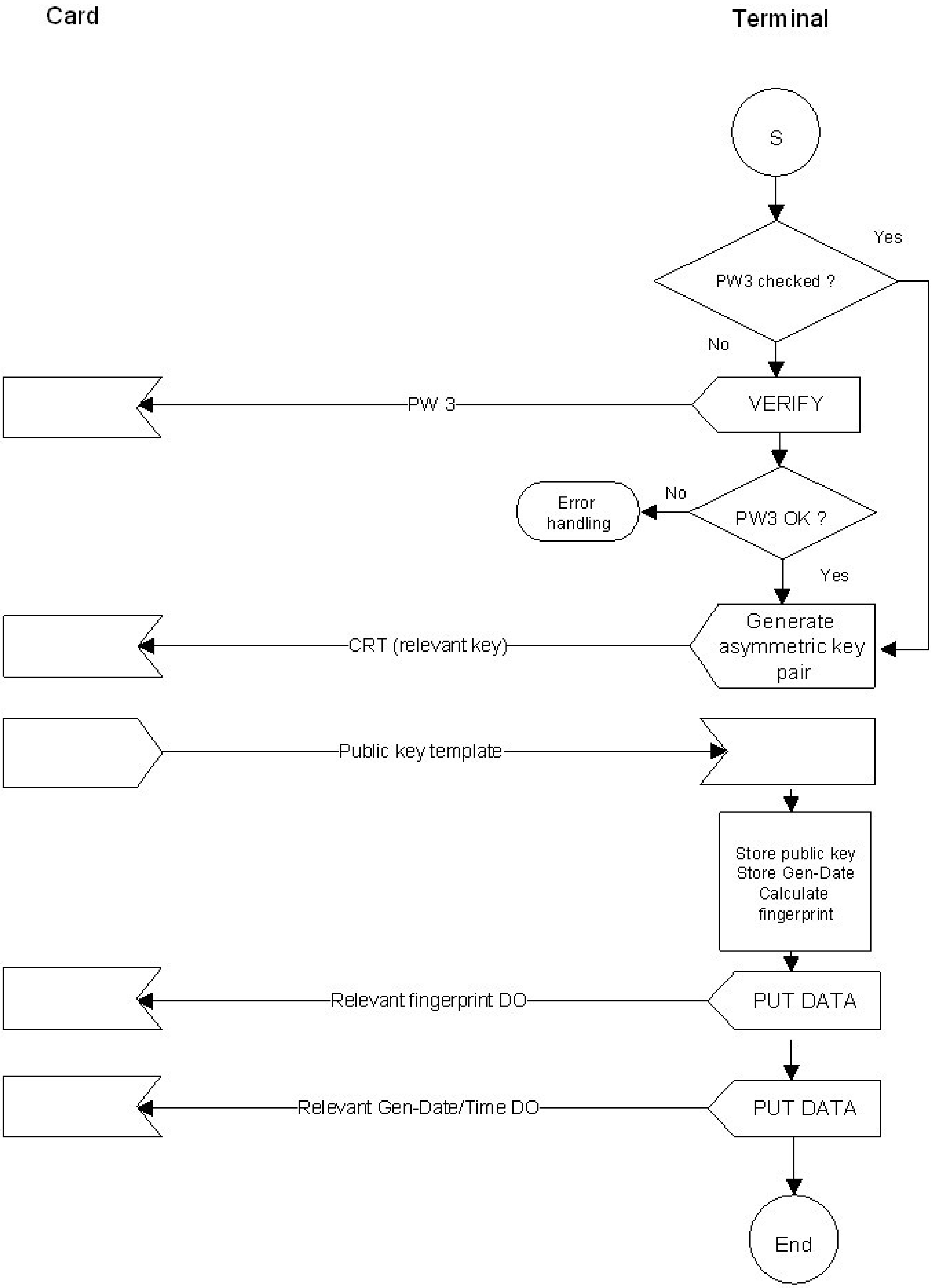
9.2 Compute Digital Signature



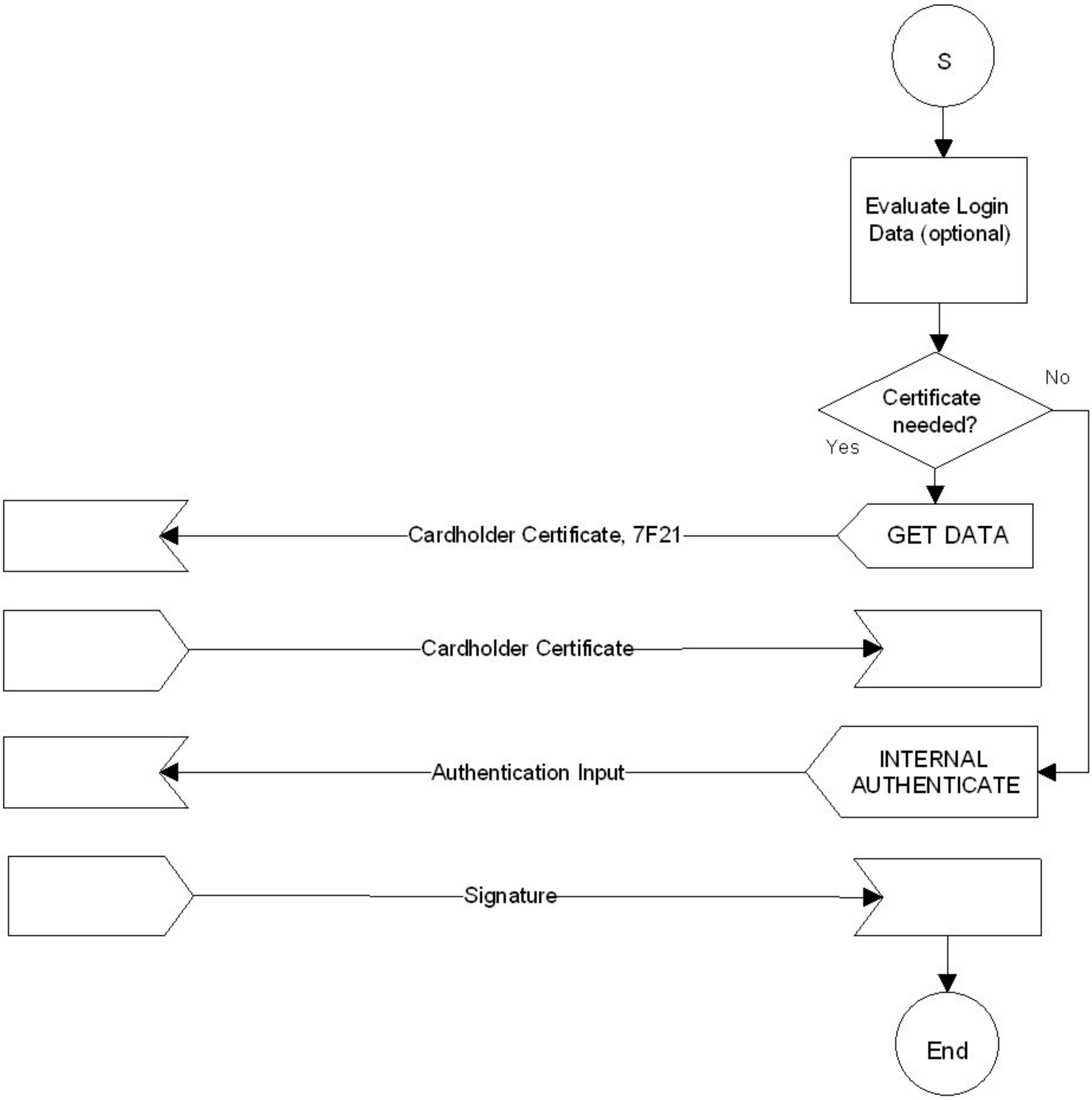
9.3 Decrypt Message



9.4 Generate Private Key



9.5 Client/Server Authentication


S
Evaluate Login Data (optional)
Certificate needed?
No
Yes
GET DATA
Cardholder Certificate, 7F21
Cardholder Certificate
INTERNAL AUTHENTICATE
Authentication Input
Signature
End