付録 3
アプリケーションインターフェースに関する標準

付録 3-1
アプリケーションインターフェース仕様

PKCS#11: IWG Conformance Profile Specification

International Working Group for ASIA PKI Forum

November 13, 2003

Table of Contents

1 Introduction

The objective of this specification is to ensure individual implementers of the PKCS#11 v2.11 standards can interoperate in Asia region. To achieve these objective, subsets of the PKCS #11 specifications have been defined and are detailed in the form of profiles. The profiles specify which calls must or should be implemented in order to be considered compliant. These profiles can then be used in the production of conformance testing tools that allow vendors to certify their compliance.

References and related documents

- RSA Laboratories PKCS #1 v2.0: RSA Cryptography Standard.
- RSA Laboratories PKCS #11 v2.11: Cryptographic Token Interface Standard.
- RSA Laboratories PKCS #12 v1.0: Personal Information Exchange Syntax Standard.
- RSA Laboratories PKCS #15 v1.1: Cryptographic Token Information Format Standard

PKCS#11 version compatibilities

This document is originally supporting PKCS#11 v2.11, but none of advanced functionalities of this version are compulsory required. So, all versions above 2.01 are compatible for this profile.

2 Definitions

ANSI: American National Standards Institute. An American standards body.

Application: The implementation of a well-defined and related set of functions that perform useful work on behalf of the user. It may consist of software and or hardware elements and associated user interfaces.

Application provider: An entity that provides an application.

ASN.1 object: Abstract Syntax Notation object as defined in ISO/IEC 8824. A formal syntax for describing complex data objects.

Function: A process accomplished by one or more commands and resultant actions that are used to perform all or part of a transaction.

ICC: Integrated Circuit Card. Another name for a smart card.

ISO: International Organization for Standardization

Password: Data that may be required by the application to be presented to the card by its user before data can be processed.

PIN: Personal Identification Number.

Provider: Authority who has or who obtained the rights to create the MF or a DF in the card.

Token: In this specification, a portable device capable of storing persistent data.

Tokenholder: Analogous to cardholder.

Uniform Resource Identifiers: a compact string of characters for identifying an abstract or physical resource. Described in RFC 2396.

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in IETF RFC 2119.

3 Symbols and Abbreviations

BER Basic Encoding Rules
DER Distinguished Encoding Rules
OID Object Identifier
PKCS Public Key Cryptography Standard
URL Uniform Resource Locator (a class of uniform resource identifiers)

4 General Overview

This document defines profiles for the PKCS #11 v2.11 specification for the interoperability in Asia region. These profiles specify the function calls necessary and assumptions made for compliance. This section outlines the model used to specify a PKCS #11 profile. Two individual profiles are specified in later sections.

4.1 Profile Model

Scope: The scope will define the purpose and limitations of the profile.

Assumptions: Explicitly states any assumptions necessary for profile conformance.

Mechanisms: States the PKCS #11 v2.11 mechanism that must be available.

Additional APIs: Specifies which additional API's must be implemented for any application that claims conformance.

Sessions: Defines the session requirements.

Threading: Defines the thread requirements.

Key Issues: Defines any key size restrictions or requirements including specific key types that must be supported.

4.2 Base APIs

The following is a basic set of APIs for IWG PKCS#11 Conformance Profile. These APIs are detailed in the PKCS #11 V2.11 specification. And these base APIs are also compliant to the base APIs of "PKCS#11: Conformance Profile Specification" written by RSA Laboratories.

C_GetFunctionList
C_Initialize
C_Finalize
C_GetInfo
C_GetSlotList
C_GetSlotInfo
C_GetTokenInfo
C_GetMechanismList
C_GetMechanismInfo
C_OpenSession
C_CloseSession
C_CloseAllSessions
C_FindObjectsInit
C_FindObjects
C_FindObjectsFinal
C_GetAttributeValue

5 The specification of key management function 1

5.1 Scope

This profile specifies the usage of multiple slots / keys in application

5.2 Assumptions

- Key generation and certification **SHOULD** be complete.
- The Location of intermediate and root certificates is undefined by the profile.
- A unique non-null CKA_ID value **MUST** exist and have proper associations for all keys and certificates.
- Private key **MUST** be a private object.
- Secret key **MUST** be a private object.
- Certificate **MUST** be a public object.
- Public key **MUST** be a public object
- X.509 keyUsage flags for Public and Private Key **SHOULD** be mapped to the PKCS#11 attributes for public key and private key respectively, as specified in the corresponding mapping table.

5.3 Sessions

MUST support one R/W and at least one simultaneous R/O sessions.

5.4 Threading

Base library locking is not **REQUIRED**.

5.5 Template Requirements

Each template **MUST** have following attributes and some attributes with certain values

Private key templates:

CKA_CLASS is CKO_PRIVATE_KEY
CKA_KEY_TYPE is CKK_RSA
CKA_TOKEN is TRUE
CKA_PRIVATE is TRUE
CKA_SENSITIVE is TRUE,
CKA_SIGN is TRUE when the key is for digital signature
CKA_UNWRAP is TRUE when the key is for key encipherment
CKA_DECRYPT is TRUE when the key is for data encipherment
CKA_ID,
CKA_MODULUS,
CKA_PUBLIC_EXPONENT,

CKA_PRIVATE_EXPONENT,
CKA_PRIME_1,
CKA_PRIME_2,
CKA_EXPONENT_1,
CKA_EXPONENT_2,
CKA_COFFICIENT

Public key templates:

CKA_CLASS is CKO_PUBLIC_KEY,
CKA_KEY_TYPE is CKK_RSA,
CKA_TOKEN is TRUE,
CKA_VERIFY is TRUE when the key is for digital signature,
CKA_WRAP is TRUE when the key is for key encipherment
CKA_ENCRYPT is TRUE when the key is for data encipherment
CKA_MODULUS,
CKA_PUBLIC_EXPONENT

Certificate templates:

CKA_CLASS is CKO_CERTIFICATE,
CKA_CERTITIFICATE_TYPE is CKC_X_509,
CKA_TOKEN is TRUE,
CKA_SUBJECT,
CKA_ID,
CKA_VALUE

Secret key templates:

CKA_CLASS is CKO_SECRET_KEY,
CKA_KEY_TYPE is CKK_DES3,
CKA_TOKEN is TRUE,
CKA_PRIVATE is TRUE
CKA_ENCRYPT is TRUE,
CKA_DECRYPT is TRUE,
CKA_VALUE

5.6 Key Issues

Key sizes **MUST** be supported for the range of 512 to 2048 bits in increments restricted by the specification.

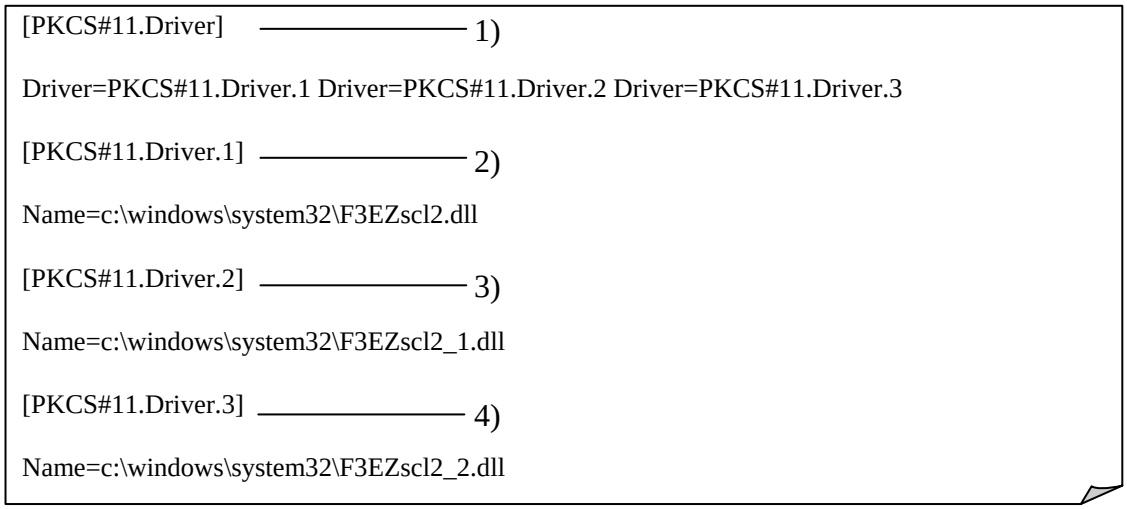
5.7 Additional APIs

The following additional APIs as defined in PKCS v2.11 **MUST** be supported.

C_Login
C_Logout

5.8 pkcs11.ini file format

The pkcs11.ini file format is as follows.



The term of 1) means the slot list.

The term of 2) to 4) means slot information. In each term, the information of the PKCS#11 library (file name and stored folder) that manages the slots is written.

This “pkcs11.ini” file **MUST** be stored in c:\Program Files\iwg\ directory.

6 The specification of key management function 2

6.1 Scope

This profile specifies an application and token that supports wrapping, unwrapping, certificate, basic private key and secret key storage.

6.2 Mechanisms

The following mechanisms **MUST** be supported

CKM_RSA_PKCS
CKM_DES3_KEY_GEN
CKM_DES3_CBC

6.3 Algorithms

The following algorithms **MUST** be supported.

CKA_WRAP
CKA_UNWRAP

6.4 Additional APIs

The following additional APIs as defined in PKCS v2.11 **MUST** be supported.

C_CreateObject
C_DestroyObject
C_GenerateKey
C_WrapKey
C_UnwrapKey

7 The specification of encryption / decryption function

7.1 Scope

This profile specifies an application and token that supports encryption, decryption, certificate, basic private key and secret key storage.

7.2 Mechanisms

The following mechanisms **MUST** be supported

CKM_RSA_PKCS
CKM_DES3_KEY_GEN
CKM_DES3_CBC

7.3 Algorithms

The following algorithms **MUST** be supported.

CKA_ENCRYPT
CKA_DECRYPT

7.4 Additional APIs

The following additional APIs as defined in PKCS v2.11 **MUST** be supported.

C_CreateObject
C_DestroyObject
C_EncryptInit
C_Encrypt
C_EncryptUpdate
C_EncryptFinal
C_DecryptInit
C_Decrypt
C_DecryptUpdate
C_DecryptFinal
C_GenerateKey

8 The specification of signing function

8.1 Scope

This profile specifies an application and token that supports signing, verification, digesting, certificate, basic private key and secret key storage.

8.2 Mechanisms

The following mechanisms **MUST** be supported

CKM_RSA_PKCS
CKM_SHA_1

8.3 Algorithms

The following algorithms **MUST** be supported.

CKA_SIGN
CKA_VERIFY

8.4 Additional APIs

The following additional APIs as defined in PKCS v2.11 **MUST** be supported.

C_SignInit

C_Sign
C_VerifyInit
C_Verify
C_DigestInit
C_Digest
C_DigestUpdate
C_DigestFinal