PKCS #11: Conformance Profile Specification

RSA Laboratories

October 1, 2000

Table of Contents

003-903076-100-000-000

1 Introduction

The broad scope and wide adoption of PKCS #11 has made it necessary to provide a mechanism that ensures individual implementers of the PKCS #11 v2.10 standard can interoperate for at least some specific subset of the specification. To accomplish this task subsets of the PKCS #11 specification 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.

1 References and related documents

- RSA Laboratories PKCS #1 v2.0: RSA Cryptography Standard.
- RSA Laboratories PKCS #11 v2.10: Cryptographic Token Interface Standard.
- RSA Laboratories PKCS #12 v1.0 (DRAFT): Personal Information Exchange Syntax Standard.

- RSA Laboratories PKCS #15 v1.1: Cryptographic Token Information Format Standard

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.

CHV: CardHolder Verification. Also called the PIN. Typically a 4 to 8 digit number entered by the cardholder to verify that the cardholder is authorized to use the card.

Cryptogram: Result of a cryptographic operation.

Data unit: The smallest set of bits that can be unambiguously referenced. Defined in ISO/IEC 7816-4.

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. See CHV.

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

Template: Value field of a constructed data object, defined to give a logical grouping of data objects. Defined in ISO/IEC 7816-6.

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 v2.10 specification. These profiles specify the function calls necessary and assumptions made for compliance. This section outlines the model used to specify a PKCS #11 profile. 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 v2.10 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.

Template Requirements: Specifies the template requirements and implementation details.

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 list of APIs that must be supported for all applications that claim PKCS #11 compliance. These APIs are detailed in the PKCS #11 V2.10 specification.

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 RSA Asymmetric Client Signing Profile

This section contains a detailed description of the RSA Asymmetric Client Profile.

5.1 Scope

This profile specifies a host application and token that supports signing, certificate and basic private key storage.

5.2 Assumptions

- Key generation and certification is complete.
- A unique non-null CKA_ID value exists and has proper associations for all keys and certificates.
- Key/Certificate pair has appropriate signing permissions.
- Private key is a private object.

- Client Authentication Certificate is a public object.
- The Location of intermediate and root certificates is undefined by the profile.

5.3 Mechanisms

CKM_RSA_PKCS must be supported.

5.4 Algorithms

CKA_SIGN must be supported.

5.5 Additional APIs

The following additional APIs as defined in PKCS v2.10 must be supported.

C_SignInit
C_Sign (Length is restricted by the mechanism, as described in table 56 of PKCS #1)
C_Login (App cannot depend on logging in as SO, User Login must be supported)
C_Logout

5.6 Sessions

A single read only session must be supported.

5.7 Threading

Base library locking is not required.

5.8 Template Requirements

None for this profile.

5.9 Key Issues

The client side MUST support a key size of 1024 bits and SHOULD support 1536 and 2048 bit keys. The host application MUST support each of these.

6 RSA Asymmetric Acceleration Profile

This section contains a detailed description of the RSA Asymmetric Acceleraton Profile.

6.1 Scope

This profile specifies a mechanism to accelerate public key cryptographic operations and public key management. Key storage, load and wrap unwrap procedures are outside of the scope of this profile.

6.2 Assumptions

- Token is in an Initialized state.
- Key generation has been performed

6.3 Mechanisms

- CKM_RSA_PKCS
- CKM_RSA_PKCS_KEY_PAIR_GEN

6.4 Algorithms

The following algorithms must be supported:

CKA_SIGN
CKA_VERIFY
CKA_ENCRYPT
CKA_DECRYPT
CKA_VERIFY_RECOVER

6.5 Additional APIs

The following additional APIs as defined in PKCS v2.10 must be supported.

C_CreateObject
C_GenerateKeyPair
C_SignInit

C_Sign
C_VerifyInit
C_Verify
C_VerifyRecover
C_VerifyRecoverInit
C_EncryptInit
C_Encrypt
C_DecryptInit
C_Decrypt
C_DestroyObject
C_SetAttributeValue
Applications must not expect to be able to set values after creation unless explicitly indicated
C_Login
Only required to support CKU_USER
C_Logout

6.6 Sessions

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

6.7 Threading

Applications must supply mutexes when they are required.

6.8 Template Requirements

C_CreateObject:
RSA Public Key
RSA Private Key
Certificate (X.509)

Private key templates:
-CKA_ID must be set same for certificate and public
-TOKEN must be supported for both settings
-Application must set SENSITIVE to TRUE if TOKEN is TRUE
-Application must set PRIVATE to TRUE if TOKEN is TRUE
-Application must not expect to be able to set EXTRACTABLE
-Tokens are not required to accept dual use attributes
-UNWRAP and DECRYPT does not constitute dual use
-Must be able to use C_SetAttributeValue with CKA_ID after creation

-Application must tolerate FALSE for CKA_SEC_AUTH
-Application must supply all CRT parameters

Public key templates:

-CKA_ID must be set same for certificate and private
-TOKEN must be supported for both settings
-PRIVATE must be set to FALSE
-Must be able to use C_SetAttribuValue with CKA_ID after creation
-Tokens are not required to accept dual use attributes
-WRAP and ENCRYPT does not constitute dual use
-VERIFY and VERIFY_RECOVER does not constitute dual use

Certificate templates:

-CKA_ID must be set the same as public and private
-Application must set TOKEN to TRUE < wording? >
-Application must set PRIVATE to FALSE < wording? >
-Token must be able to accept all certificate attributes
-Application supplied certificate attributes must be consistent with the certificate
-Token is not required to verify consistency

6.9 Key Issues

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

7 DH Asymmetric Acceleration Profile

This section contains a detailed description of the DH Asymmetric Acceleration Profile.

7.1 Scope

This profile is intended for applications that make extensive use of ephemeral DH keys and require acceleration through the use of the token. Specifically tokens that support this profile could be used by applications that support the IKE protocol. The application must be able to extract the DH shared secret from the token if it is of type generic secret.

7.2 Assumptions

- Token is in an Initialized state.

7.3 Mechanisms

- CKM_DH_PKCS_KEY_PAIR_GEN
- CKM_DH_PKCS_DERIVE

7.4 Algorithms

The following algorithms must be supported.

CKA_DERIVE

7.5 Additional APIs

The following additional APIs as defined in PKCS v2.10 must be supported.

C_GenerateKeyPair
C_DeriveKey
C_DestroyObject

7.6 Sessions

Must support at least ten R/O simultaneous sessions.

7.7 Threading

Applications must supply mutexes when they are required.

7.8 Template Requirements

C_GenerateKeyPair:
DH Public Key
DH Private Key

DH Private key templates:
-CKA_TOKEN must be supported with a value set to FALSE
-CKA_DERIVE must be supported with value set to TRUE
-CKA_VALUE_BITS must be supported
-CKA_EXTRACTABLE supported for both settings
-CKA_SENSITIVE be supported for both settings
-Application must not set CKA_DECRYPT, CKA_SIGN, CKA_SIGN_RECOVER or CKA_UNWRAP to TRUE

DH Public key templates:

-CKA_TOKEN must be set same for both private and public templates

-CKA_BASE need not be padded with leading zeros

-Application must not set CKA_ENCRYPT, CKA_VERIFY, CKA_VERIFY_RECOVER or CKA_WRAP to TRUE

C_DeriveKey:

Generic Secret Key

-CKA_TOKEN must be supported for session objects (CKA_TOKEN set to FALSE)

-CKA_EXTRACTABLE must be supported with value set to TRUE

-CKA_SENSITIVE must be supported with value set to FALSE

-CKA_VALUE_LEN must be supported with value set to the size in bytes of the DH modulus. This means that the secret key must if necessary be padded with leading zeros up to the size of the modulus

7.9 Key Issues

When CKA_VALUE_BITS is set during key pair generation, the actual number of bits in the private key generated must not be less than the value specified.

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

8 Large Application Profile

This section contains a detailed description of the Large Application Profile.

8.1 Scope

This profile is meant to be a profile that would support the needs of larger applications.

8.2 Assumptions

- The token is in an initialized state.

8.3 Mechanisms

- CKM_RSA_X509 (MUST support 1024 bit keys)
- CKM_RSA_PKCS

8.4 Algorithms

The following algorithms must be supported.

CKA_SIGN
CKA_DECRYPT

8.5 Additional APIs

The following additional APIs as defined in PKCS v2.10 must be supported.

C_GetFunctionList
C_SetPIN
C_GetSessionInfo
C_Login
C_Logout
C_CreateObject
C_DestroyObject
C_GetAttributeValue
C_SetAttributeValue
C_FindObjectsInit
C_FindObjects
C_FindObjectsFinal
C_SignInit
C_Sign
C_Encrypt
C_EncryptInit
C_DecryptInit
C_Decrypt
C_GenerateKeyPair
C_Unwrap

8.6 Sessions

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

8.7 Threading

Applications must supply mutexes when they are required.

8.8 Template Requirements

C_CreateObject:

RSA Public Key
RSA Private Key
Certificate (X.509)
Data

Private key templates:

Application must set TOKEN to TRUE

Public key templates:

PRIVATE must be set to FALSE

Certificate templates:

The application must set TOKEN to TRUE.

Data templates:

TOKEN must be supported for both settings
PRIVATE must be supported for both settings

C_SetAttributeValue:

Applications must be allowed to change the label after creation.

Applications must not expect to be able to set values after creation unless explicitly indicated.

C_Login:

Must support for both CKU_USER and CKU_SO.

8.9 Token Memory

The token memory size must be at least 8K.