INTERNATIONAL STANDARD

ISO/IEC 24727-2

First edition
2008-10-01

Identification cards — Integrated circuit card programming interfaces —

Part 2: Generic card interface

Cartes d'identification — Interfaces programmables de cartes à puce —

Partie 2: Interface de carte générique

Reference number
ISO/IEC 24727-2:2008(E)

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

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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

ISO/IEC 24727-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Cards and personal identification*.

ISO/IEC 24727 consists of the following parts, under the general title *Identification cards — Integrated circuit card programming interfaces*:

— *Part 1: Architecture*

— *Part 2: Generic card interface*

— *Part 3: Application interface*

— *Part 4: API administration*

The following parts are under preparation:

— *Part 5: Testing*

— *Part 6: Registration authority procedures for the authentication protocols for interoperability*

Introduction

ISO/IEC 24727 defines interoperable programming interfaces to integrated circuit cards. Programming interfaces are defined for all card lifecycle stages and for use with integrated circuit cards.

ISO/IEC 24727 is written with sufficient detail and completeness that independent implementations of each part are interchangeable and can interoperate with independent implementations of the other parts.

This part of ISO/IEC 24727 specifies a command-level programming interface to contactless integrated circuit cards and cards with contacts that is a concretization of the concepts, data structures and commands found in the following documents:

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

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

— ISO/IEC 7816-9, *Identification cards — Integrated circuit cards — Part 9: Commands for card management*

— ISO/IEC 7816-15, *Identification cards — Integrated circuit cards — Part 15: Cryptographic information application*

— ISO/IEC 20060, *Information technology — Open Terminal Architecture (OTA) specification — Virtual machine specification*

The commands and data objects described in this part of ISO/IEC 24727 are consistent with the commands and data objects found in these documents which will be referred to as the base documents.

This part of ISO/IEC 24727 maximizes the fungibility of independent realizations of its prescriptions. This property of this part of ISO/IEC 24727 is realized by positing a minimally sufficient subset of the base standards which realizes their core functionality through the minimization of the number of options provided.

Identification cards — Integrated circuit card programming interfaces —

Part 2:
Generic card interface

1 Scope

This part of ISO/IEC 24727 defines a generic card interface for integrated circuit cards. This interface is presented as:

— command-response pairs for interoperability,

— card and application capability description and determination.

This part of ISO/IEC 24727 is based on ISO/IEC 7816-4, ISO/IEC 7816-8, ISO/IEC 7816-9, and ISO/IEC 7816-15.

2 Normative references

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

ISO/IEC 24727-1, *Identification cards — Integrated circuit card programming interfaces — Part 1: Architecture*

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

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

ISO/IEC 7816-9, *Identification cards — Integrated circuit cards — Part 9: Commands for card management*

ISO/IEC 7816-15, *Identification cards — Integrated circuit cards — Part 15: Cryptographic information application*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 24727-1 and the following apply.

3.1
data object
information seen at the interface consisting of the concatenation of a mandatory ISO/IEC 8825 DER-encoded tag field, a mandatory ISO/IEC 8825 DER-encoded length field and a conditional value field

3.2
file
structure for application and/or data in the card, as seen at the generic card interface when processing commands

3.3
translation code
procedural software that transforms commands on the generic card interface to commands implemented on an integrated circuit card

4 Abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO/IEC 24727-1 and the following apply.

ATS	answer to select, as defined in ISO/IEC 14443-3
DF	dedicated file
DO	data object
FCP	file control parameters
FID	file identifier
RFU	reserved for further use

5 Organization for interoperability

This clause specifies a subset of the structure, commands and data structure defined in ISO/IEC 7816-4, ISO/IEC 7816-8 and ISO/IEC 7816-9.

The following can not be specified at the generic card interface:

- short file identifiers;
- logical channels;
- files with record structure.

The physical card mapped to the generic card interface by the translation code may use a short EF identifier, logical channels, and record structure files.

5.1 Command-response pairs for interoperability

5.1.1 Command and response encoding

Requests at the GCI are logically equivalent to command APDUs as specified in ISO/IEC 7816-4, ISO/IEC 7816-8 and ISO/IEC 7816-9.

Confirmations at the GCI are logically equivalent to response APDUs as specified in ISO/IEC 7816-4, ISO/IEC 7816-8 and ISO/IEC 7816-9.

The following interface may be used to send a generic card interface command directly to an implementation of this part of ISO/IEC 24727:

sequence-of-bytes ExecuteCommand(sequence-of-bytes command)

This interface sends a command to the ISO/IEC 24727-2 implementation and returns as its value the response of the ISO/IEC 24727-2 implementation.

Further interfaces may be defined in other parts of ISO/IEC 24727.

5.1.2 Class byte

Table 1 lists the class byte values that shall be used in commands on the generic card interface.

Table 1 – CLA Values on the GCI

b8	b7	b6	b5	b4	b3	b2	b1	Description
0	-	-	0	-	-	-	-	The command is the last or only command of a chain
0	-	-	1	-	-	-	-	The command is not the last command of a chain
1	1	1	1	1	1	1	1	The command is for the Part 2 implementation

This part of ISO/IEC 24727 shall support command chaining only for the transmission of data strings too long for a single command; i.e. constant INS, P1 and P2 across all commands in the chain.

For transmission of requests acted upon by the ISO/IEC 24727-2 implementation, generally without transmission of APDUs to the card, CLA = 'FF' shall be used.

5.1.3 Instruction byte

Tables 2 and 3 list the instruction byte values that should be used in commands at the GCI as these commands guarantee the standardized independence of the ISO/IEC 24727-2 and ISO/IEC 24727-3 implementations.

A GCI request with an INS not found in Table 2 shall be sent directly to the card and the card-interface response shall be returned to the entity having made the GCI request.

Commands with instruction bytes listed in Table 3 shall be acted on by the ISO/IEC 24727-2 implementation and shall not be provided to the translation script.

Table 2 – Requests on the GCI Handled by the Translation Script

Command Name	INS	Package	Limitations
SELECT	‘A4’	A	SELECT by file identifier (P1-P2 = ’00-04’ or ’00-0C’) and SELECT by DF name (P1-P2 = ’04-04’ or ’04-0C’) with return of FCP data object or no data shall be supported. (See Note)
READ BINARY	‘B0’	A	Bit 8 of P1 shall be set to 0.
READ BINARY	‘B1’	A	P1 and P2 shall be set to ‘00’.
UPDATE BINARY	‘D6’	A	Bit 8 of P1 shall be set to 0.
UPDATE BINARY	‘D7’	A	P1 and P2 shall be set to ‘00’.
GET DATA	‘CA’ ‘CB’	A	None.
PUT DATA	‘DA’ ‘DB’	A	When PUT DATA references a data object that already exists it shall be overwritten.
GENERATE ASYMMETRIC KEY PAIR	‘46’ ‘47’	B	Out of scope
VERIFY	‘20’	A	P2 is not zero.
VERIFY	‘21’	A	P2 is not zero.
CHANGE REFERENCE DATA	‘24’	A	None.

GET CHALLENGE	‘84’	A	None.
INTERNAL AUTHENTICATE	‘88’	A	None.
EXTERNAL AUTHENTICATE	‘82’	A	None.
MUTUAL AUTHENTICATE	‘82’	A	None.
GENERAL AUTHENTICATE	’86’ ‘87’	A	None.
PERFORM SECURITY OPERATION: COMPUTE DIGITAL SIGNATURE	‘2A’	A	P1=’9E’ P2=’9A’ Command data field: - Absent (hash value provided via PERFORM SECURITY OPERATION:HASH
PERFORM SECURITY OPERATION: VERIFY DIGITAL SIGNATURE	‘2A’	A	P1=’00’ P2=’A8’ Command data field: - DO ‘9E’
PERFORM SECURITY OPERATION: HASH	‘2A’	A	P1=’90’ P2=’80’ or ’9A’ Command data field: 1) - DO ‘90’ (intermediate hash value || amount of bits already hashed) || DO ‘80’ (final text block) or 2)- DO ‘90’ hash value
PERFORM SECURITY OPERATION:VERIFY CERTIFICATE	‘2A’	A	P1-‘00’ P2=’AE’ or ‘BE’ Command data field: - DO ‘7F21’ (card verifiable certificate)
PERFORM SECURITY OPERATION: ENCIPHER	‘2A’	A	P1=’86’ P2=’80’ Command data field: data to be enciphered
PERFORM SECURITY OPERATION: DECIPHER	‘2A’	A	P1=’80’ P2=’86’ Command data field: data to be deciphered (PI || cryptogram)
MANAGE SECURITY ENVIRONMENT	‘22’	A	SET (P1=’x1’) and RESTORE (P1=’F3’)
CREATE FILE	‘E0’	B	Only FCP data objects in Table 9 are supported. The created file becomes the current file.
DELETE FILE	‘E4’	B	Only P1-P2 = ’00-00’ is supported. After deletion of the file the parent of the deleted file becomes the currently selected dedicated file.
ACTIVATE FILE	‘44’	B	Only P1-P2 = ‘00-00’ is supported
DEACTIVATE FILE	‘04’	B	Only P1-P2 = ‘00-00’ is supported
RESET RETRY COUNTER	'2C'	A	None
GET RESPONSE	'C0'	A	Only P1-P2 = '00-00' is supported The status word 6985 means there are no data to retrieve

Note: In the case of SELECT by DF name with return of the FCP (P1-P2=’04-04’), a returned FCP data object may contain a data object with tag ‘87’ indicating the elementary file that contains the card-application capability description.

Table 3 – INS Values on the GCI Acted on by the ISO/IEC 24727-2 Implementation (CLA='FF')

Command Name	INS	P1 P2	Package	Limitations
COLD RESET	'00'	'0000'	A	Lc absent, Le = '00'.
WARM RESET	'00'	'00FF'	A	
DEACTIVATE CONTACTS	'00'	'0100'	A	Lc and Le absent
DEACTIVATE CONTACTS AND EJECT	'00'	'0200'	A	
SELECT PROCEDURAL ELEMENT	'A4'	'0400'	A	Lc in the range 5..16, Le absent. A standard data field shall be an AID containing the OID of an ISO standard defining the implementation, according to ISO/IEC 7816-4. A data field proprietary to the implementation shall start with 'FX' .
GET DATA	CA		A	Unless the DOs to be transmitted have application-class tags defined in ISO/IEC 7816 or ISO/IEC 24727, the tags shall be of the context-dependent class. When PUT DATA references a data object that already exists, it shall be overwritten. Particular tags in a PUT DATA may trigger the execution of a procedure by the called element. If there is more than one parameter to transmit to the procedure, those parameters shall be transmitted within a constructed DO. According to clause 5.2, the status code '0000' indicates proper execution of the procedure.
PUT DATA	DA		A	
LIST READERS	CA	'7F64'	A	Lc absent, Le = '00' Returns the value of DO 7F64. This value is a concatenation of DOs encapsulating reader names in UTF8 format.

The physical card mapped to the generic card interface by the translation code may use other ISO/IEC 7816 compliant commands.

Instruction values received at the GCI including those in Table 2 may trigger a procedural element in a capability description. See 6.3.

Package A shall be required for operational use. Packages A and B shall be required for card management use.

A successful completion of the RESET command shall reset both the ISO/IEC 24727-2 implementation and the card. The reset of the ISO/IEC 24727-2 implementation shall include setting the CCD and all the ACDs to 'undefined'.

The response data in the R-APDU of the RESET C-APDU shall be the historical bytes of the card ATR, ATS or answer to ATTRIB if they are available. The status words shall be '0000' for successful completion and otherwise '0F00'.

5.1.4 File descriptor byte

Table 4 lists the file descriptor byte values that shall be used in the FCP on the GCI. Files seen on the GCI are not shareable.

Table 4 – File Descriptor Byte Values on the GCI

b8	b7	b6	b5	b4	b3	b2	b1	Description
0	0	1	1	1	0	0	0	Dedicated file
0	0	0	0	0	0	0	1	Working elementary file, transparent structure
0	0	1	1	1	0	0	1	Working elementary file, TLV structure for BER-TLV data objects

5.2 Card states for interoperability

Tables 5 and 6 list the states that shall be used in implementations of the generic card interface and describes the state transition events between these states.

Table 5 – Card and Application States and State Transition Events on the GCI

State Name	Always Defined	Type of State Change	State Transition Event
Currently selected application	Yes	**Set**	SELECT by DF name; e.g., application identifier
Currently selected dedicated file	Yes	**Set**	SELECT by file identifier of a dedicated file
		Set	CREATE FILE with the new dedicated file becoming the currently selected dedicated file
		Set	DELETE FILE of the currently selected dedicated file with the new dedicated file becoming the parent of the deleted dedicated file
Currently selected elementary file	No	**Set**	SELECT by file identifier of an elementary file
		Set	CREATE FILE with the newly created elementary file becoming the currently selected elementary file
		Unset	SELECT by DF name
		Unset	SELECT by file identifier of a dedicated file
		Unset	DELETE FILE of the currently selected elementary file or currently selected dedicated file
		Unset	CREATE FILE of a dedicated file

Table 6 – Currently selected files after the successful execution of commands on the GCI

COMMAND	Current application/DF	Current elementary file
SELECT by DF name	Change to the specified application/DF	Cleared and non existent
SELECT DF by file identifier	Change to the specified DF	Cleared and non existent
SELECT EF by file identifier	No change	Change to specified EF
CREATE FILE of DF	Change to the specified DF	Cleared and non existent
CREATE FILE of EF	No change	Change to the specified EF
DELETE FILE of DF	Change to the parent DF in the case of deletion of the currently selected DF	Cleared and non existent in the case of deletion of the DF that have currently selected EF
DELETE FILE of EF	No change	Clear and non existent
Immediately after the RESET command, the currently selected application/DF shall be MF or the default selected application/DF. The currently selected EF is “cleared and non-existent”.		

5.3 Status words for interoperability

The status words that shall be used on the generic card interface are listed in Table 7.

Table 7 – Status Words for Interoperability

	Symbol	Value	Meaning
Normal	OK	'9000'	Successful completion of command
	MORE	'61xx'	Successful completion of command with at least xx bytes of additional response data available
Warning	EOP-NOCHANGE	'62xx'	Unexpected end of processing leaving the state of non-volatile memory unchanged from its state immediately before the start of the execution of the command.
	EOD	'6282'	End of data reached
	EOP-RC	'63Cx'	Wrong reference data – x tries left
	EOP-CHANGED	'63xx' other than '63Cx'	Unexpected end of processing leaving the state of non-volatile memory changed from its state immediately before the start of the execution of the command.
Execution Error	ABORT-NO CHANGE	'64xx'	End of processing due to error condition leaving the state of non-volatile memory unchanged from its state immediately before the start of the execution of the command.
	ABORT-CHANGED	'65xx'	End of processing due to error leaving the state of non-volatile memory changed from its state immediately before the start of the execution of the command.
	ABORT-SECURITY	'66xx'	End of processing due to error condition involving a security condition leaving the non-volatile memory in an undefined state.
Checking Error	WRONG LENGTH	'6700'	Wrong length
	SECURITY CONDITION	'6982'	Security condition not satisfied
	REFERENCE DATA BLOCKED	'6983'	Reference data is blocked
	CONDITION OF USE	'6985'	Conditions of use not satisfied
	DATA FIELD	'6A80'	Incorrect parameters in the command data field
	FUNCTION NOT SUPPORTED	'6A81'	Function not supported; e.g. no additional logical channels available
	FILE NOT FOUND	'6A82'	File or application not found
	P1-P2	'6A86'	Incorrect parameters P1-P2
	DATA NOT FOUND	'6A88'	Referenced data not found
	BAD INS	'6D00'	Instruction is not supported or invalid
	BAD CLA	'6E00'	Class code is not supported
	UNDEFINED	'6F00'	No precise diagnosis
Response produced by Generic Card Access Layer	OK	'0000'	Successful processing by ISO/IEC 24727-2 implementation including CCD and ACD procedural elements
	SIGNATURE INVALID	'02xx'	Signature on translation script not verifiable
	EXCEPTION	'0080'	Response data contain a language-defined exception
	NOT MAPPED	'0A81'	No translation provided by ISO/IEC 24727-2 procedural element
	IFD NOT FOUND	'0A82'	Interface device not available
	CARD MISSING	'0A88'	Card not found
	UNDEFINED	'0F00'	No precise diagnosis

All SW1 SW2 values, returned at the generic card interface, with SW1 not equal to '6X' or '9X' do not originate from a response APDU from the card, but from the ISO/IEC 24727-2 middleware. '0X YZ' has the same meaning as '6X YZ' issued by a card for X>0. Values defined so far appear in Table 7. More values may be defined in further parts of this standard. For conformance to this part of the standard, status words from the card and from procedural elements which are not found in this table shall be mapped to ‘6F00’ and ‘0F00’ respectively.

All SW1s starting with '0', '1', '2', '3', '4', '5', '7', '8', 'A', 'B', 'C', 'D', 'E' are either defined in this standard or RFU by ISO/IEC JTC1/SC17.

The use of all SW1s starting with ‘F’ is proprietary.

5.4 Data structures for interoperability

Data structures for interoperability shall be implemented as files or BER-TLV data objects. These files and BER-TLV data objects are defined in ISO/IEC 7816-4.

Data objects may be managed using the GET DATA and PUT DATA commands on the generic card interface. Data objects may also be contained in and managed by card-applications. The security attributes of the data objects in an elementary file of TLV structure are contained in the FCP of the elementary file.

Each card-application shall be universally identified by an ISO/IEC 7816-4 application identifier.

Tables 8 through 13 describe the templates and tags that shall be used on the generic card interface.

More templates and data objects may be included in further parts of ISO/IEC 24727.

Table 8 – Interoperability Data Objects for Templates

Symbol	Tag	Description	Tag Class	Context
FCP	‘62’	Data file control parameter template	Application	Global
AT	'A4'	Control reference template for authentication	Context-Specific	Manage Security Environment and Perform Security Operation commands
CCT	'B4'	Control reference template for cryptographic checksum		
DST	'B6'	Control reference template for digital signature		
CT	'B8'	Control reference template for confidentiality		
HT	‘AA’	Hash template		

Table 9 – Interoperability Data Objects in the File Control Parameter Template (FCP)

Symbol	Tag	Description
SIZE	‘80’	Number of data bytes in the file excluding structural information
ALLOC	‘81’	Number of bytes allocated to the EF or the DF including structural information
FDB	‘82’	File descriptor byte (1 byte)
FID	‘83’	File identifier
DFNAME	'84'	Dedicated file name; generally, an application identifier
FID-ACD	‘87’	Identifier of an EF containing an extension of the file control information
SEC-EXP	'AB'	Security attribute template in expanded format
SEC-COM	‘8C’	Security attribute in compact format
SEC-DO	‘A0’	Security attribute template for data objects

In Table 9, if FID-ACD (tag '87') is present it shall be the file identifier of elementary file containing the capability description of the card-application

Table 10 – Interoperability Data Objects in the Authentication Control Reference Template (AT)

Symbol	Tag	Description
CM-REF	'80'	Cryptographic mechanism reference
SEC-KEY/PuKR	'83'	Key reference of a secret key or public key reference
SES-KEY/PrKR	'84'	Key reference of a session key or private key reference

Table 11 – Interoperability Data Objects in the Cryptographic Checksum Control Reference Template (CCT)

Symbol	Tag	Description
CM-REF	'80'	Cryptographic mechanism reference
SEC-KEY	'83'	Key reference of a secret key
SES-KEY	'84'	Key reference of a session key

Table 12 – Interoperability Data Objects in Digital Signature Control Reference Template (DST)

Symbol	Tag	Description
CM-REF	'80'	Cryptographic mechanism reference
PuKR	'83'	Public key reference
PrKR	'84'	Private key reference

Table 13 – Interoperability Data Objects in the Confidentiality Control Reference Template (CT)

Symbol	Tag	Description
CM-REF	'80'	Cryptographic mechanism reference
SEC-KEY/PuKR	'83'	Key reference of a secret key or public key reference
SES-KEY/PrKR	'84'	Key reference of a session key or private key reference

Additional data objects, particularly those pertaining to secure messaging, appear in other parts of ISO/IEC 24727.

5.5 Card-applications for interoperability

5.5.1 Alpha card-application

The card-application with application identifier 'E8 28 81 C1 17 02' shall be the alpha card-application. The alpha card application shall exist and be selectable on the GCI or emulated at the SAL layer.

The alpha card-application shall provide application-independent card information as defined in ISO/IEC 7816-4 such as card, file, and card-application management information.

5.5.2 Cryptographic information application

The cryptographic information application is defined in ISO/IEC 7816-15. If a cryptographic information application profile is present as indicated in the CCD (see below), then the cryptographic information application shall be selectable on the GCI.

An example of the implementation of the ISO/IEC 7816-15 cryptographic information application is shown in Annex B.

6 Capability descriptions

There are two types of capability descriptions, the card capability description (CCD) and an application capability description (ACD) for each card-application. When retrieved from the generic card interface the card capability description shall be retrieved under tag '7F62' and the application capability description shall be retrieved under tag '7F63'.

6.1 Card capability description (CCD)

The alpha card-application shall support retrieval of CCD data object (tag '7F62').

Table 14 lists the data objects that may be found in the CCD. These data objects shall apply to all ISO/IEC 24727-compliant applications on the card and may include a list of card-applications available on the card and procedural code mapping between the card's native commands and the commands described in Table 2.

Table 14 –Data Objects in the CCD ('7F62')

Symbol	Tag	Description	Mandatory/ Optional	Value	Note
PRO	'80'	Profile of ISO/IEC 24727-2 with which this CCD complies	Mandatory	'00'	Provided by the card
SAID	'A0'	Sequence of application identifiers of card-applications	Optional	Concatenation of data objects with tag '4F'	May be constructed by the ISO/IEC 24727-2 implementation based on information from the card
LANG	'A1'	Procedural element description template (See 6.3)	Optional	Data objects as defined in Table 16.	May be constructed by the ISO/IEC 24727-2 implementation based on information from the card
LANG-URL	'5F50'	URL of the code that performs the translation	Optional		
CIA-PROFILES	'81'	CIA profiles present on the generic card interface	Optional	Bit string	Bit i set to 1 indicates that profile i is present. Bit 0 indicates the presence of the profile in Annex A. All other bits are RFU.
CIA-PROFILES-AUTOMATIC	'82'	CIA profiles present on the generic card interface	Optional	Bit string	Bit i set to 1 indicates that the profile shall be generated by the ISO/IEC 24727-2 implementation Bit 0 indicates the presence of the profile in Annex A. All other bits are RFU.
DIGITAL-SIGNATURE-ON-CODE	'5F3D'	Digital signature information for procedural element	Optional	Data object of digital signature block	Key infrastructure for digital signature is out of scope
IF-PROFILE	'83'	Profile of ISO/IEC 24727-3 interface	Optional	'00'	Provided by the card. If exists, the card supports ISO/IEC 24727-3 interface.

This part of ISO/IEC 24727 defines profiles of the cryptographic information application, ISO/IEC 7816-15. The cryptographic information application data in these profiles shall be found either in the alpha card-application or in the cryptographic information application. See Annex A for a profile definition.

6.2 Application capability description (ACD)

Each card-application including the alpha card-application may be described by an ACD data object ('7F63').

Table 15 lists the data objects that may be found in an ACD. These data objects contain information about the card-application with which it is associated.

Table 15 –Data Objects in Application Capability Description ('7F63')

Symbol	Tag	Description	Mandatory/ Optional	Value	Note
LANG	'A1'	Procedural element description template (See 6.3)	Optional	Data objects as defined in Table 16	Provided by the card-application or the ISO/IEC 24727-2 implementation
LANG-URL	'5F50'	URL of the code that performs the translation	Optional		
SERVICE-DESCRIPTION	'7F66'	Description of the services supported by the card-application	Optional	Value field of data object is the concatenation of a DER-encoded CIAInfo value followed by DER-encoded ISO/IEC 7816-15 **CIOChoice** values as described in Annex C	
SERVICE-DESCIPTION-LOCATION	'7F67'	URL of a description of the services supported by the card-application	Optional	Value of URL is the location of the resource containing the concatenation of DER-encoded ISO/IEC 7816-15 CIOChoice values as described in Annex C	
DIGITAL-SIGNATURE-ON-CODE	'5F3D'	Digital signature information for procedural element	Optional	Data object of digital signature block	Key infrastructure for digital signature is out of scope

6.3 Procedural elements

A procedural element in a capability description shall be a translation code to process any GCI request belonging to Table 2 and every relevant card-application confirmation according to the TranslationCode function described below. Procedural elements in the CCD or an ACD are resident on the card itself or are referenced using a URL.

The procedural element description template given in Table 16 describes the procedural description technology used in the CCD or an ACD and contains and/or points to executable code in this procedural

description technology to perform the translations between GCI requests/confirmations and card interface commands/responses APDU.

Table 16 – Data Objects in the Procedural Element Description Template ('A1')

Symbol	Tag	Description	Mandatory/ Optional	Value
LANG-OID	'06'	Object identifier of the standard or specification describing the procedural language.	Optional	{iso(1) standard(0) iso20060(20060)}
LANG-CODE	'81'	Translation code	Optional	LANG-OID specific Tokens

When the procedural element (tag 'A1') in a capability description contains a data object for a URL (tag '5F50') and the URL refers to a document properly formatted as a capability description (see Tables 14 and 15), then the procedural element in the capability description shall be augmented with the procedural element retrieved from the referred document.

6.3.1 Model of computation for procedural elements

A procedural element in a capability description shall be a translation code intended to process the arguments conveyed by the TranslationCode function. Procedural elements can be explicitly selected by a SELECT command (see Table 3), especially if they do not mandate transmission of APDUs to the card. It is not precluded that interactions with the card occur.

The other features of this model apply to scripts recovered from the card. They do not apply to code downloaded from the outside world.

A procedural element in a capability description shall be a function with one Boolean argument and one byte array argument.

When the procedural element is placed in execution to transform a GCI command APDU to one or more card interface command APDUs, the Boolean argument shall be set to TRUE.

When the procedural element is placed in execution to transform a card response APDU to a GCI confirmation, the Boolean argument shall be set to FALSE.

Upon entry, if the Boolean argument to the procedural element is TRUE, the other argument, the octet array, shall contain the octets comprising the GCI command APDU.

Upon entry, if the Boolean argument to the procedural element is FALSE, the argument array shall contain the octets comprising the confirmation.

Upon exit, the procedural element shall set the Boolean argument to TRUE to return the transformed byte array argument to the Part 3 implementation. The procedural element shall set the Boolean argument to FALSE to send the transformed byte array to the card.

The signature of the entry point to the translation code shall be

TranslationCode(Boolean b IN/OUT, Array c IN/OUT)

6.3.2 Use of procedural elements

The ACD of the currently selected card-application and the CCD shall be tried in this order to find the procedural element in its ACD for handling a GCI request or card-interface response APDU.

If the currently selected application has provided a procedural element then each GCI request or card-interface response APDU shall be given to this procedural element.

If the currently selected application has not provided a procedural element and the CCD has provided a procedural element then each GCI request or card-interface response APDU shall be given to this procedural element.

If a GCI request is received on the GCI for which there is no procedural element in either the CCD or ACD of the currently selected application, then the request shall be sent to the card and the card-interface response shall be returned to the entity having made the GCI request.

6.4 Determining the value of capability descriptions

6.4.1 General principle

If the value of the capability description is not already present in the ISO/IEC 24727-2 middleware then this value shall be determined by retrieval of data objects according to the procedure described below.

6.4.2 Determining the value of the CCD

Determining the value of the CCD shall use application-independent card services defined by ISO/IEC 7816-4.

The value of the CCD may be determined immediately after card reset as follows. If the interindustry data element 'initial access data' is present in the historical bytes of the ATR, ATS or answer to ATTRIB, the data object '7F62' may be determined using the initial data string.

If not, the CCD shall be determined using one of the procedures below. The order in which the procedures defined below shall be tried is not defined. If all these procedures fail to determine a CCD the card does not comply with this standard.

- reading the EF.ATR, where data object '7F62' may be present;
- a GET DATA command with either
    - INS='CA'; P1-P2='7F62' and Le='00' or
    - INS='CB'; P1-P2='3FFF' and command data field containing '5C027F62'

  which may return the CCD in the response data field;
- by selection of the alpha card-application using the AID 'E8 28 81 C1 17 02' followed by a GET DATA command as described above.

If the list of applications has not been determined using the above procedures, then EF.DIR shall be read to determine the list of card-applications.

6.4.3 Determining the value of an ACD

Determining the value of an ACD may be performed immediately after the card-application has been selected using one of the following procedures:

- reading a file referenced in the response to the SELECT command under tag '87';
- a GET DATA command with either
    - P1-P2='7F63' and Le='00' or
    - P1-P2='3FFF' and command data field containing '5C027F63'

  which may return the ACD in the response data field.

Annex A
(informative)

Profiles for the cryptographic information application on the generic card interface

A.1 Profile A

The presence of an ISO/IEC 7816-15 cryptographic information application conformant with this profile is indicated by setting bit 1 of the first byte in the value field of the CIA-PROFILES data object in Table 14 to 1.

A.1.1 EF.CIAInfo

EF.CIAInfo is mandatory. The file identifier of EF.CIAInfo is '5032'.

A.1.2 EF.OD

EF.OD is mandatory. The file identifier of EF.OD is '5031'. EF.OD should not contain any **trustedPublicKeys**, **trustedCertificates**, or **usefulCertificates** components.

A.1.3 EF.PrKD

EF.PrKD may be present. The following constraints apply to each private key object referenced in EF.PrKD:

- The **CommonKeyAttributes.startDate** and the **CommonKeyAttributes.endData** components may be present but their values should be ignored by a client application conformant with ISO/IEC 24727.
- The **CommonKeyAttributes.accessFlags**, **CommonKeyAttributes.keyReference**, and **CommonKeyAttributes.algReference** components should be present.
- The **CommonPrivateKeyAttributes.generalName** component may be present.

A.1.4 EF.PuKD

EF. PuKD may be present. The following constraints apply to each public key object referenced in an EF.PuKD:

- For the **CommonObjectAttributes**, any components besides the **label** component should be ignored by a client application conformant with ISO/IEC 24727.
- The **CommonKeyAttributes,algReference** component should be present.
- The **CommonKeyAttributes.accessFlags**, **CommonKeyAttributes.startDate**, and **CommonKeyAttributes.endDate** components may be present.
- The **CommonPublicKeyAttributes.trustedUsage** and the **CommonPublicKeyAttributes.generalName** components may be present.

A.1.5 EF.SKD

EF.SKD may be present. The following constraints apply to each secret key object referenced in an EF.SKD:

- The **CommonKeyAttributes,algReference** component should be present.
- The **CommonKeyAttributes.accessFlags**, **CommonKeyAttributes.startDate**, and **CommonKeyAttributes.endDate** components may be present.
- The **CommonSecretKeyAttributes.keyLen** component should be present.

A.1.6 EF.CD

EF.CD may be present. The following constraints apply to each certificate object referenced in an EF.CD:

- The **CommonCertificateAttributes.certHash** and **CommonCertificateAttributes.validity** components may be present.
- The **indirect** option of **ObjectValue** should be used for all certificate types.

A.1.7 EF.AOD

EF.AOD may be present. The following constraints apply to each authentication object referenced in an EF.AOD:

- The **CommonObjectAttributes.flags** and **CommonObjectAttributes.userConsent** components may be present.
- For password objects, the **PasswordAttributes.path** component should be present.

A.1.8 EF.DCOD

EF.DCOD may be present. The following constraints apply to each data container object referenced in an EF.DCOD:

- The **CommonObjectAttributes.userConsent** component may be present.
- Only the **opaqueDO** choice should occur as a **DataContainerObjectChoice** value.
- The **indirect** option of **ObjectValue** should be used for all **OpaqueDOAttributes** values.

Annex B
(informative)

Instances of profile A

B.1 eSign K Specification

```
-- This example of Profile A describes a CWA 14890 (eSign K specification)
-- compliant signature application.
--
-- The application makes use of two Security Environments. SE#1 is used in a trusted
-- environment that is under the control of the card holder. In SE#1 commands are
-- applied without Secure Messaging. SE#2 is used in an untrusted environment where a
-- device authentication with session key establishment for Secure Messaging is required.
-- The card holder makes use of the Display Message (DM) to verify that a trusted channel
-- has been established. The DM can be read after a successful device authentication with
-- Secure Messaging only.
--
ESignK-SignatureApplication
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
   CIAInfo, CIOChoice, AuthenticationObjectChoice,
   PrivateKeyChoice, PublicKeyChoice,
   CertificateChoice, DataContainerObjectChoice
    FROM
   CryptographicInformationFramework;


-- Definition of ISO 7816-15 Directory Files
EFODF ::= SEQUENCE OF CIOChoice
EFAODF ::= SET OF AuthenticationObjectChoice
EFPrKDF ::= SEQUENCE OF PrivateKeyChoice
EFPuKDF ::= SEQUENCE OF PublicKeyChoice
EFCDF ::= SEQUENCE OF CertificateChoice
EFDCODF ::= SEQUENCE OF DataContainerObjectChoice

-- EF.CIAInfo
eSignK-EFCIAInfo CIAInfo ::=
{  version v2,
   profileIndication {"CWA 14890"},
   serialNumber ''H,
   label "Signature Application",
   cardflags { authRequired, prnGeneration },
   seInfo
    { { se 1,
        aid 'A000000167455349474E'H }, -- AID of eSignK application
         { se 2,
         aid 'A000000167455349474E'H } }, -- AID of eSignK application

   supportedAlgorithms
    {
      -- Hash algorithms
      -- AlgID: 0x10, SHA-1
      { reference 1,             -- Unique Reference
        algorithm 544,           -- PKCS#11 Mechanism Type CKM_SHA_1 = 0x220
        parameters  NULL: NULL,  -- Type of parameters is NULL and Value is NULL
        supportedOperations {hash},
        objId {1 3 14 3 2 26 },
        algRef 16                -- Is equvivalent to 0x10, see CWA 14890-1, table 13-1
      },

      -- Digital signature algorithms
      -- AlgID: 0x11,  RSA with DSI acc. to ISO/IEC 9796-2 with random number and SHA-1
      { reference 2,             -- Unique Reference, Cross reference from PrKDF
```

```
        algorithm 2147483648,     -- algorithm Not defined in PKCS#11,
                                  -- Vendor defined 0x80000000
        parameters NULL: NULL,    -- Type of parameters is NULL and Value is NULL
        supportedOperations {compute-signature},
        objId {1 3 36 3 4 3 2 1},
        algRef 17                 -- Is equivalent to 0x11, see CWA 14890-1, table 13.1
      },

      -- AlgID: 0x12, RSA with DSI acc. to PKCS#1 and SHA-1
      { reference 3,              -- Unique Reference, Cross reference from PrKDF
        algorithm 6,              -- PKCS#11 Mechanism Type CKM_SHA1_RSA_PKCS
        parameters NULL: NULL,    -- Type of parameters is NULL and Value is NULL
        supportedOperations {compute-signature},
        objId {1 2 840 113549 1 1 5},
        algRef 18                 -- Is equivalent to 0x12, see CWA 14890-1, Table 13-1
      },

      -- Device authentication algorithm
      -- AlgID: 0x17, Key Transport Protocol defined in CWA 14890-1
      {
         reference 4,
         algorithm 2147483649,    -- algorithm Not defined in PKCS#11,
                                  -- Vendor defined 0x80000001
         parameters NULL: NULL,   -- Type of parameters is NULL and Value is NULL
         supportedOperations {compute-signature, verify-signature},
         objId {1 3 36 7 2 1 1},  -- see CWA 14890-1 Key Transport Protocol
         algRef 23                -- Is equivalent to 0x17, see CWA 14890-1, Table 13-2
      },

      -- Card Verifiable (CV) certificate signature verification
      { reference 5,              -- Unique Reference, Cross reference from PuKDF
        algorithm 2147483650,     -- algorithm Not defined in PKCS#11,
                                  -- Vendor defined 0x80000002
        parameters NULL: NULL,    -- Type of parameters is NULL and Value is NULL
        supportedOperations {verify-signature},
        objId {1 3 36 3 4 3 2 1},
        -- algRef is not used (key and algorithm are selected by the Certification
        -- Authority Reference CAR provided in the Card Verifiable CV certificate)
      }
    }
 }

-- EF.ODF
eSignK-EFODF EFODF ::=
 {
   authObjects : path : { efidOrPath '4003'H },
   privateKeys : path : {efidOrPath '4001'H},
   publicKeys : path : {efidOrPath '4002'H },
   certificates : path : {efidOrPath  '4005'H },
   dataContainerObjects : path : { efidOrPath '4006'H }
 }

-- EF.AODF
eSignK-EFAODF EFAODF ::=
 {
   -- PIN.CH.AUT
   pwd :
    {
      commonObjectAttributes
       { label "global password",
         authId '03'H, -- Cross-Reference to PUK.CH.AUT
         -- in SE#2 the VERIFY and CHANGE REFERENCE DATA command shall be applied
         -- with Secure Messaging. The corresponding session keys are established by
         -- means of a device authentication.
         -- For SE#1 no security conditions apply, i.e., these commands can always be
         -- executed (without Secure Messaging)
         accessControlRules
         {
```

```
          { accessMode { execute },
            securityCondition authReference:
              {
                authMethod { secureMessaging, extAuthentication },
               seIdentifier 2
              }
          }
          }
     },

    classAttributes
     { authId '01'H },

    typeAttributes
     { pwdFlags { initialized},
       pwdType ascii-numeric,
       minLength 4,      -- in characters
       storedLength 0,   -- in bytes, padding is not required
       maxLength 8,      -- in characters
       pwdReference 1,   -- 0x01 Key Id
       -- Path is not required as PIN.CH.AUT is a global password
     }
  },

-- PIN.CH.DS
pwd :
  {
    commonObjectAttributes
     { label "Signature password",
       -- The same access control rules apply as in the case of the global password
       accessControlRules
        {
         { accessMode { execute },
           securityCondition authReference:
             {
               authMethod { secureMessaging, extAuthentication },
              seIdentifier 2
             }
         }
       }
     },

    classAttributes
     { authId '02'H },

    typeAttributes
     { pwdFlags { local, unblock-disabled, initialized },
       -- no resetting codes are supported
       pwdType ascii-numeric,
       minLength 6,           -- in characters
       storedLength 0,        -- in bytes, padding is not required
       maxLength 8,           -- in characters
       pwdReference 129,      -- 0x81 Key Id
       path { efidOrPath '3F 00 3F 01'H }
       -- Path of the DF.ESIGN that has to be selected prior to the VERIFY command
       -- (local password)
     }
  },

-- PUK.CH.AUT
pwd :
  {
    commonObjectAttributes
     { label "resetting code for the global password",
       -- In SE#2 the command RESET RETRY COUNTER shall be applied
       -- with Secure Messaging. The corresponding session keys are established by
       -- means of a device authentication.
       -- For SE#1 no security conditions apply, i.e., the command can always be
```

```
          -- executed (without Secure Messaging)
          accessControlRules
           {
            { accessMode { execute },
              securityCondition authReference:
                {
                   authMethod { secureMessaging, extAuthentication },
                  seIdentifier 2
                }
            }
           }
        },

       -- for cross-reference from PIN.CH.AUT
       classAttributes
        { authId '03'H },

       typeAttributes
        { pwdFlags { local, initialized, unblockingPassword },
          pwdType ascii-numeric,
          minLength 8,            -- in characters
          storedLength 0,         -- in bytes, padding is not required
        }
      }
    }


-- EF.PrKDF
eSignK-EFPrKDF EFPrKDF ::=
 {
   -- SK.CH.DS
   privateRSAKey :
    {
      commonObjectAttributes
       { label "Signature Key",
         flags { private },
         authId '02'H,  -- Cross-Reference to PIN.CH.DS
          userConsent 1,
          -- In SE#1 a user authentication by means of the signature password PIN.CH.DS
          -- is required prior to every PSO: COMPUTE DIGITAL SIGNATURE command.
          -- In SE#2 the PSO command shall be applied with Secure Messaging. The
         -- corresponding session keys are established by means of a device
         --  authentication. In addition a user verification by means of PIN.CH.DS is
         -- required prior to every use of the PSO command.
          accessControlRules
          {
            { accessMode { execute },
              securityCondition or:
              { and: { authId: '02'H, -- Cross-Reference to PIN.CH.DS
                       authReference: { authMethod { userAuthentication },
                                        seIdentifier 1
                                      }
                     },
                and: { authId: '01'H, -- Cross-Reference to PIN.CH.AUT
                       authReference: { authMethod { secureMessaging,
                                                     extAuthentication,
                                                        userAuthentication },
                                        seIdentifier 2
                                      }
                     }
              }
            }
          }
          }
       },

       classAttributes
        { iD '01'H, -- shall share the same iD with certificate
          usage { sign , signRecover, nonRepudiation},
```

```
        native TRUE,
        accessFlags { sensitive,
                      alwaysSensitive,
                      neverExtractable,
                      cardGenerated },
        keyReference 132, -- 0x84 KID in the card
         algReference
          {
            2, -- RSA ISO with SHA1
            3  -- RSA PKCS#1 with SHA1
          }
       },

      typeAttributes
       {
          value indirect : path : {efidOrPath ''H},
        modulusLength 1024
       }
    },

   -- SK.ICC.AUT
   -- The private key of the ICC used for device authentication and session
   -- key establishment.
   -- The key is used in the key transport protocol specified in CWA 14890-1.
   privateRSAKey :
    {
      commonObjectAttributes
       { label "SK.ICC.AUT",
         flags { private },
    },

      -- shall share the same iD with Card Verifiable (CV) certificate C_CV.ICC.AUT
      classAttributes
       { iD '02'H,
         usage { decipher, signRecover },
         native TRUE,
         accessFlags { sensitive,
                       alwaysSensitive,
                       neverExtractable },
         keyReference 17, -- 0x11 KID in the card
          algReference
           {
             4  -- device authentication, key transport protocol
           }
       },

      typeAttributes
       { value indirect : path : {efidOrPath ''H },
         modulusLength 1024
       }
    }
 }

-- EF_PuKDF
eSignK-EFPuKDF EFPuKDF ::=
 {
   -- PK.RCA.CS-AUT
   -- The public key of the Root CA that is stored as security anchor in the ICC. The key
   -- is used to verify CV certificates in the context of the device authentication.
   publicRSAKey :
    {
      commonObjectAttributes
       { label "PK.RCA.CS-AUT"},

      classAttributes
       { iD '03'H,
         usage { verifyRecover },
         native TRUE,
```

```
          -- The Certificate Holder Reference (CHR) is used as key reference
          -- see CWA 14890-1, Chapter 14
          keyReference 1122334455667788,
          algReference
           {
             5 -- algorithm for CV certificate verification
           }
        },

       typeAttributes
        { value indirect : path : { efidOrPath ''H },
          modulusLength 1024
        }
    }
 }

-- EF.CDF
eSignK-EFCDF EFCDF ::=
 {
   -- C_X509.CH.DS
   -- The certificate of the card holder for the digital signature service
   x509Certificate :
    {
      commonObjectAttributes
       { label "certificate for signature service" },

      -- shall share the same id with private key
      classAttributes
       { iD '01'H,
         authority FALSE },

       -- Path to EF.C.X509_1.CH.DS where the certificate is stored
      typeAttributes
       { value indirect : path : {efidOrPath '3F 00 3F 01 C0 00'H } }
    },

   -- C_X509.CA.CS-DS
   -- The certificate of the CA that is issuer of C_X509.CH.DS
   x509Certificate :
    {
      commonObjectAttributes
       { label "CA certificate for signature service"},

      classAttributes
       { iD '01'H,
         authority TRUE },

      -- Path to EF.C_X509.CA.CS where the certificate is stored
      typeAttributes
       { value indirect : path : {efidOrPath '3F 00 3F 01 C6 08'H} }
    },

   -- C_CV.ICC.AUT
   -- The Card Verifiable certificate of the ICC used in the device authentication service
   cvCertificate :
    {
      commonObjectAttributes
       { label "C_CV.ICC.AUT" },

      -- shall share the same id with private key
      classAttributes
       { iD '02'H,
         authority FALSE },

      -- Path to EF.C_CV.ICC.AUT where the certificate is stored
      typeAttributes
       { value indirect : path : {efidOrPath '3F 00 2F 03'H} }
    }
```

```
 }

-- EF_DCODF
eSignK-EFDCODF EFDCODF ::=
 {
   -- Display Message
   -- The Display Message mechanism is used by the card holder to make sure that
   -- a trusted channel between the interface device and the ICC is established,
   -- see CWA 14890-1.
   opaqueDO :
    {
      commonObjectAttributes
       { label "Display Message",
         flags {private, modifiable},
         accessControlRules
          {
            { accessMode { update },
              securityCondition or:
               { and: { authId: '01'H, -- Cross-Reference to PIN.CH.AUT
                        authReference: { authMethod { userAuthentication },
                                         seIdentifier 1
                                       }
                      },
                 and: { authId: '01'H, -- Cross-Reference to PIN.CH.AUT
                        authReference: { authMethod { secureMessaging,
                                                      extAuthentication,
                                                         userAuthentication },
                                         seIdentifier 2
                                       }
                      }
               }
            },

            { accessMode { read },
              securityCondition authReference:
               { authMethod {secureMessaging,extAuthentication },
                 seIdentifier 2
               }
            }
          }
       },

      -- applicationName or applicationOID have to be present
      classAttributes
       { applicationName "A000000167455349474E" }, -- AID of eSignK application

      -- path to the EF that contains the Display Message
      typeAttributes
       indirect : path : {efidOrPath '3F 00 3F 01 D0 00'H }
    }
 }

END
```

Annex C
(normative)

Cryptographic information application for card-application service description

Associated with each card-application via a data object with tag '7F66' or '7F67' in the Card-Application Capability Description is a DER-encoded ISO/IEC 7816-15 cryptographic information application that describes the services offered by the card-application. This information is used by an ISO/IEC 24727-3 implementation to translate calls on the ISO/IEC 24727-3 application interface to commands on the ISO/IEC 24727-2 generic card interface. The value field of the '7F66' data object (or the data referenced by the URL in the '7F67' data object) shall contain a DER-encoded **CardApplicationServiceDescriptionValue**. The **CardApplicationServiceDescriptionValue** is a **ISO/IEC 7816-15 CIAInfo** value which shall be followed by zero or more ISO/IEC 7816-15 **CIOChoice** values each of which encodes the **objects** choice of **PathOrObjects**.

```
CardApplicationServiceDescription ::= SEQUENCE {
  ciaInfo CIAInfo,
  cioChoice   SEQUENCE OF CIOChoice
}
```

The initial DER-encoded **CIAInfo** value describes the cryptographic algorithms, authentication protocols and security environments supported by the card-application.

The following sequence of DER-encoded **CIOChoice** values contains a **CIOChoice** entry for each ISO/IEC 24727 Data Set in the card-application, a **CIOChoice** entry for each ISO/IEC 24727 Differential-Identity in the card-application and a **CIOChoice** entry for each ISO/IEC 24727 Service in the card-application.

Each ISO/IEC 24727 Data-Set in an ISO/IEC 24727 card-application is represented by an ISO/IEC 7816-15 **dataContainerObjects** component element.

Each ISO/IEC 24727 Differential-Identity in an ISO/IEC 24727 card-application is represented by ISO/IEC 7816-15 authObjects, privateKeys, publicKeys, trustedPublicKeys or secretKeys component elements.

Each ISO/IEC 24727 Service in an ISO/IEC 24727 card-application is represented by an ISO/IEC 7816-15 **dataContainerObjects** element.

The DER-encoded concatenation of **CIOChoice** values comprising the Service Description of an ISO/IEC 24727 card-application may contain additional ISO/IEC 7816-15 **CIOChoice** values such as **privateKeys**, **publicKeys**, **secretKeys** and **certificates** components that are associated with Differential-Identities using the **authId** attribute. Any of the attributes specified by ISO/IEC 7816-15 may appear in these additional **CIOChoice** values.

Cryptographic Algorithms and Authentication Protocols

The leading **CIAInfo** value describes the cryptographic algorithms implemented by the card-application, in particular the cryptographic algorithms used in authentication protocols.

The mandatory **version** component is set to the edition of ISO/IEC 7816-15 to which the Service Description complies.

The mandatory **cardflags** component describes properties of the card-application per ISO/IEC 7816-15.

Elements of the **SEQUENCE OF AlgorithmInfo** list describe the cryptographic algorithms supported by the card-application. Reference is made to a specific algorithm using the **reference** field. The description of an algorithm shall include the object identifier (**objId**) of the algorithm and the algorithm reference (**algRef**) of the algorithm within the card-application.

ISO/IEC 24727 Data-Sets and Data Structures for Interoperability

Each Data-Set in an ISO/IEC 24727 card-application is represented in the Service Description of the card-application by an ISO/IEC 7816-15 **dataContainerObjects** element that is a **sequence of Data Container Information Objects**. The first Data Container Information Object describes the Data-Set and each subsequent Data Container Information Object describes a single Data Structure for Interoperability within the Data-Set.

The ISO/IEC 7816-15 **label** attribute in the Common Object Attributes of the first ISO/IEC 7816-15 Data Container Information Object in the sequence of data container objects comprising an ISO/IEC 24727 Data Set is the Name of the Data Set.

The ISO/IEC 7816-15 **accessControlRules** attribute of the first ISO/IEC 7816-15 Data Container Information Object in the sequence of Data Container Information Objects comprising an ISO/IEC 24727 Data-Set implements the ISO/IEC 24727 access control list for the Data Set and thus for all Data Structures for Interoperability (DSIs) in the Data Set using the access mode mapping described in Table C.4 below.

Each subsequent ISO/IEC 7816-15 Data Container Information Object in the sequence of Data Container Information Objects comprising an ISO/IEC 24727 Data Set represents an ISO/IEC 24727 Data Structure for Interoperability contained in the ISO/IEC 24727 Data Set.

The ISO/IEC 7816-15 **label** attribute in the Common Object Attributes of the ISO/IEC 7816-15 Data Container Information Objects representing an ISO/IEC 24727 Data Structure for Interoperability is the Name of the Data Structure for Interoperability.

The **indirect** CHOICE shall be indicated for the ObjectValue of typeAttributes of the ISO/IEC 7816-15 Data Container Information Object representing a DSI. The path described here shall be the mapping from the DSI Name to the realization of the DSI on the generic card interface. The path thus includes the location of the DSI within the card-application, the offset to the first byte of the DSI at this location and the length of the data in the DSI in bytes.

Table C.1 – ISO/IEC 7816-15 Encoding of a Data Set

Data-Set	**Attribute**	**Description**
Common Object Attributes	label	Name of the data-set
	accessControlRules	Access control list of the data-set; viz. for all DSIs in the data-set
Common Data Container Object Attributes	<unused>	Either applicationName or applicationOID shall be present and shall not have the value NULL. The value shall be ignored.
Data Object Attributes	<unused>	ObjectValue shall be present with CHOICE direct of value NULL and shall be ignored.
DSI		
Common Object Attributes	Label	Name of the DSI
Common Data Container Object Attributes	<unused>	Either applicationName or applicationOID shall be present and shall not have the value NULL. The value shall be ignored.
Data Object Attributes	iso7816DO.relative.path	Path to the DSI in the card-application, the offset to the first byte of the data at this location and the length of the data in bytes

ISO/IEC 24727 Differential-Identities

Each Differential-Identity in an ISO/IEC 24727 card-application is represented in the Service Description of the card-application by an ISO/IEC 7816-15 **authObjects** component that contains **exactly one Authentication Information Object**. These Authentication Information Objects describe all Differential-Identities recognized by the card-application not just the Differential-Identities whose authentication state variables appear in access rules.

The ISO/IEC 7816-15 **label** attribute in the Common Object Attributes of the ISO/IEC 7816-15 Authentication Information Object describing an ISO/IEC 24727 Differential-Identity is the Name of the Differential-Identity.

The ISO/IEC 7816-15 **accessControlRules** attribute of each Authentication Information Object implements the ISO/IEC 24727 access control list for the Differential-Identity using the access mode byte mapping described in Table 1 below.

The ISO/IEC 7816-15 **authId** attribute in the Common Object Attributes of the ISO/IEC 7816-15 Authentication Information Object describing an ISO/IEC 24727 Differential-Identity is optional. If present it contains the value found in the **reference** field of an entry in the **supportedAlgorithms** list in **ciaInfo** with the meaning that this algorithm is used to authenticate the Differential-Identity.

The ISO/IEC 7816-15 **authId** attribute in the Common Authentication Object Attributes of the ISO/IEC 7816-15 Authentication Information Object describing an ISO/IEC 24727 Differential-Identity is a unique identifier of the Differential-Identity within the Service Description. It is used for cross-referencing attributes of other information objects to this Differential-Attribute.

The ISO/IEC 7816-15 **authReference** attribute in the Common Authentication Object Attributes of the ISO/IEC 7816-15 Authentication Information Object describing an ISO/IEC 24727 Differential-Identity is the key reference used to refer to the Differential-Identity in security environments and access rules.

The **path** attribute in type-specific CIO attributes of the Authentication Information Object if present references the Marker associated with the Differential-Identity.

Table C.2 – ISO/IEC 7816-15 Encoding of a Differential-Identity

Differential-Identity	Attribute	Description
Common Object Attributes	label	Name of the Differential-Identity
	accessControlRules	Access control list for the Differential-Identity
	authId	Value of the reference field in a member of the supportedAlgorithms list in ciaInfo
Common Authentication Object Attributes	authId	Unique identifier of Differential-Identity within the card-application
	authReference	Key reference of the Differential-Identity
Auth Object Attributes	<any>	Description of parameters of the authentication used to authenticate the Differential-Identity

ISO/IEC 24727 Card-Application Services and Actions

Each Card-Application Service in an ISO/IEC 24727 card-application is represented in the Service Description of the card-application by an ISO/IEC 7816-15 **dataContainerObjects** element that contains a **sequence of Data Container Information Objects**. The first Data Container Information Object describes the Service. Subsequent Data Container Information Objects are optional and if present each describes an Action in the Service.

The ISO/IEC 7816-15 **label** attribute in the Common Object Attributes of the first ISO/IEC 7816-15 Data Container Information Object in the sequence of Data Container Information Objects is the Name of the Card-Application Service.

The ISO/IEC 7816-15 **accessControlRules** attribute in the Common Object Attributes of the first ISO/IEC 7816-15 Data Container Information Object in the sequence of Data Container Information Objects comprising an ISO/IEC 24727 Service represents the ISO/IEC 24727 access control list for the Service using the mappings described in Table C.4 below.

The ISO/IEC 7816-15 **iD** attribute in the Common Data Container Object Attributes of the first ISO/IEC 7816-15 Data Container Information Object in the sequence of Data Container Information Objects comprising an ISO/IEC 24727 Service contains a free-form description of the Service.

Each subsequent ISO/IEC 7816-15 Data Container Information Object in the sequence of Data Container Information Objects comprising an ISO/IEC 24727 Service describes one ISO/IEC 24727 Action in the ISO/IEC 24727 Service described by the first Data Container Information Object.

The ISO/IEC 7816-15 **label** attribute in the Common Object Attributes of an ISO/IEC 7816-15 Data Container Information Object in describing an Action is the Name of the Action.

The **indirect** CHOICE shall be indicated for the ObjectValue of typeAttributes of the ISO/IEC 7816-15 Data Container Information Object representing a DSI. The path described shall be the path to the executable code implementing the Action. The executable code may in fact realize an entire card-application; e.g. the executable code may be an ISO/IEC 20060 module, a MULTOS cardlet or a Java Card applet.

If a service or an action is not described within an ISO/IEC 7816-15 dataContainerObject, then its access condition should be set to "Never".

Table C.3 – ISO/IEC 7816-15 Encoding of a Service

Service	Attribute	Description
Common Object Attributes	label	Name of the service (per ISO/IEC 24727-3)
	accessControlRules	Access rules for the actions in the service whose Target is the Access Control List related to the service
Common Data Container Object Attributes	RFU	Either applicationName or applicationOID shall be present and shall not have the value NULL. The value shall be ignored.
Data Object Attributes	objectValue	ObjectValue with CHOICE indirect with the ReferenceValue being the location of executable code implementing the service within the card-application or NULL
Action		
Common Object Attributes	label	Name of the action (per ISO/IEC 24727-3); see mappings in Table C.4.
	accessControlRules	Access rules for the actions in the service whose Target is the Card-Application
Common Data Container Object Attributes	RFU	Either applicationName or applicationOID shall be present and shall not have the value NULL. The value shall be ignored.
Data Object Attributes	objectValue	ObjectValue with CHOICE indirect with the ReferenceValue being the location of executable code implementing the action within the card-application or NULL

Mapping of Actions to Access Mode Bytes

Table C.4 – ISO/IEC 24727-3 Actions mapped to the ISO/IEC 7816-15 CommonObjectAttributes.label

ISO/IEC 24727-3 Action	CommonObjectAttributes.label (Mandatory)
ACLList	“ACL_LIST”
ACLModify	“ACL_MODIFY”
CardApplicationEndSession	“CA_END_SESSION”
CardApplicationStartSession	“CA_START_SESSION”
CardApplicationDisconnect	“CA_DISCONNECT”
CardApplicationConnect	“CA_CONNECT”
CardApplicationCreate	“CA_CREATE”
CardApplicatonDelete	“CA_DELETE”
CardApplicatonServiceCreate	“CA_SERVICE_CREATE”
CardApplicationServiceLoad	“CA_SERVICE_LOAD”
CardApplicationServiceDelete	“CA_SERVICE_DELETE”
CardApplicationList	“CA_LIST”
CardApplicationServiceList	“CA_SERVICE_LIST”
CardApplicationServiceDescribe	“CA_SERVICE_DESCRIBE”
ExecuteAction	“EXECUTE_ACTION”
DataSetCreate	“DS_CREATE”
DataSetDelete	“DS_DELETE”
DataSetSelect	“DS_SELECT”
DataSetList	“DS_LIST”
DSICreate	“DSI_CREATE”
DSIDelete	“DSI_DELETE”
DSIList	“DSI_LIST”
DSIWrite	“DSI_WRITE”
DSIRead	“DSI_READ”
DIDList	“DID_LIST”
DIDCreate	“DID_CREATE”
DIDDelete	“DID_DELETE”
DIDUpdate	“DID_UPDATE”
DIDGet	“DID_GET”
DIDAuthenticate	"DID_AUTHENTICATE"
Encipher	“ENCIPHER”
Decipher	“DECIPHER”
GetRandom	“GET_RANDOM”
Hash	“HASH”
Sign	“SIGN”
VerifySignature	“VERIFY_SIGNATURE”
VerifyCertificate	“VERIFY_CERTIFICATE”