USENIX
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

The following paper was originally published in the

USENIX Workshop on Smartcard Technology

Chicago, Illinois, USA, May 10–11, 1999

PKCS #15—
A Cryptographic-Token Information Format Standard

Magnus Nyström
RSA Laboratories

For more information about the USENIX Association:
Phone: 1 510 528 8649 FAX: 1 510 548 5738
Email: office@usenix.org WWW: http://www.usenix.org

PKCS #15 – A Cryptographic Token Information Format Standard

Magnus Nyström
RSA Laboratories, Bedford MA 01730, USA
E-mail: magnus@rsa.com

Abstract

We identify the need for a portable format for storage of user credentials (certificates, keys) on cryptographic tokens such as integrated circuit cards (IC cards). Given this need, a recent proposal in the area, RSA Laboratories' PKCS #15 is described and compared with previous and related work.

1 Background and Motivation

Cryptographic tokens, such as Integrated Circuit Cards (IC cards or "smart cards"), are capable of providing a secure storage and computation environment for a wide range of user credentials such as keys, certificates and passwords. Because of this, it is widely recognized (cf. [2], [15] and [25]) that they offer great potential for secure identification of users of information systems and electronic commerce applications. For a general introduction to IC cards and their use, cf. [3] or [17].

Unfortunately, the use of these tokens for authentication and authorization purposes is hampered by the lack of interoperability at several levels (cf. [1]). First, the industry lacks standards for storing a common format of digital credentials (keys, certificates, etc.) on them. This has made it difficult to create applications that can work with credentials from a variety of technology providers. Attempts to solve this problem in the application domain invariably increase costs for both development and maintenance. They also create a significant problem for end-users since credentials are tied to a particular application running against a particular application-programming interface to a particular hardware configuration.

Second, mechanisms to allow multiple applications to effectively share digital credentials have not yet reached maturity. While this problem is not unique to cryptographic tokens – it is already apparent in the use of certificates with World Wide Web browsers, for example – the limited room on many tokens together with the consumer expectation of universal acceptance will force credential sharing on credential providers. Without agreed-upon standards for credential sharing, acceptance and use of them both by application developers and by consumers will be muted.

To optimize the benefit to both the industry and end-users, it is important that solutions to these issues be developed in a manner that supports a variety of operating environments, application programming interfaces, and a broad base of applications. Only through this approach can the needs of constituencies be supported and the development of credentials-activated applications encouraged, as a cost-effective solution to meeting requirements in a very diverse set of markets.

The purpose of the work we describe in this paper, PKCS #15 [19], has therefore been to:

- Enable interoperability among components running on various platforms (platform neutral);
- Enable applications to take advantage of products and components from multiple manufacturers (vendor neutral);
- Enable the use of advances in technology without rewriting application-level software (application neutral); and
- Maintain consistency with existing, related standards while expanding upon them only where necessary and practical.

By fulfilling these objectives, PKCS #15 is a first step to ensure that token-holders will be able to use their cryptographic tokens to electronically identify themselves to any application regardless of the application's token interface. The ultimate goal is a situation in which a token-holder can use any card from any manufacturer to identify himself or herself to any application running on any platform.

2 Related Work

2.1 DC/SC

"Digital Certificates on Smart Cards," DC/SC [1], was a collaborative effort mainly between CertCo, Litronics

and GemPlus, initiated to facilitate the interoperability of applications using digital certificates stored on IC cards. DC/SC concentrated on the problem of finding all digital certificates stored on a particular user's IC card. The proposed solution was to add an extra elementary file at the root level on the card system (ISO/IEC 7816-4 compliant IC cards were assumed), in which applications would read and write information about known certificates on the card.

The objective of DC/SC was very similar to PKCS #15's objective: To enhance portability of user credentials stored on IC cards. The main difference was perhaps that DC/SC only concentrated on certificates and did not intend to create a new card application for general credential storage. DC/SC eventually folded its work into the SEIS specification.

2.2 SEIS

SEIS, Secured Electronic Information in Society, is a Sweden-based non-profit association. One of SEIS' most important projects has been to specify an Electronic Identity IC card with an Electronic ID Application. This includes means for secure, electronic authentication of the cardholder; generation of legally acceptable Digital Signatures; and support for session key exchange, e.g. protection of message confidentiality. The intention is that these functions will be implemented using "off the shelf" IC cards. The usage is intended for security services both within, as well as between, organizations.

So far, two specifications have been developed which relate closely to PKCS #15: SEIS S1 [21] and SEIS S4 [22]. SEIS S1 is a detailed definition of an electronic identity card application. It specifies where and how information about certificates, keys and PINs shall be stored on a compliant IC card. SEIS S4 is a profile that puts some further restrictions on the format and the size of used keys.

Being a pure electronic identification application based on public-key technology, SEIS S1 and S4 do not contain support for any other key types than private RSA keys and supports only X.509 [10] certificates. Furthermore, there is no support for general security-related data objects.

Although it is more generic, PKCS #15 is a continuation of the SEIS work. The concept of a standardized, generic IC card format is a cornerstone of the SEIS architecture.

2.3 The WAP consortium

WAP, the Wireless Application Protocol Forum [12], is a consortium of companies involved in creating a de-facto standard for wireless information and telephony services on digital mobile phones and other wireless terminals. A part of this work is to enable secure identification of subscribers to these services. For this reason, subscribers will be issued IC cards (SIM cards, "Subscriber Identification Module"), which are to be inserted in phones, and the *WAP Identity Module Specification* (WIM) [26] has therefore basically the same objectives as PKCS #15. The current draft version, v0.7 is in fact designed as a PKCS #15 profile.

2.4 Other related work

This section contains a survey of other specification-, standardization- and product-efforts, related to PKCS #15.

2.4.1 The PC/SC specification

The *Interoperability Specification for ICCs and Personal Computer Systems* [20], or PC/SC for short, is a workgroup formed by leading IC card and personal computer vendors such as GemPlus, Schlumberger and Microsoft. The intention was to develop a specification that could facilitate the interoperability necessary to allow Integrated Circuit Card (ICC) technology to be effectively utilized in the PC environment, and in a manner which would "support both existing and future IC card-based applications" [20].

Part 8 of the specification, "Recommendations for ICC Security and Privacy Devices" [16], does mention (without binding requirements) a few formats for storage of information, but other than this, the PC/SC specification does not deal with the contents of the IC card itself. Hence, credential portability is not covered by the PC/SC specification. The intention with PC/SC is that each platform the user accesses with his IC card will have a *service provider* interface installed for that particular card. The drawback of this is that it forces cardholders to choose particular cards and vendors. One additional problem with this architecture is that it relies on the IC card vendor for providing both the card interface and the functional interface. An alternative solution would have been to separate this into two layers: one handling the card interface and one handling the format interface. With separated layers and an open card format standard, a generic PKCS#15 layer could be

installed and card layers added for each card the user/customer has. A user could choose any card type, have it personalized by any application vendor which personalizes cards in accordance with PKCS#15 and use it on any PC/SC system which has a PKCS#15 format service provider installed and, in addition, a service provider for that particular card type.

2.4.2 OpenCard Framework

The *OpenCard Framework* (OCF) [23] provides a common programming interface for both the smart card reader and the application on the card. By basing the architecture on Java technology, the intention is to receive enhanced portability and interoperability. The version 1.1 specification also enables, through an adaptation layer, interaction with existing PC/SC 1.0 supported reader devices.

Although the OCF simplifies the task for application builders, and has made explicit the distinction between a card layer and a format layer, it does not deal with the problem of credential portability. The *ApplicationManagement* layer is a rudimentary service layer in principle only supporting application listing and selection. OCF applications still need explicit knowledge of the location and structure of files contained in card applications (*card layout* in OCF terminology). Therefore, a cardholder will still be restricted to use his proprietary-formatted card only on those platforms, for which a card-application aware OCF application has been installed, which probably will be within a particular domain.

2.4.3 The JavaCard specification

The *JavaCard* specification [24], developed by Sun Microsystems, defines a subset of the Java language for use on IC cards and other embedded systems. The intention is to simplify card-application development by offering a familiar high-level programming language interface to developers of card-side applications. By implementing a version of the Java Virtual Machine on top of existing IC card operating systems, applications become portable and easier to develop.

The specification does not deal with card layouts or information formats, instead it defines a framework for creating applications. Even though it is possible to implement PKCS #15 on a JavaCard, it would probably be more natural to define a generic 'Electronic ID' JavaCard application (*cardlet* in established terminology). The JavaCard specification of such an application would instead of specifying internal file formats define the command interface used to store and retrieve security-related information as well as to execute cryptographic commands.

2.4.4 MultOS

MULTOS [13] is another attempt to simplify card-side application development and portability of card-side applications. MULTOS is a card operating system that implements an Application Abstract Machine (AAM). The intention is that by using services offered by the AAM, application programmers do not need specific knowledge about underlying hardware.

Similar to the JavaCard case, it probably makes more sense to define a standard "MULTOS Electronic ID application", specifying the command interface rather than actual file formats in the MULTOS case.

3 An Overview of PKCS #15

Having described the problem background and related work, we now proceed to describe the standard itself, and requirements that have influenced its design.

3.1 Design Goals

Token neutral

- In order not to favor any particular brand or type of IC card, PKCS #15 has been designed in such a way that the standard can be implemented on any IC card with basic ISO/IEC 7816 compatibility. For example, no assumptions about IC card commands except those defined in ISO/IEC 7816-4 [4] have been made. In fact, it should even be possible to implement it on memory cards and software tokens.

- In order to support memory cards and tokens that do not have built-in support for encryption, provision has been made to allow stored objects to be encrypted and/or integrity-protected.

- Finally, in order to support tokens which do not have the notion of "files", PKCS #15 has been designed to allow all information to be stored in one contiguous block.

Standards compliant

- In order to be aligned with ISO/IEC 7816-5 [5] and ISO/IEC 7816-6 [6], the information format (as

seen by an interface communicating with the card) has been defined in ASN.1 [9].

- The format allows storage of information concepts such as *security environments*, defined in ISO/IEC FCD 7816-8 [7].

- The "object-oriented" approach chosen for PKCS #11 [18], treating keys, certificates and other data as objects with attributes and values, has been adopted for PKCS #15 as well. It has a proven record and eases PKCS #11-based implementations.

Self-contained

- Given an IC card with a PKCS #15 application on the chip, in order to be able to use the card for secure identification, it must be possible for a host-side application to find out which algorithms, keys and certificates that are present on the card. This led to the inclusion of *TokenInfo* files and *Object Directory* files, see Section 3.2.

- Applications also need to know how objects are protected, and procedures for accessing them. They also need to know which objects that are possible to update and which are not. This requirement has been met by introducing appropriate security attributes and *authentication objects* that can be referenced from protected objects like keys. Authentication objects are stored in *Authentication Object Directory* files, see Section 3.2.

Flexible and modular structure

- It should not be necessary to store all PKCS #15-relevant objects in the PKCS #15 application. Sometimes a certificate may already exist on the token when the PKCS #15 application is stored on it, for example. Therefore, an extra level of indirection has been introduced, giving the PKCS #15 application the ability to refer to objects like certificates stored in other dedicated files (or at other places like LDAP accessible directories).

- When dealing with IC cards, the problem of *card tearing* (cf. [3], pp. 175–176) has to be considered. This means that every update needs to be as atomic as possible, and if interrupted due to premature card removal, should not leave the card in an inconsistent state. This led to the decision to make the file structure record-oriented and modular.

- Since it is anticipated that PKCS #15 will be used in a number of applications for different purposes and with different security requirements, no particular requirements in terms of access restrictions have been mandated; an appendix giving general recommendations has been provided, however. This appendix also builds on ideas expressed in ISO/IEC CD 7816-9 [8].

3.2 File Structure and Motivations

The content of the PKCS #15 dedicated file is a bit dependent on the type of IC card and its intended use, but the following file structure is likely to be the most common, especially when the card is intended to be used for identification/authentication purposes:

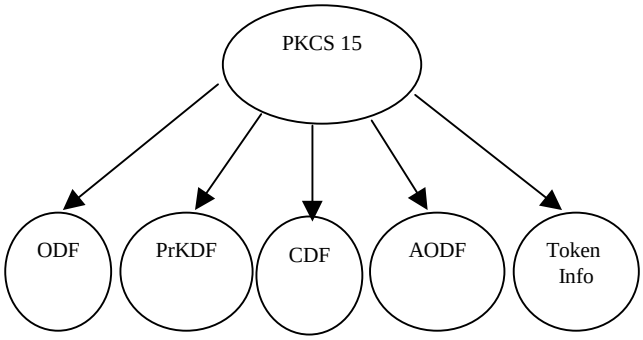

Figure 1: Contents of DF(PKCS15)

The contents and purpose of major elementary files in the PKCS #15 directory are described below.

The Object Directory File, ODF

The mandatory elementary file ODF (Object Directory File) consists of pointers to other elementary files (PrKDFs, PuKDFs, SKDFs, CDFs, DODFs and AODFs), each one containing a directory over PKCS #15 objects of a particular class. The ODF therefore has a record-oriented structure, with each record merely being a pointer to another directory file.

Cryptographic Key Directory Files, PrKDFs, SKDFs and PuKDFs

These elementary files can be regarded as directories of keys known to the PKCS#15 application. PrKDFs contain information about private keys, PuKDFs contain information about public keys and SKDFs contain information about secret (symmetric) keys. They are all

optional, but at least one file of a particular kind must be present on an IC card which contains keys (or references to keys) of that particular kind, known to the PKCS#15 application. The files contain general key attributes such as labels, key usage restrictions, identifiers, type of algorithm, key size (if applicable), etc. Furthermore, they contain pointers to the keys themselves.

Certificate Directory Files CDFs

These elementary files can be regarded as directories of certificates known to the PKCS#15 application. They are optional, but at least one CDF must be present on an IC card which contains certificates (or references to certificates) known to the PKCS#15 application. They contain general certificate attributes such as labels, identifiers, certificate type, etc. They also contain pointers to the certificates themselves. When a certificate contains a public key corresponding to a private key that is also known to the PKCS #15 application, the certificate and the private key will share a common identifier. This simplifies look-up of the private key given the certificate and vice versa.

Trusted Certificate Directory Files

These elementary files have the same syntax as ordinary CDFs, but only contain trusted certificates. In the context of PKCS #15, “trusted certificates” are CA certificates not possible to be replaced by the cardholder.

Authentication Object Directory Files AODFs

These elementary files can be regarded as directories of authentication objects (e.g. PINs) known to the PKCS#15 application. They are optional, but at least one AODF must be present on an IC card, which contains authentication objects restricting access to PKCS#15 objects. They contain generic authentication object attributes such as (in the case of PINs) allowed characters, PIN length, PIN padding character, etc. Furthermore, they contain pointers to the authentication objects themselves (e.g. in the case of PINs, pointers to the directory in which the PIN file resides). Authentication objects are used to control access to other objects such as keys. Each object in this file has a unique reference number, which is used for cross-reference purposes from e.g. the PrKDFs to link keys with authentication objects.

Data Object Directory Files, DODFs

These files can be regarded as directories of data objects (other than keys or certificates) known to the PKCS#15 application. They are optional, but at least one DODF must be present on an IC card which contains such data objects (or references to such data objects) known to the PKCS#15 application. They contain general data object attributes such as identification of the application to which the data object belongs, whether it is a private or public object, etc. In addition to this, they contain pointers to the data objects themselves.

The TokenInfo file

The mandatory TokenInfo elementary file contains generic information about the token as such and its capabilities, as seen by the PKCS #15 application, e.g. supported algorithms, token serial number, etc. In order to save some storage space, provisions for cross-referencing algorithm information from the PrKDFs, PuKDFs and SKDFs to this file has been made.

4 An Example Application

PKCS #15 v1.0 contains a profile of the information format for electronic identification purposes. This profile specifies the use of two private keys, two user certificates and two PINs. Each private key is to be protected with a separate PIN and also linked to a corresponding certificate. The intention is that the cardholder use one key (and associated PIN) for general authentication purposes and key exchanges, and the other key (and associated PIN) strictly for non-repudiation (or digital signature) purposes, like signing documents.

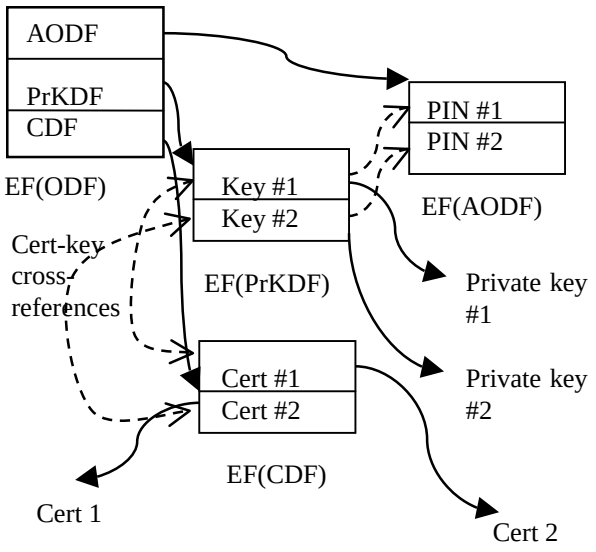

Figure 2: Logical file structure of PKCS #15's Electronic ID profile

If some 3rd-party application-specific data is added to a token containing the Electronic ID profile of PKCS #15[1], the token will be useable not only for general identification purposes but for that 3rd-party application's purposes as well. An example of this is the profile suggested in WIM [26], in which the cardholder will be able to use the card not only as a cellular phone card enabling phone calls and related services, but also enabling Internet access from any PKCS #15 aware application. If PKCS #15 is well received, it is likely that similar cases will occur in a number of other environments, like on-line banking, as secure credit cards, etc.

5 Summary and Future Work

PKCS #15 was announced in September 1998. Shortly thereafter, the SEIS specifications were adopted as national standards in Sweden. At the same time, the WAP Forum submitted its first Identity Module draft containing an IC card file structure. Coordinating with and using experiences from these efforts as well as other ones has been and continues to be an important part of this project. The first official version of PKCS #15 is expected to be available in April 1999.

The current draft version of the standard does not deal at all with more advanced IC cards like JavaCards or MULTOS cards. As mentioned in this paper, since these cards are able to store and execute more complex applications, the equivalent of PKCS #15 in this environment would probably be a "standard electronic ID application," defining a service interface rather than an information format.

The current lack of standardization of security-related commands for IC cards presents a more serious problem with regard to interoperability. In order to achieve interoperability with PKCS #15, an application needs not only to have a format layer implementing support for the PKCS #15 format, but also a card adaptation layer, implementing support for proprietary security-related commands set for a range of IC cards. If or when a set of such commands becomes formally standardized and adapted by IC card vendors, this card adaptation layer would become superfluous. ISO/IEC 7816-8 [7] is intended to solve the problem of standardization in this area, and hopefully card vendors will adjust to this standard in a timely manner.

ISO/IEC JTC1 SC17 has recently discussed [11] a possible new work item defining an Electronic ID Application for identification cards. While the prospects for such a work item remain unclear, PKCS #15 could clearly be one candidate for this format. PKCS #15 has also been suggested as the card format for the national identity IC card in Finland [14].

6 Acknowledgement

The author wishes to acknowledge valuable suggestions and comments from Zoltan Kelemen, Security Dynamics, and Burt Kaliski, RSA Laboratories.

7 References

[1] DC/SC, "Interoperability Specification for Digital Certificates – Storage of Digital Certificates on ICCs," draft version 0.2, Digital Certificates on Smart Cards Working Group, 1998.

[2] S. Elliot and C. Loebbecke, "Smart card based Electronic Commerce: Characteristics and Roles," *Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98) (IEEE)*, 1998.

[3] S. Guthery S and T. Jurgensen, *Smart Card Developer's Kit*, Macmillian Technical Publishing, 1998.

[1] Preferably in the form of PKCS#15 *Data Objects*.

[4] ISO/IEC 7816-4, "Information Technology – Identification cards – Integrated Circuit(s) cards with contacts – Part 4: Interindustry commands for interchange," International Organization for Standardization, 1995.

[5] ISO/IEC 7816-5, "Information Technology – Identification cards – Integrated Circuit(s) cards with contacts – Part 5: Numbering systems and registration procedure for application identifiers," International Organization for Standardization, 1994.

[6] ISO/IEC 7816-6, "Information Technology – Identification cards – Integrated Circuit(s) cards with contacts – Part 6: Interindustry data elements," International Organization for Standardization, 1996.

[7] ISO/IEC 7816-8 Final Committee Draft, "Information Technology – Identification cards – Integrated Circuit(s) cards with contacts – Part 8: Security related interindustry commands," International Organization for Standardization, 1998.

[8] ISO/IEC 7816-9 Committee Draft, "Information Technology – Identification cards – Integrated Circuit(s) cards with contacts – Part 9: Additional interindustry commands and security attributes," International Organization for Standardization, 1998.

[9] ISO/IEC 8824-1, "Information Technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation," International Organization for Standardization, 1997.

[10] ISO/IEC 9594-8, "Information Technology – Open Systems Interconnection – The Directory: Authentication framework," International Organization for Standardization, 1997.

[11] ISO/IEC JTC1/SC17, "Report of the 11th Plenary Meeting of ISO/IEC JTC1/SC17, Berlin, Germany, 1998-10-21/23," Document N1429, International Organization for Standardization, October 1998.

[12] P. King, "The Wireless Application Protocol (WAP)," *Proceedings of RSA Data Security Conference '99*, San Jose, USA, 1999.

[13] H. Kingdon, "MULTOS – The card for every lifestyle," *Proceedings of CardTech/SecurTech '98 West*, San Jose, USA, 1998.

[14] M. Kontio, Personal communication, 1999.

[15] T. Monk and H. Dreifus, *Smart Cards: A Guide to Building and Managing Smart Card Applications*, John Wiley & Sons, 1997.

[16] PC/SC, "Interoperability Specification for ICCs and Personal Computer Systems – Part 8: Recommendations for ICC Security and Privacy Devices," The PC/SC Workgroup, December 1997 (Available from http://www.smartcardsys.com).

[17] W. Rankl, W. Effing, *Smart Card Handbook*, John Wiley & Sons, June 1997.

[18] RSA Laboratories, "PKCS #11: Cryptographic Token Interface Standard," version 2.01, December 1997 (Available from ftp://ftp.rsa.com/pub/pkcs/pkcs-11).

[19] RSA Laboratories, "PKCS #15: Cryptographic Token Information Format Standard," version 1.0, April 1999 (Available from ftp://ftp.rsa.com/pub/pkcs/pkcs-15).

[20] P. Sarlin, "PC/SC Technical Overview," *Proceedings of CardTech/SecurTech West*, San Jose, USA, 1996.

[21] SEIS, "SEIS Cards – Electronic ID Application v2.0," The Association for Secured Electronic Information in Society, 1998 (Available from http://www.seis.se).

[22] SEIS, "SEIS Cards – EID Implementation Profiles v2.0," The Association for Secured Electronic Information in Society, 1998 (Available from http://www.SEIS.se).

[23] F. Seliger, "OCF: Java API für e-business Anwendungen mit Smart Cards", *Proceedings of OOP'99*, Münich, Germany, 1998.

[24] Sun Microsystems, Inc., "The JavaCard 2.1 Platform Specifications," Sun Microsystems, 1999 (Available from http://java.sun.com/products/javacard).

[25] E. Turban, D. McElroy, "Using Smart Cards in Electronic Commerce," *Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98) (IEEE)*, 1998.

[26] WAP, "Wireless Application Protocol – Identity Module Specification – Part: Security", draft v0.7, Wireless Application Protocol Forum, 1999.