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

Version 1.1

Author: Achim Pietig

PPC Card Systems GmbH

September 27, 2004

Author:

Achim Pietig

PPC Card Systems GmbH
Senefelderstr. 10
33100 Paderborn
Germany
Email: a.pietig@ppc-card.de
achim@pietig.com

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 “CHV Status Bytes” (C4) mandatory for GET DATA as single object.

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.

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
CHV	Card Holder Verification
CLA	CLAss byte
DEC	DECipher
dec.	Decimal
DF	Dedicated File
DO	Data Object
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
FCP	File Control Parameter
INS	INStruction byte
MF	Master File
OS	Operating System
PK	Public Key
RFU	Reserved for Future Use
RSA	Rivest-Shamir-Adleman
SE	Security Environment
SIG	SIGnature
SK	Secret Key
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format 8 (compatible with 7-bit US-ASCII for all charcters < 80)

2 General requirements

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

Card ->

- ATR fully according to ISO 7816-3.
  - The OpenPGP application does evaluate historical characters for ‘Card capabilities’.
- As single transmission protocol T=1 or T=0 (ISO 7816-3) is allowed.
  - T=1 is preferred (chaining support, extended Lc/Le).
- The card may support different transmission protocols.
  - ATR may show different transmission protocols (e.g. T=0 and T=1).
  - PPS selection should be supported for different protocols.
- High speed modes according to ISO 7816-3 (Fi/Di) are requested (maximum as possible for the chip).
  - Maximum values are given in ATR.
  - PPS (protocol parameter selection) should be supported.
- Extended Lc and Le fields are recommended.
  - The card shall announce this feature in ‘Card capabilities’.
  - If extended Lc/Le are not supported, the card shall support command chaining and/or GET RESPONSE for large data objects (if present).

Reader (informative) ->

- PC/SC driver shall be supported.
  - PC/SC should be available for several platforms (e.g. Win32, Linux, Macintosh)
- T=1 and T=0 shall be supported.
- High-Speed protocols should be supported (PPS).
- Extended Le/Lc should be supported.

2.1 Limitations to this version

This version of the OpenPGP application in the terminal and also in the card has some restrictions. Main reason is that actual cards and card readers (with PCSC) do not support all requirements.

Terminal:

- Extended Lc/Le may not be supported (short Lc/Le is used even if card supports extended)
- ECC and DSA are not supported (only RSA algorithm is used for all functions)

Card:

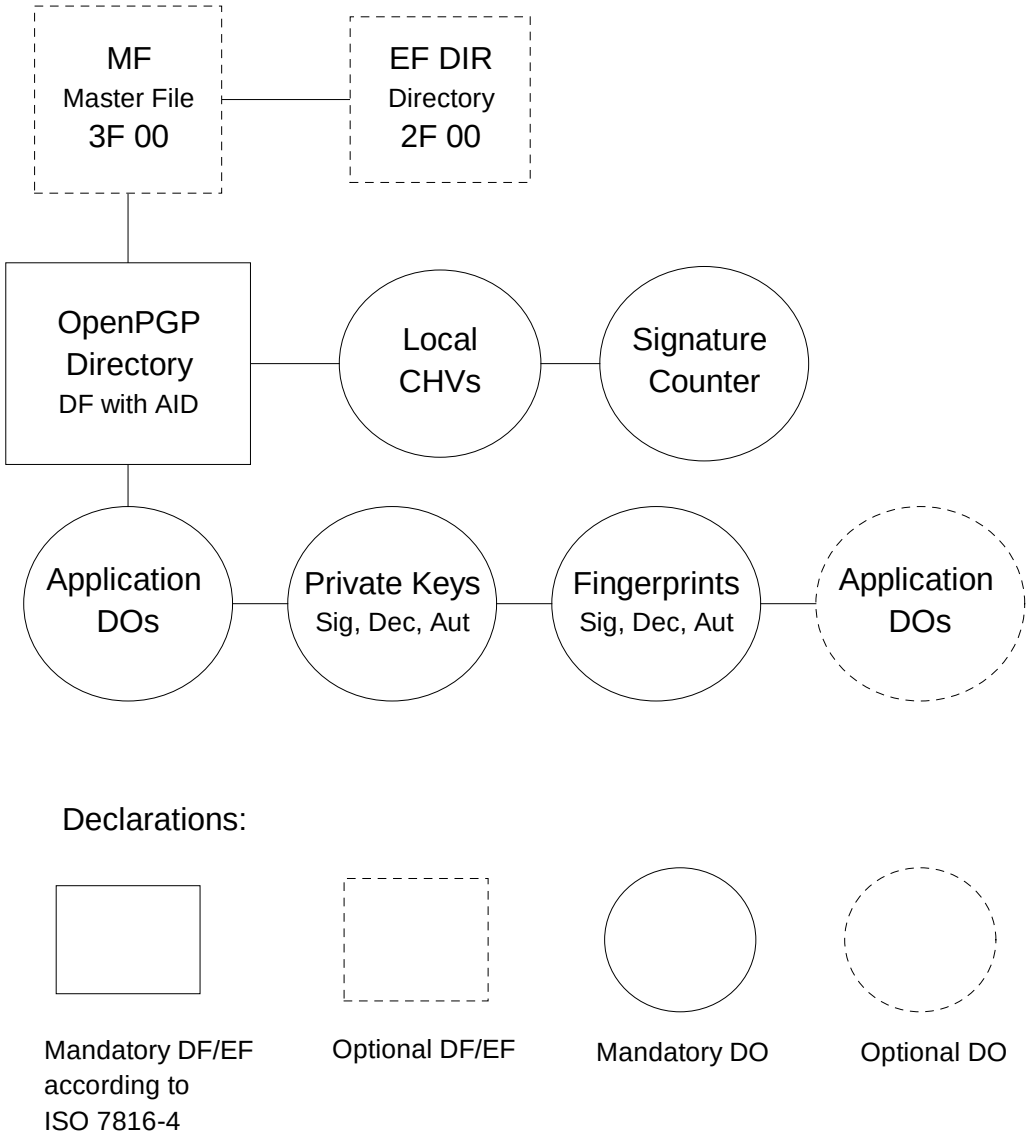
- High-Speed protocol may not be supported (terminal assumes standard values of ISO in that case)
- Extended Lc/Le may not be supported
- The card shall support RSA (minimum 1024 bit)

PCSC:

- High-Speed protocol may not be supported (terminal assumes standard values of ISO in that case)
- Extended Lc/Le may not be supported

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, CHV) 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).

4.1 Data files and objects in the MF or other DFs

The OpenPGP application may use global CHVs (Card Holder Verification). These data may be stored in the Master File (MF), any other Dedicated File (DF) or in a specific way of the current OS. In addition global information for all applications and specific keys may be stored in the MF, if present.

4.1.1 EF_DIR

This file under the MF (file identifier: ’2F00’) may, if present, 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. It may be used also for cards that do not support SELECT FILE with partial AID. The following entries should be added:

- 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

4.2 DF_OpenPGP

The application directory of the OpenPGP 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. The directory contains all data objects of the application with the exception of the data objects of other directories whose contents are accessed as well. This is an option, so it is possible to have all data objects in between the application directory. All given FCPs (File Control Parameter) are information for the implementation, but optional for presenting in a response of a SELECT FILE command. The OpenPGP application in the

terminal will not evaluate Tags in the FCP. The OpenPGP application retrieves the AID (DO with Tag 4F) with GET DATA in between the application related data (Tag 6E).

Possible File Control Parameter (FCP) of the DF_OpenPGP

Tag	Encoding	Description
83	Any	File ID
84	D27600012401xxxxxxxxxxxxxxxx0000	DF name (AID, Application IDentifier)

4.2.1 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 AID for the OpenPGP application 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 OpenPGP	Application	Version	Manu-facturer	Serial number	RFU

RID	Registered application provider identifier (unique identification of OpenPGP), ISO 7816-5
PIX	Proprietary application identifier extension (defined for OpenPGP application)
Application	Indication of the application
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

01 OpenPGP application (standard)

...

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. For that reason every card manufacturer or personalizer 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 from a manufacturer has a unique serial number. The manufacturer is responsible that no 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 starts with 00 00 00 01 for the first application of a manufacturer and is incremented automatically by him.

4.2.2 User authentication in the OpenPGP application

The OpenPGP application uses three local passwords for user authentication, called Card Holder Verification (CHV1 and CHV2 with 6 characters minimum, CHV3 with 8 characters minimum). The format of the CHVs is UTF-8 (case sensitive), the maximum length supported by the card for each CHV is declared in the ´CHV maximum length´ DO. Only the relevant bytes are used in the CHV commands, no fillers or paddings are added. The storage of the CHVs is dependent on the current OS. Global CHVs may be used but mapped to the application as local. CHV1 is used as access condition for the command PSO:CDS, CHV2 is used for PSO:DEC, INT-AUT, GET DATA and PUT DATA. The OpenPGP application uses CHV3 as resetting code for the RESET RETRY COUNTER command and as access condition for PUT DATA and GENERATE ASYMMETRIC KEY PAIR (generation of keys). All CHVs use an error counter with an initial value of 3. This error counter is readable with GET DATA. After correctly verifying the CHV, the access status of the corresponding CHV 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 personalization or CHV letter, then a default content is assumed: CHV1 and CHV2 = “123456” (6 bytes, 313233343536); CHV3 = “12345678” (8 bytes, 3132333435363738). It is highly recommended that the card holder changes this values. Because the usage of CHV1 and CHV2 is similar for the user, it is up to the terminal application to synchronise these values.

4.3 Data objects (DO)

To keep the interface to terminals simple and for the reason to transport the OpenPGP application to other OS easier, 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. All DOs should be defined as shareable and may be used by other applications as well. Simple DOs (S) return only the value with GET DATA. Constructed DOs (C) 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 are optional for retrieving with GET DATA as single DO, the OpenPGP application uses the ´normal´ DOs (mostly constructed) only. The order of DOs in a constructed DO may vary.

Tag	Format	Description
0101	S	Optional DO for private use, max. 254 bytes (binary, proprietary), can be used to store any information.
0102	S	Optional DO for private use, max. 254 bytes (binary, proprietary), can be used to store any information.
0103	S	Optional DO for private use, max. 254 bytes (binary, proprietary), can be used to store any information.
0104	S	Optional DO for private use, max. 254 bytes (binary, proprietary), can be used to store any information.
5E	S	Login data, max. 254 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).

Tag	Format	Description
5F50	S	Uniform resource locator (URL, as defined in RFC 1738), up to 254 bytes. The URL should contain a Link to a set of public keys in OpenPGP format, related to the card.
65	C	Cardholder Related Data (Tag)
5B	*S*	*Name (up to 39 bytes, 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 bytes (ISO 7816-4)*
5E	*S*	*Login data, max. 254 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). The DO is optional for Application Related Data, if not present the terminal can get it as single DO.*
73	*C*	*Discretionary data objects (Tag)*
C0	*S*	*Extended capabilities* *1 byte, Flaglist*
C1	*S*	*Algorithm attributes signature* *1 Byte Algorithm ID, according to RFC 2440* *further bytes depending on algorithm (e.g. length modulus and length exponent)*
C2	*S*	*Algorithm attributes decryption*
C3	*S*	*Algorithm attributes authentication*
C4	S	CHV Status Bytes (7 bytes, binary) 1st byte: 00 = CHV1 only valid for one PSO:CDS command 01 = CHV1 valid for several PSO:CDS commands 2nd byte: max. length for CHV1 3rd byte: max. length for CHV2 4th byte: max. length for CHV3 Byte 5, 6 and 7 (first byte for CHV1, second byte for CHV2 and third byte for CHV3): Usage counter of CHV1, CHV2 and CHV3. If 00 then the corresponding CHV 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*

Tag	Format	Description
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*
FF	C	Reading of all data objects of the application at once (e.g. 5F50 L DO 65 L DOs 6E L DOs 7A L DO) Mandatory only for cards with extended Lc/Le (see Card capabilities in ATR).

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 bytes (binary)
0102	S	Optional DO for private use, max. 254 bytes (binary)
0103	S	Optional DO for private use, max. 254 bytes (binary)
0104	S	Optional DO for private use, max. 254 bytes (binary)
5B	S	Name
5E	S	Login data
5F2D	S	Language preferences
5F35	S	Sex
5F50	S	Uniform resource locator (URL)

Tag	Format	Description
C4	S	Optional DO (announced in Extended capabilities). 1st CHV Status Byte (1 byte binary): 00 = CHV1 only valid for one PSO:CDS command 01 = CHV1 valid for several PSO:CDS commands
C7	S	Fingerprint (binary, 20 bytes) for signature key, format according to RFC 2440
C8	S	Fingerprint (binary, 20 bytes) for decryption key
C9	S	Fingerprint (binary, 20 bytes) for authentication key
CA	S	1st CA-Fingerprint in list (binary, 20 bytes)
CB	S	2nd CA-Fingerprint in list (binary, 20 bytes)
CC	S	3rd CA-Fingerprint in list (binary, 20 bytes)
CE	S	Generation date/time of signature key (4 bytes Big Endian, format according to RFC 2440
CF	S	Generation date/time of decryption key (4 bytes Big Endian)
D0	S	Generation date/time of authentication key (4 bytes Big Endian)
E0	C	Optional DO (announced in Extended capabilities). Private key template for signature (used for key import). Storing of a private key for signature sets the corresponding digital signature counter to zero (000000).
E1	C	Optional DO (announced in Extended capabilities). Private key template for decryption
E2	C	Optional DO (announced in Extended capabilities). Private key template for authentication

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, 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

With this table the card 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 Extended capabilities:

b8	B7	b6	b5	B4	B3	b2	b2	Meaning
1	-	-	-	-	-	-	-	Extended Lc value not supported. This flag is only valid (evaluated) if Extended Lc/Le fields are announced in Card capabilities (ATR). The card supports Extended Le values only (e.g. reading long DOs with GET DATA). In commands with Lc and Le present the format of Lc shall be Extended but the value shall not exceed the maximum Short value.
-	1	-	-	-	-	-	-	Support for GET CHALLENGE
-	-	1	-	-	-	-	-	Support for Key Import (DOs E9-EB available)
-	-	-	1	-	-	-	-	CHV Status byte changeable (DO C4 available for PUT DATA)
-	-	-	-	1	-	-	-	Support for Private use DOs (0101-0104)
-	-	-	-	-	0	0	0	RFU

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 of the card also and are related to the output format of the corresponding command.

RSA:

Byte	Length	Value
01	01	Algorithm ID (RFC 2440) 01 = RSA
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

This version defines only the content for the RSA algorithm.

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. Only mandatory (necessary) values are used. The function does not matter how the key is stored in the card internally. It is assumed that the card has the functionality to generate the internal values (e.g. key parts for Chinese Remainder Theorem by use of RSA) from the input. The function does not set the value of the corresponding fingerprint.

This version defines the input for RSA keys only. The order of the DOs is mandatory.

E0, E1 or E2	xx	Tag to indicate a private key data object (signature, decryption or authentication)	
xx = length	C0	xx	Public Exponent e
	C1	xx	Prime1 p
	C2	xx	Prime2 q

The length of the DOs 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 (Tag E0), 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	VERIFY of corresponding CHV	Card Holder Verification = password
RESET RETRY COUNTER	VERIFY of CHV3	
PUT DATA	Various	Depending on Data Objects
GENERATE ASYMMETRIC KEY PAIR	VERIFY of CHV3, Always	Generation with CHV
PSO: COMPUTE DIGITAL SIGNATURE	VERIFY of CHV1	
PSO: DECIPHER	VERIFY of CHV2	

Command	Access condition	Description
INTERNAL AUTHENTICATE	VERIFY of CHV2	
GET CHALLENGE	Always	
Other commands	Never	Exceptions may be commands for personalization

Access conditions for Data Objects:

Data Object	READ	WRITE	Description
Private use (0101)	Always	Verify CHV2	
Private use (0102)	Always	Verify CHV3	
Private use (0103)	Verify CHV2	Verify CHV2	
Private use (0104)	Verify CHV3	Verify CHV3	
Login data (5E)	Always	Verify CHV3	
URL (5F50)	Always	Verify CHV3	
Name (5B)	Always	Verify CHV3	
Language preference (5F2D)	Always	Verify CHV3	
Sex (5F35)	Always	Verify CHV3	
AID (4F)	Always	Never	Writing possible only during personalization (manufacturer)
Extended capabilities (C0)	Always	Never	Writing possible only during personalization
Algorithm attributes (C1 – C3)	Always	Never	Writing possible only during personalization
CHV Status bytes (C4)	Always	Verify CHV3	Only 1st byte can be changed, other bytes only during personalization
Fingerprints (C5, C7 – C9)	Always	Verify CHV3	
CA-Fingerprints (C6, CA – CC)	Always	Verify CHV3	
Generation date/time of key pairs (CD – D0)	Always	Verify CHV3	
DS-Counter (93)	Always	Never	
Private key signature (E0)	Never	Verify CHV3	
Private key decryption (E1)	Never	Verify CHV3	
Private key authentication (E2)	Never	Verify CHV3	

6 Historical bytes (ATR)

After receiving the ATR (Answer To Reset) from a card, the ´format byte (T0)´ indicates the presence of Historical characters in bits 1- 4, according to ISO 7816-3. For the OpenPGP application the presence of a DO Card capabilities is relevant. It may be found in a coding of the Historicals according to ISO 7816-4.

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.

- 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).
- If the first Historical byte is set to '80', then the remaining Historical bytes consist of optional consecutive COMPACT-TLV data objects; the last data object may carry a status indicator of one, two or three bytes.
- If the category indicator byte is set to '10', then the subsequent byte is the DIR data reference. The bytes after the reference are coded in COMPACT-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.

If the Historicals are not present or not in ISO format or if Tag /Length ´73´ is not found, then the application assumes that the card does support short Lc/Le only.

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 and -8. 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 CHV1, CHV2 or CHV3
CHANGE REFERENCE DATA	24	01	81/82/ 83	Change of CHV1, CHV2 or CHV3 (new reference data)
RESET RETRY COUNTER	2C	02	81/82	Resets retry counter of CHV1 or CHV2 and sets new value for CHV1 or CHV2. In the command data the new CHV is present.
PUT DATA	DA	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

Command	INS	P1	P2	Comment
				the card, digital signature in response
DECIPHER	2A	80	86	Input: Padding indicator byte (always 00) and encrypted data, length of encrypted data must 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 or 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
ENVELOPE	C2	00	00	Only used under T=0 (e.g. commands with extended Lc/Le)

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 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 (FCP) 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	FCP or empty
SW1-SW2	9000 or specific status bytes

7.2.2 VERIFY

With this command one of the CHVs of the application is verified.

Command:

CLA	00
INS	20
P1	00
P2	81 (CHV1) or 82 (CHV2) or 83 (CHV3)
Lc	xx (min. 06 for CHV1/2, min. 08 for CHV3, max. see DO ´C4´)
Data field	Corresponding CHV
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 CHVs of the application can be changed. The command needs a previous VERIFY for the CHV to change.

Command:

CLA	00
INS	24
P1	01
P2	81 (CHV1) or 82 (CHV2) or 83 (CHV3)
Lc	xx (min. 06 for CHV1/2, min. 08 for CHV3, max. see DO ´C4´)
Data Field	New CHV
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 CHV1 or CHV2 can be reset, 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 CHV3 only.

Command:

CLA	00
INS	2C
P1	02
P2	81 (CHV1) or 82 (CHV2)
Lc	xx (min. 06 for CHV1/2, max. see DO ´C4´)
Data field	New CHV
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 capsulated 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 card works with short Le and the data exceeds the maximum length of a response, 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
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 (simple is provided only) 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 CHV3 (except DO 0101 and DO 0103 after correct verification of CHV2).

Command:

CLA	00
INS	DA
P1	xx (00 if Tag has a length of one byte only)
P2	xx
Lc	xx
Data Field	Addressed data
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 blocks within the command GET DATA.

Command:

CLA	00
INS	C0
P1	00
P2	00
Lc	Empty
Data field	Empty
Le	00

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 (DSA or ECC) or the DSI (Digital Signature Input for RSA) 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 CHV1. The command internally checks the CHV Status byte DO (first byte), if the value is 00, then the CHV1 is reset and has to be verified again for following command.

Command:

CLA	00
INS	2A
P1	9E
P2	9A
Lc	Length of subsequent data field (23 for RSA, 14 for DSA/ECC)
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 is generated 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: ASN.1-Sequence of digestAlgorithm and digest

The DigestInfo to be delivered to the card when using this signature format has the following coding:

a) SHA-1 with OID: { 1 3 14 3 2 26 }
   DigestInfo: 3021 3009 06052B0E03021A 0500 0414 || hash value (20 bytes)
b) RIPEMD-160 with OID: { 1 3 36 3 2 1 }
   DigestInfo: 3021 3009 06052B24030201 0500 0414 || hash value (20 bytes)

DigestInfo for DSA:
The DSI consists of the hash value calculated using SHA-1 or RIPEMD-160 (20 bytes, dec.).

DigestInfo for Elliptic Curves:
The DSI consists of the hash value which was calculated using SHA-1 or RIPEMD-160 (20 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.

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 CHV2 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. Triple-DES).

The card is not involved in the encipherment 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.

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.

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) followed by cryptogram
Le	00

Response:

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

Input in case of ECC:

Due to the fact that ECC does not support en- and decryption directly, a special variant with use of ECC is needed for this purpose. In this version of the OpenPGP application ECC decryption is not defined and will be added in a later version. ECC decryption is defined for example in ANSI X9.63 and Elliptic Curve Integrated Encryption Scheme (ECIES), which was proposed by Abdalla, Bellare and Rogaway.

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 a DSI compliant to PKCS#1, the card does an internally padding and calculates a signature with the corresponding secret key for authentication. The mechanism can be used for example with Secure Shell (SSH) or SSL/TLS. The command can be used after correct presentation of CHV2 only.

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, Lc <= 51 for 1024 bit modulus; for DSA/ECC: Hash value with SHA-1 or RIPEMD-160, Lc = 20 (values are decimal)
Le	00

Response:

Data field	Authentication data
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.

7.2.11 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. 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 CHV3 for the generation of a key pair. Reading of a public key is always possible.

Command:

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

Response:

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

Defined CRTs for command (generation of key pair or reading of public key):

Digital signature:

B6 00

Confidentiality:

B8 00 (only valid for RSA and ECC)

Authentication:

A4 00

Defined DOs for response:

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. 41)

Set of public key data objects for DSA

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, e.g., 20 bytes)

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)

Set of public key data objects for ECC

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 bytes)

7.2.12 GET CHALLENGE

This optional command (announced in Extended Capabilities) generates a random number of any length. It is a service to the terminal application because smart cards often generate high sophisticated random numbers by certified hardware.

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

7.3 Command usage under different I/O protocols

The OpenPGP application uses T=1 protocol (ISO 7816-3) as standard protocol. 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 readers PC/SC driver.

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 for all standard commands (last or only command of a chain)

7.5 Secure messaging (SM)

The OpenPGP application does not use secure messaging in this version.

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 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 (partial):

SW1	SW2	Description
61	xx	Command correct, xx bytes available in response (normally used under T=0 or for GET DATA with short Le and long DOs under any protocol)
65	81	EEPROM failure
67	00	Wrong length (Lc)
69	82	CHV wrong CHV not checked yet (command not allowed)
69	83	CHV blocked (error counter zero)
69	85	Condition of use not satisfied
6A	80	Incorrect parameters in the data field
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

DIN (2000):
DIN V66291-1 (Prenorm): Chipcards with digital signature application/function according to SigG and SigV, Part 1: Application Interface, Version 1.0

ISO/IEC (2004):
ISO/IEC CD 7816-3, Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols

ISO/IEC (2004):
ISO/IEC FDIS 7816-4, Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange

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

ISO/IEC (2004):
ISO/IEC DIS 7816-8, Identification cards - Integrated circuit(s) cards with contacts, Part 8: Interindustry commands for a cryptographic toolbox

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

TeleTrusT Deutschland e.V. (2000):
German Office Identity Card, (Elektronischer Dienstausweis), Version 1.0

The Internet Society (1998):
RFC 2440: OpenPGP Message Format

9 Flow Charts

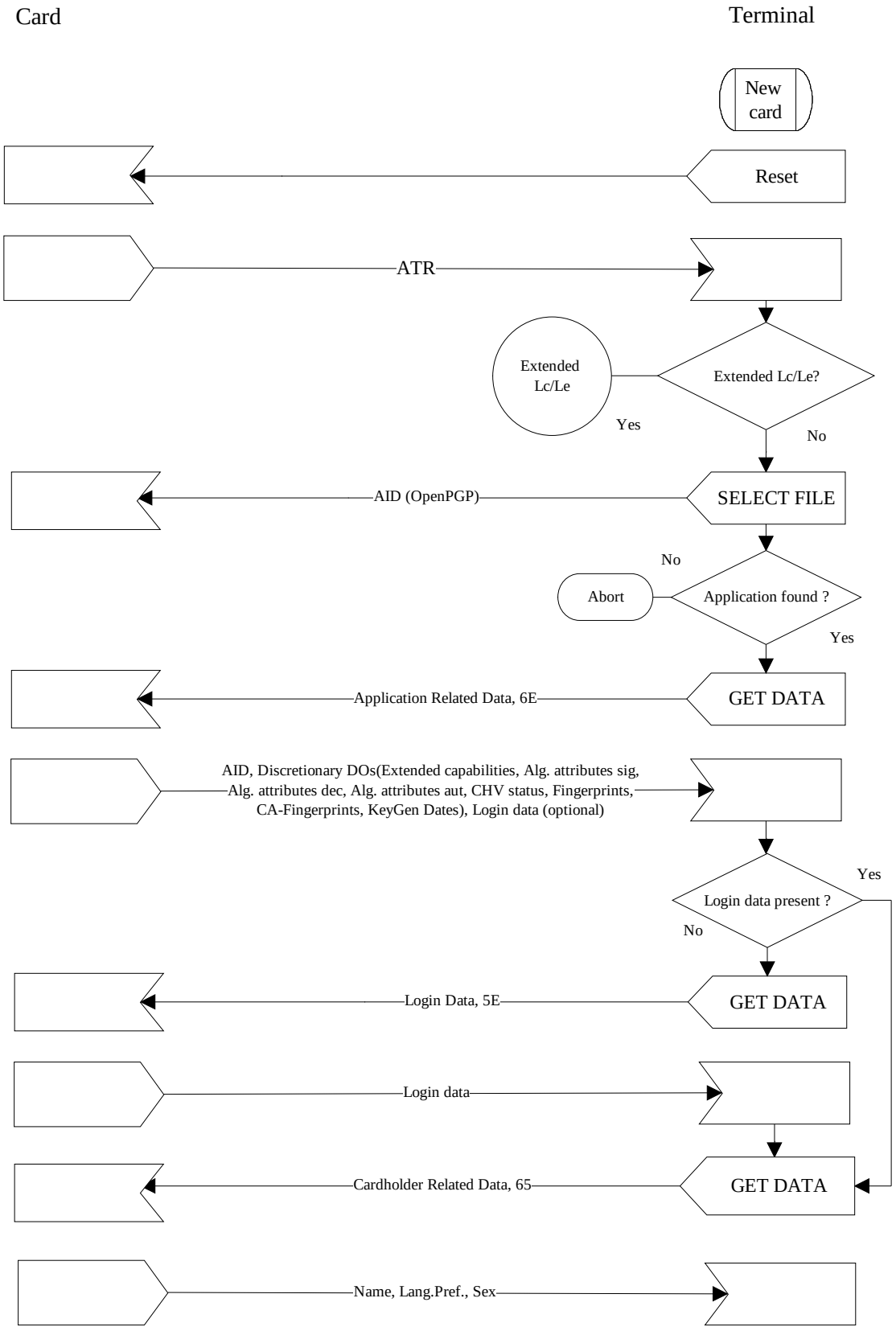
The communication scenarios illustrate some possibilities for the use of the OpenPGP application. Only a few functions are described, there are several additional functions available.

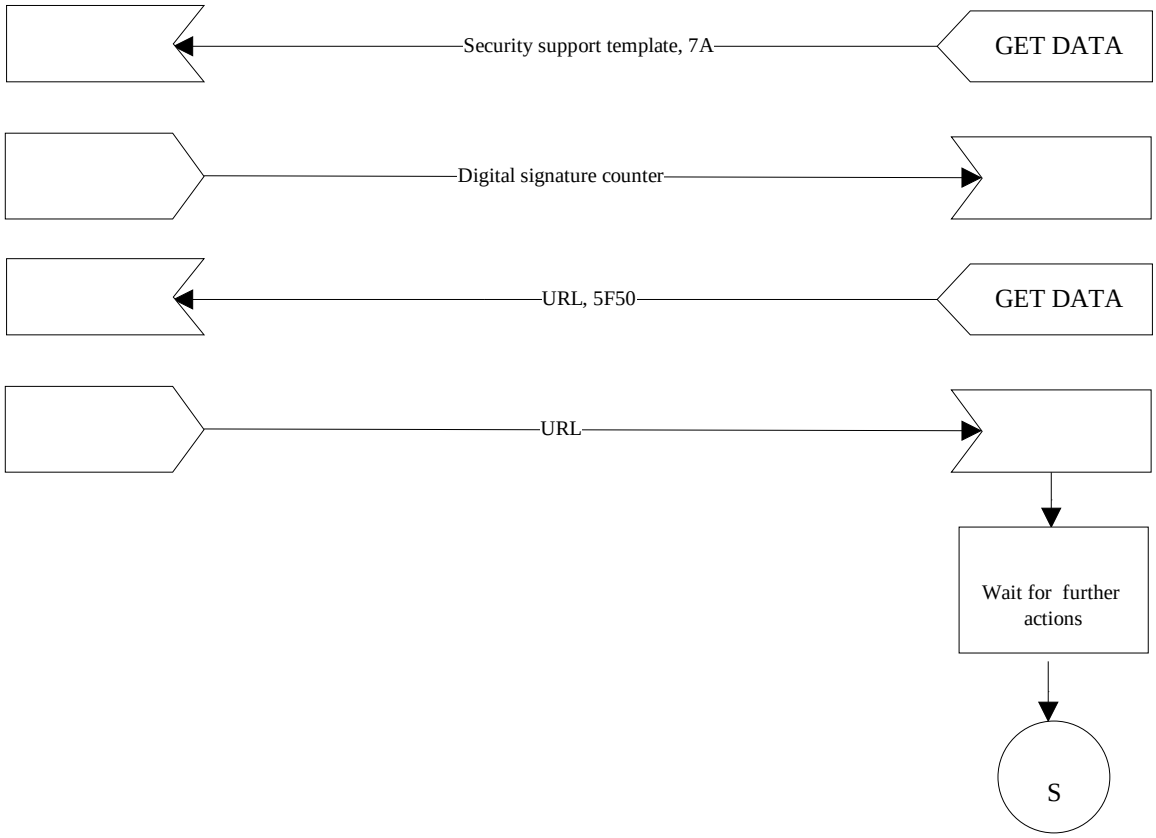
In principle, the application sequences to be realised apply to the application structure described in the specification. The realisation of the application sequences is generally made possible by the global commands provided to the card by the operating system, taking account of the security structure.

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.

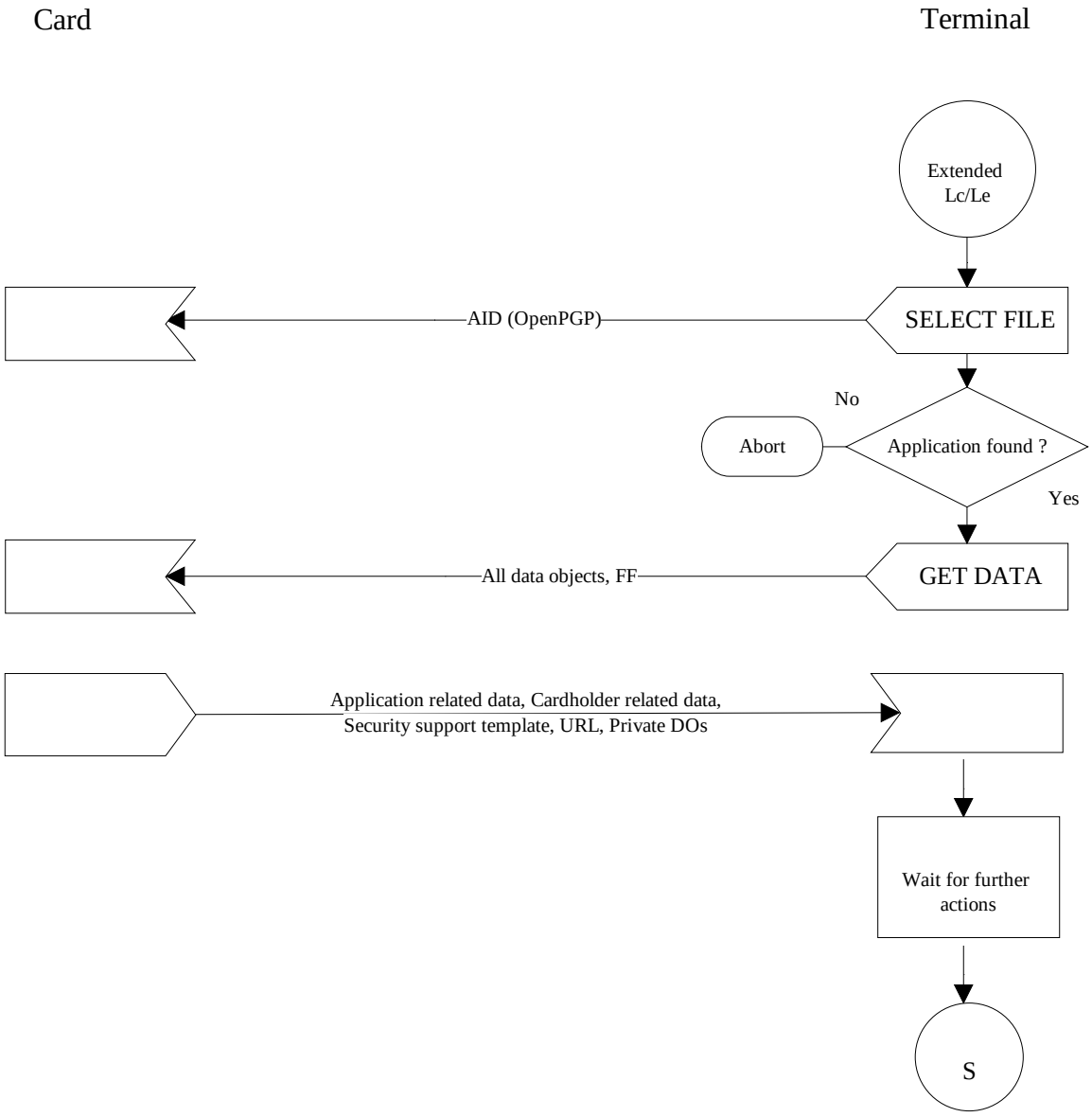
As long as the security guidelines required by the applications are observed, the modification of the following scenarios is possible.

9.1 Application Start for cards with Short Lc/Le

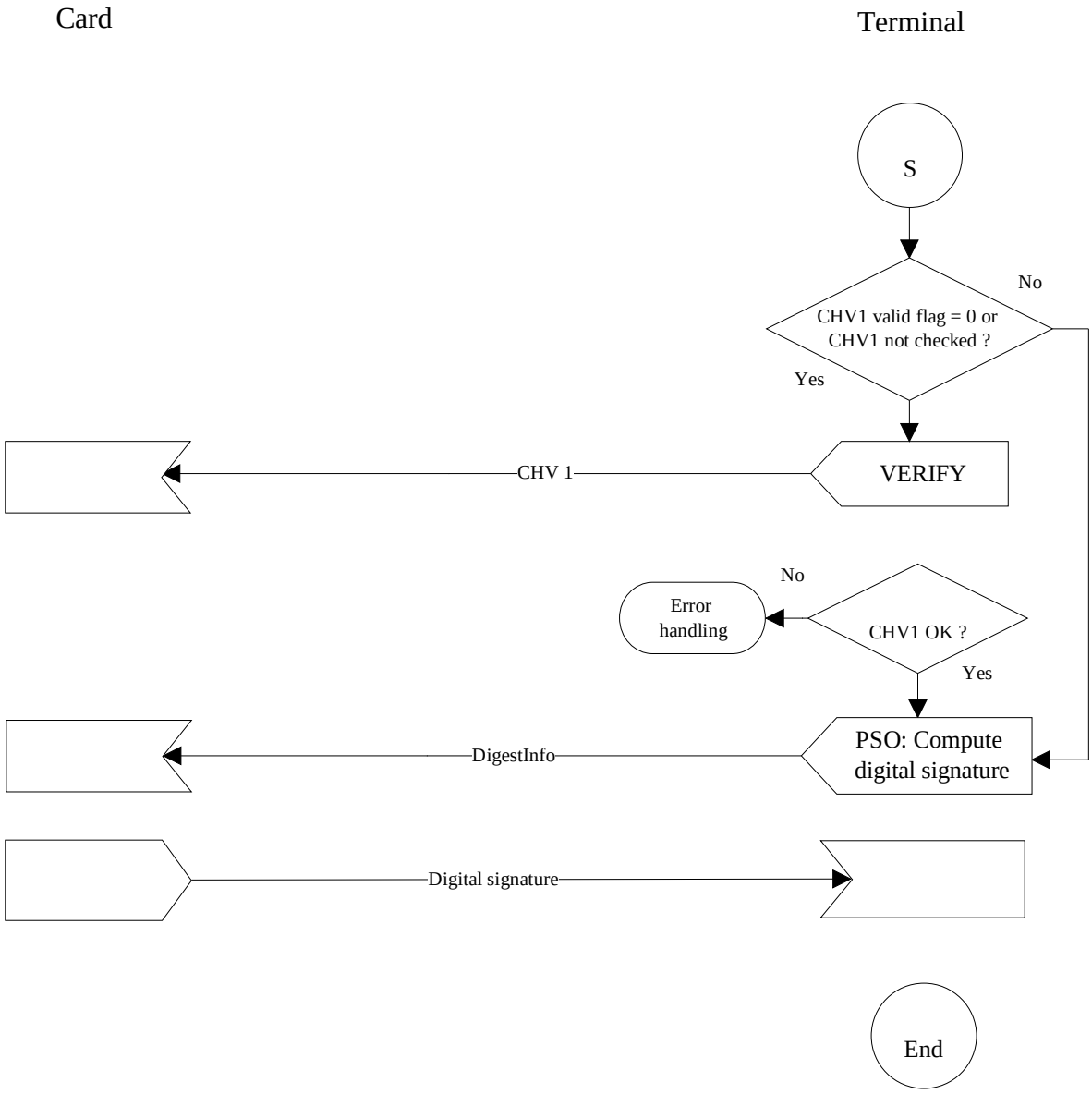



Security support template, 7A
GET DATA
Digital signature counter
URL, 5F50
GET DATA
URL
Wait for further actions
S

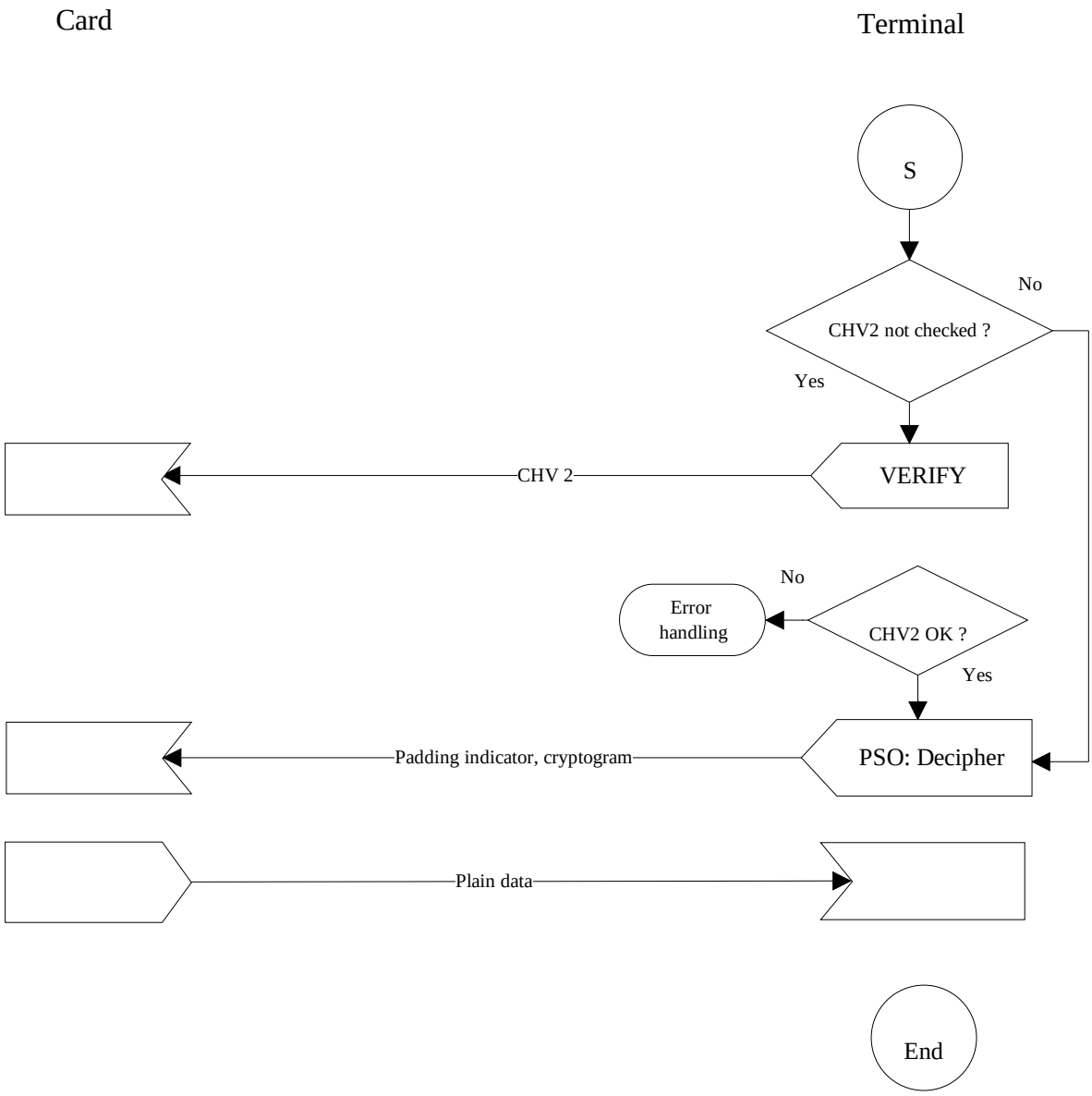
9.2 Application Start for cards with Extended Lc/Le



9.3 Compute digital signature



9.4 Decrypt message



9.5 Generate private key