DRAFT AMENDMENT ISO/IEC 24727-4:2008/DAM 1

ISO/IEC JTC **1** Secretariat: **ANSI**

Voting begins on **2012-04-23** Voting terminates on **2012-09-23**

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОММИСИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

ICS 35.240.15

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

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 24727-4 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: Application programming interface (API) administration*
— *Part 5: Testing*
— *Part 6: Registration authority procedures for the authentication protocols for interoperability*

Introduction

ISO/IEC 24727 is a set of programming interfaces for interactions between integrated circuit cards (ICCs) and external applications to include generic services for multi-sector use. The organization and the operation of the ICCs conform to ISO/IEC 7816-4.

ISO/IEC 24727 is relevant to ICC applications desiring interoperability among diverse application domains. ISO/IEC 7498-1:1994 is used as the layered architecture of the client-application to card-application connectivity. That is, the client-application, through the application interface, assumes that there is a protocol stack through which it will exchange information and transactions among card-applications using commands conveyed through the message structures defined in ISO/IEC 7816. The semantics of action requests through the interface defined in ISO/IEC 24727-3 refers to application protocol data units (APDUs) as characterized through the interface defined in ISO/IEC 24727-2, and in the following International Standards:

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

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

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

The goal of ISO/IEC 24727 is to maximize the applicability and solution space of software tools that provide application interface support to card-aware client-applications. This effort includes supporting the evolution of card systems as they become more powerful, peer-level partners with existing and future applications while minimizing the impact to existing solutions conforming to ISO/IEC 24727.

By conforming to this part of ISO/IEC 24727, interoperable implementations of ISO/IEC 24727-3 and ISO/IEC 24727-2 can be realized. Implementation details are not defined within this part of ISO/IEC 24727; it is assumed that an implementation conforms to an accepted security policy. The specific security policy is outside the scope of ISO/IEC 24727

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 shall be updated and/or amended. The rules governing the use of various marshalling/un-marshalling procedures shall be aligned with the amendment to ISO/IEC 24727-3.

Identification cards — Integrated circuit card programming interfaces —

Part 4:
Application programming interface (API) administration

AMENDMENT 1

All new or changed text in this amendment is underlined in the clauses being replaced. All removed text will show with a strike-through. When merging all such text into the base standard, the underlining is to be removed and any words showing a strike-through should be removed. In the case of entirely new annexes relative to the base standard will be so indicated by a statement at the beginning of the annex. In that case, the "underline" convention will NOT be used. Rather, it is to be interpreted that the entire annex wording is to be merged into the base document.

[Editor's 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.]

1 Scope

ISO/IEC 24727 defines a set of programming interfaces for interactions between integrated circuit cards and external applications to include generic services for multi-sector use.

This part of ISO/IEC 24727 standardizes the connectivity and security mechanisms between the client-application and the card-application.

It specifies API-Administration of service-independent and implementation-independent ISO/IEC 24727 compliant modules, including security, that enables action requests to a specific card-application of an ICC such that, when coupled to data model and content discovery operations, the card-application can be used by a variety of client-applications.

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

This clause is a new addition to ISO/IEC 24727-4.

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 Exclusion

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.

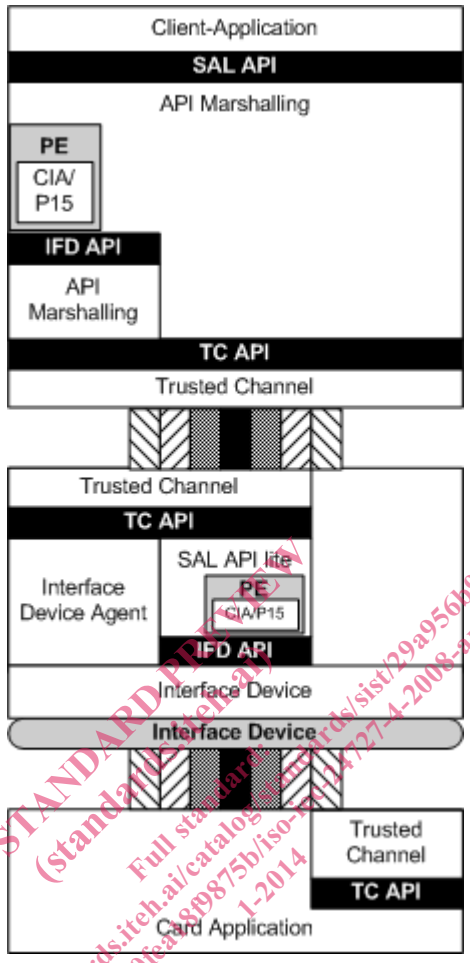

Figure 7-1 —Network stack

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 (see Annex below)
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 :

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	↔	↔	↔		↔	↔

Table 7-1 – Command Progression

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