INTERNATIONAL STANDARD

ISO/IEC 24727-4

First edition
2008-11-01

AMENDMENT 1
2014-04-01

Identification cards — Integrated circuit card programming interfaces —

Part 4: Application programming interface (API) administration

AMENDMENT 1

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

Partie 4: Administration d'interface de programmation (API)

AMENDEMENT 1

Reference number
ISO/IEC 24727-4:2008/Amd.1:2014(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.

Amendment 1 to ISO/IEC 24727-4:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Cards and personal identificaition*.

XML encodings have become more and more used in the field of IAS (Identity, Authentication and (digital) Signature), Identity Management and general networking communication. To enhance interoperability with existing networking systems and federated identification and authorization systems (e.g. SAML, OpenID, etc.) standardization of an XML representation of the API and data structures of ISO/IEC 24727-3 is essential.

In order to support this addition to the ISO/IEC 24727-3 scope, the relevant stack configurations in ISO/IEC 24727-4 will be updated and/or amended. The rules governing the use of various marshalling/un-marshalling procedures will be aligned with the amendment to ISO/IEC 24727-3.

This Amendment has been prepared to:

1. Extend and update as necessary the stack configurations to address the XML representation such that it is compatible with the relevant XML-based standards (e.g. SAML).
2. Clarify use of secure messaging.
3. As a result of Amendments under development for other parts of 24727, portions of this standard may be deleted and referenced.
4. Consider additional forms of secure messaging and consider separating the security of information transferred across a general network versus security of information transferred across the card interface.
5. Refine TC_API to allow channel initiation for various mechanisms; e.g. web service communication (SOAP PAOS), AJAX.
6. Update the current XML specifications to align with ISO and not import 3 party schemas e.g. OASIS.
7. Remove ambiguities by elaborating and re-specifying concepts that may not be clear in the current standard.
8. Incorporate concepts that are captured in other parts of ISO/IEC 24727 but are more relevant for ISO/IEC 24727-4.
9. Include C and Java bindings in a Normative Annex (for C) and an Informative Annex (Java); moved from Part 5

10. Consider relocating data structure generation to the local machine level; e.g. remote ICC Stack

Note: A hybrid numbering scheme for new figures and tables is used (e.g. Figure 7-1) in order to fit more seamlessly with the original document; i.e. to NOT require many modifications to simply change figure and table numbers.

Identification cards — Integrated circuit cards programming interfaces —

Part 4: Application programming interfaces (API) administration

AMENDMENT 1

Page 12

Add the following new subclause before Clause 6:

5.7 Extensions of the Service Access Layer

5.7.1 Extensions under ISO/IEC 7816-15

This clause comprises an enhancement of the functionality assigned to the local platform through which is accessed the card-application. In so doing, it re-uses the existing ISO/IEC 24727-4 middleware stack configurations with only a change bearing upon the location of the component handling ISO/IEC 7816-15 - based registry. The main objective being to alleviate the workload of the client-application platform by supplying it upon request with pre-built data structures for interoperability (ISO/IEC 24727-3).

5.7.2 SAL API Lite

ISO/IEC 24727-2 AMD 1 envisions explicit (normative and informative elements, including the use of the ISO/IEC 7816-15 based Registry. As such, ISO/IEC 24727-2 standardizes the use of ISO/IEC 7816-15 data structures as "discovery information" that is communicated throughout the ISO/IEC 24727 stack. To achieve this discovery process, it is necessary to locate an ISO/IEC 7816-15 based Procedural Elements component within the platform connected to the card-application for the Remote-ICC-stack and the Opaque-ICC-stack (see Figure 7-1) as follows:

- This component is in charge of linking the entities defined at the Service Access Layer (API) (e.g. Differential Identity, DataSet, DSI) with typical "on-card" entities such as keys, files and Access Control Rules. Accordingly enabled with this Registry processing capability, this component is ONLY in charge of retrieving the ISO/IEC 7816-15 based Registry, parsing it, and generating the interoperability data structures representing the objects handled by the SAL API in conformance with ISO/IEC 24727-3.

- this component is part of a SAL API Lite implementation that implements a subset of existing SAL API herein called SAL API Lite

- the SAL API Lite is comprised ONLY of the the SAL API functions that serve to retrieve data structures for interoperability as DataSet, DSI, DID, ACL, etc. to supply it to the client-application upon request.

5.7.3 SAL API Lite Supported Requests:

The SAL API Lite interface provides a subset of the full ISO/IEC 24727-3 API. Specifically included are the following requests:

- **CardApplicationConnect(),**
- **CardApplicationList**(),

- **CardApplicationServiceList**(),
- **CardApplicationServiceDescribe**(),
- **DataSetList**(),
- **DataSetSelect**(),
- **DSIList**(),
- **DSIRead**(),
- **DIDList**(),
- **DIDGet**(),
- **ACLList**()

The SAL API Lite requests are limited in the following ways:

5.7.3.1 Registry Delegation

The Service Access Layer implemented on the same platform as the client-application shall delegate the Registry processing capability to the SAL API Lite implementation.

5.7.3.2 Registry Bootstrap

The SAL API Lite implementation shall start the ISO/IEC 7816-15 based Registry processing at bootstrap and save processing time to the Service Access Layer implementation by making available all the data structures defined at SAL API Lite interface.

5.7.3.3 Registry Connection

The SAL API Lite implementation shall provide to the application running on a local machine to which the card-application is connected e.g to allow for GUI to expose the user all or part of his on-card ISO/IEC 7816-15 based Registry.

5.7.3.4 ICC-Resident-Stack Exclustion

The SAL API Lite component does not fit in an ICC-Resident-stack for the following reasons: the SAL-Agent is located on-card and not visible to the outside world, and additionally, on ICC-Resident-stack, the data structures for interoperability can be built-in structures already available on-card, so they do not need to be derived from a ISO/IEC 7816-15 based Registry.

5.7.3.5 Static Library Option

Optionally, the SAL API Lite implementation may rely on a static or dynamic library to explore the on-card ISO/IEC 7816-15 based Registry. To this intent, informative Annex E is proposed with a comprehensive description and binding for an API typically handling on-card ISO/IEC 7816-15 based Registry.

Figure 7-1 illustrates a Full Network Stack that makes use of a SAL API Lite registry facility. In addition, this figure illustrates the use of a procedural element that can translate the formal language representation of the ISO/IEC 24727-3 API functions directly into card specific APDUs. The only required ISO/IEC 24727-2 GCI APDU is the ENVELOPE APDU. The contents of the ENVELOPE APDU can be, for example, DER-TLV represented API functions. Thus, the procedural elements can function as a full Service Access Layer.

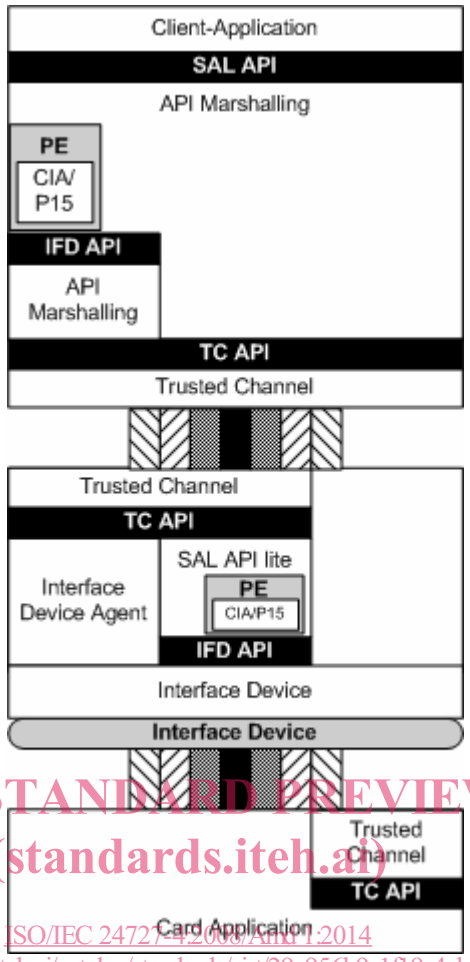

Figure 7-1 — Network stack

5.7.4 Secure Messaging

Within a stack configuration as shown in Figure 7-1, a client-application may establish an end-to-end secure channel with a card-application. The secure APDU commands generated by the SAL implementation are delivered as IFD API marshalled commands to the local host where the Interface Device Agent redirects them to the Interface Device. The SAL API Lite component does not process secure APDUs. Accordingly, whenever the ACL controlling the access to a data structure e.g. DataSet, requires secure messaging, the transaction shall go through the following sequence :

1. client-application sends for the first time during a session a DataSetList call through SAL API
2. the API call is marshalled into either XML or DER-TLV and sent through TC to the SAL API Lite component
3. the SAL API Lite builds the requested data structure (i.e DataSetNameList) as part of the DataSetListResult and returns it through TC layer to the SAL.
4. the SAL hands on the DataSetListResult to the client-application.
5. the client-application sends an ACLList call through SAL API with a parameter targetType denoting a data-set and a parameter targetName related to a named DataSet that was retrieved from the just received list of DataSetNameList (before this call, the client-application may optionally selects a named DataSet and sends a DataSetSelect call through SAL API)
6. the API call is marshalled into either XML or DER-TLV and sent through TC to the SAL API Lite component

7. the SAL API Lite builds the requested data structure (i.e AccessControlList) as part of the ACLListResult and returns it to through TC layer to the SAL.
8. the SAL hands on the ACLListResult to the client-application.
9. the client-application parses the set of AccessRules contained in ACLListResult and figures out which Differential-Identity is to be verified in order to get access to the DataSet contents (with the assumption that the targetted DataSet is protected with a DID controlling access to its contents e.g DIDRead requiring mutual authentication with establishment of secure messasing).
10. the client-application sends a DSIList call through SAL API
11. the API call is marshalled into either XML or DER-TLV and sent through TC to the SAL API Lite component.
12. the SAL API Lite builds the requested data structure (i.e DSINameList) as part of the DSIListResult and returns it to through TC layer to the SAL. in order to allow the client-application to read properly the DSI contents, the SAL API Lite maps the DSI attributes onto on-card attributes; for this purpose the SAL API Lite uses the Cryptographic Information Application (CIA or P15) to figure out the on-card attributes related to each DSI in the DSIListResult. Alternatively, the SAL API Lite may send a SELECT APDU command to the card with parameter P2 requesting the Control Parameter (CP) Template (see ISO/IEC 7816-4); accordingly, a DO'98' within CP denotes DO instances, a DO'9B' within CP denotes EF instances in either DF or ADF, and a DO'97' within CP denotes DF instances in either DF or ADF. To map those on-card attributes onto DSI attributes, an additional parameter is required in DSIListResult definition in ISO/IEC 24727-3.
13. the SAL hands on the DSIListResult to the client-application
14. the client-application sends a DIDAuthenticate (according step 9 above) to fulfil the access rules controlling the reading of DSI contents.
15. the SAL parses the DID requirements (i.e DIDName, DIDAuthenticationData) and starts the authentication protocol with the card i.e SAL generates the appropriate APDU commands and send them out to the card through IFD API and TC layer.
16. Once the end-to-end secure messaging is established between the card and the SAL, the SAL delivers the API return code to the client-application
17. the client-application then sends a DSIRead call through the SAL API.
18. the SAL generates the corresponding APDU command conforming to DSI attributes received in step 12 above; then SAL secures the APDU commands with current session keys and sends it via the IFD API, TC and Interface Device to the Card Application;
19. the SAL recovers the DSI contents and builds the dsiContent as part of the DSIReadResult, then return it to the client-application.

Table 7-1 indicates the messages exchanged between SAL API and SAL API Lite during a transaction with secure messaging :

Table 7-1 – Command Progression

Client-Application	SAL	IFD Proxy	TC	IFD Agent	SAL API Lite	IFD	Card-Application
					Registry processing		
DataSetList	DataSetList	↔	↔	↔	DataSetList		
DataSetSelect	DataSetSelect	↔	↔	↔	DataSetSelect		
ACLList	ACLList	↔	↔	↔	ACLList		
DSIList	DSIList	↔	↔	↔	DSIList		
DIDAuthenticate	APDU C-RP (*)	↔	↔	↔		↔	↔
	← SECURE MESSAGING →						
DSIRead	APDU C-RP	↔	↔	↔		↔	↔

(*) APDU C-RP = APDU command-response pair i.e GET DATA or READ BINARY

Page 47

Add the following new clause before Annex A:

8 Registry Implementations

8.1 ISO/IEC 7816-15 registry implementation

To achieve interoperability among various Service Access Layer implementations, it is necessary to record the mechanisms used to translate between ISO/IEC 24727-3 API functions and ISO/IEC 24727-2 GCI APDUs. This clause defines how to prepare this record in the form of an ISO/IEC 7816-15 formatted Registry.

Included is the representation of ISO/IEC 24727-3 Card-Application discovery information as an ISO/IEC 7816-15 Cryptographic Information Application.

The ISO/IEC 7816-15 representation of an ISO/IEC 24727 card-application contains all the information that the ISO/IEC 24727-2, ISO/IEC 24727-3 implementations need to realize the ISO/IEC 24727 specified interoperable connection between the client-application and the card-application.

Data Structures for Interoperability are discovery information made available to the SAL by the card through the bootstrap mechanism, unless it is provided by other means. The SAL API or the SAL API Lite shall undertake the processing of the discovery information in order to dynamically generate the data structures defined in ISO/IEC 24727-3 as Access Control List, Differential-Identity, Data-Set and Data Structures for Interoperability (DSI), and Card-Application Services and Actions. These data structures shall be surfaced to the client-application upon request through the SAL API or the SAL API Lite.

To allow for coding of access control rules conforming to ISO/IEC 7816-15 implementation, a mapping of SAL API actions onto ISO/IEC 7816-15 *accessMode* attributes is described. This mapping is laid on the extension of accessMode according the second amendment of ISO/IEC 7816-15.

The information in the ISO/IEC 7816-15 representation of an ISO/IEC 24727 card-application is referred to informally in ISO/IEC 24727-1 as the *discovery information* and in ISO/IEC 24727-4 as an *ISO/IEC 7816-15 based Registry.*

To implement the ISO/IEC 7816-15 based Registry, several ASN.1 definitions are available:

- Clause 8.1.3.1 establishes the ASN.1 types for translation of SAL API calls into card-specific APDU; the APDU command parameters rendering each SAL API Action can be so determined by the issuer and DER-TLV encoded then hosted on-card as part of the Cryptographic Information Application or on a remote repository that can be accessible though the interface of which the XML-Binding is described in Annex I [informative] “interoperable access to the repository”.

- Annex H[informative] illustrates the actual ASN.1 reference Module implementing clause 8.1.3.1. Common ASN.1 types are imported from ISO 24727-3; only types newly designed in the Amendment are defined within this module.

- Annex G[informative] illustrates application of the rules and guidelines described in clause 8.1.1 “ISO/IEC 24727-3 data structures mapping” to build an ISO/IEC 7816-15 based Registry for a fictitious service called “myService”. In G.1.1.2.1, an ASN.1 value notation illustrates an implementation example; accordingly, DER-encoding may be derived from this notation.

8.1.1 ISO/IEC 24727-3 data structures mapping

The data structures that may be surfaced to the client-application upon request are DataSet, DSI, ACL, Differential-Identities, list of Card-Application Services, list of Differential-Identities.

The SAL generates these data structures from the information available in CardApplicationServiceDescription that is defined as an ASN.1 SEQUENCE of ISO/IEC 7816-15 CIAInfo value along with consecutive ISO/IEC 7816-15 CIOChoice(s) value(s).

The interpretation of CardApplicationServiceDescription value in terms of ACL, Differential-Identities, Services, DataSet or DSI, is based on the mapping described in the following sub-clauses.

8.1.1.1 DataSet

Table 8-1 indicates the mapping of a DataSet onto a DataContainerObjectChoice and describes how the DataSet's ACL can be derived from it. In this and following tables, an “X” in the “ACL items” column indicates that this attribute may be controlled through an Access Control List (ACL) entry.

Table 8-1 — Mapping DataSet to ACL

ISO/IEC 7816-15 attribute	**ISO/IEC 7816-15 sub-attribute(s)**	**ISO/IEC 24727-3**		**Comments**
		Description	**ACL items**	
DataContainerObject-Choice iso7816DO commonObjectAttributes	commonObjectAttributes.label	DataSet Name	X	The label is ASCII encoded and shall evaluate to '**DATA-SET**' used as a prefix denoting the target type, concatenated with the DataSet actual name.
	accessControlRule.accessMode	Action	X[1]	Shall conform to Table 10 and Table 8-8
	accessControlRule.securityCondition	Ref to DID		authId referring to a CIO within the cryptographic information application (DF.CIA)
ClassAttributes	classAttributes.applicationName			Either applicationName or application OID of the card-application owning the DataSet. Shall not be NULL.
TypeAttributes	<unused>			NULL
ISO/iEC 7816 DO container for the DSI(s) within the DataSet				
CommonObjectAttributes	commonObjectAttributes.label	DSIName		The DSI Name shall be ASCII encoded.
	accessControlRule.accessMode	<unused>		DSI's accessRules are controlled by DataSet

ISO/IEC 7816-15 attribute	ISO/IEC 7816-15 sub-attribute(s)	ISO/IEC 24727-3		Comments
		Description	ACL items	
	accessControlRule.securityCondition	<unused>		DSI's accessRules are controlled by DataSet accessRules
ClassAttributes		<unused>		
TypeAttributes	efidOrTagChoice	location of the actual resource (EF,DF, ADF, DO) hosting the DSI content.		File Id or extended Path (acc. ISO/IEC 7816-15:2004/AM2) to the DSI resource
[1] In case the action is controlled by a logical combination of DIDs, several consecutive securityCondition values may provide different security conditions applying to the same access Mode denoting the SAL API action. The logical operation combining the DIDs is AND or OR depending on the ASN.1 encoding of the securityContitions.				

The DIDs referenced by the access rules protecting a DataSet are either Authentication Information Objects or Private Objects or Secret Key. Secret Key and Private Key mapping to Differential-Identities is presented in subsequent tables.

8.1.1.2 CardApplication

Table 8.2 shows the mapping of a CardApplication onto a DataContainerObjectChoice and describes how the CardApplication's ACL can be derived from it.

Table 8-2 — mapping CardApplication to DataContainerObjectChoice

ISO/IEC 7816-15 attribute	ISO/IEC 7816-15 sub-attribute(s)	ISO/IEC 24727-3		Comments
		Description	ACL items	
DataContainerObject-Choice ISO/IEC 7816 DO commonObjectAttributes	commonObjectAttributes.label	Card-Appli-cation Name	X	The label is ASCII encoded and shall valuate to **'CARD-APPLICATION'** used as a prefix denoting the target type, concatenated with the CardApplication actual name.
	accessControlRule.accessMode	SAL API Action	X [1]	SAL API actions shall conform to value in Table 10 and Table 8-8

ISO/IEC 7816-15 attribute	ISO/IEC 7816-15 sub-attribute(s)	ISO/IEC 24727-3		Comments
		Description	ACL items	
	accessControlRule.securityCondition	Ref to DID		May be authId referring to a CIO within the cryptographic information application (DF.CIA)
ClassAttributes	classAttributes.applicationName			Either applicationName or application AID. Shall not be NULL.
TypeAttributes	<unused>			NULL
ISO/IEC 7816 DO container for each target resource of the CardApplication (DataSet Name, DID Name)				
CommonObjectAttributes	commonObjectAttributes.label	DataSet or DID Name		The target name shall be ASCII encoded with a prefix denoting the target type concatenated with the actual target name.
	accessControlRule.accessMode	<unused>		Refer to the named target
	accessControlRule.securityCondition	<unused>		Refer to the named target
ClassAttributes		<unused>		
TypeAttributes	efidOrTagChoice	location of the actual resource (EF,DF, ADF, DO) hosting the named target.		File Id or extended Path (acc. ISO/IEC 7816-15:2004/AM2) to the named target resource

[1] In case the Action is controlled by a logical combination of DIDs, several consecutive securityCondition values may provide different security conditions applying to the same access Mode denoting the SAL API Action. The logical operation combining the DIDs is AND or OR depending on the ASN.1 encoding of the securityConditions.