

CFCA

中金金融认证中心标准

30005.01—2013

SM2 数字证书申请及应用 PKCS#11 接口调用规范

Interface specification of PKCS#11

for SM2 certificate enrollment and application

2013-06-01 发布

2013-06-01 实施

中金金融认证中心

发布

目 录

1. 范围 1

2. 规范性引用文件..... 1

3. 术语和定义 1

4. PKCS#11 接口版本 2

5. 容器、证书、密钥区分..... 2

6. SM2 数字证书申请调用 PKCS#11 接口规范 2

 6.1 SM2 数字证书申请流程..... 2

 6.2 PKCS#11 接口描述..... 3

7. SM2 数字证书导入调用 PKCS#11 接口规范 5

 7.1 SM2 签名证书导入 5

 7.2 SM2 加密证书及私钥导入流程..... 6

8. SM2 私钥及签名结构 8

 8.1 SM2 导入私钥结构 8

 8.2 SM2 签名结构 9

9. PIN 码框..... 9

10. 代码签名 9

SM2 数字证书申请及应用 PKCS#11 接口调用规范

1. 范围

本规范中，描述了通过 PKCS#11 接口实现 SM2 数字证书申请及应用时，所涉及接口的实现标准。关于 PKCS#11 支持 SM2 算法的相关定义、说明请参照《SM2 PKCS#11 规范》。本规范中未涉及的接口，请参照 PKCS#11 标准。

2. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件，凡是不注日期的引用文件，其最新版本适用于本文件。

PKCS#11 v2.20	Cryptographic Token Interface Standard
GM/T 0002-2012	SM4 分组密码算法
GM/T 0003-2012	SM2 椭圆曲线公钥密码算法
GM/T 0004-2012	SM3 密码杂凑算法
GM/T 0009-2012	SM2 密码算法使用规范
GM/T 0010-2012	SM2 密码算法加密签名消息语法规范

3. 术语和定义

数字证书

也称公钥证书，由证书认证机构(CA)签名的包含公开密钥拥有者信息、公开密钥、签发者信息、有效期以及扩展信息的一种数据结构。按用途可分为签名证书、加密证书。

公钥

非对称密码算法中可以公开的密钥。

私钥

非对称密码算法中，只能由拥有者使用的不公开密钥。

SM2 密码算法

一种椭圆曲线密码算法，密钥长度为 256 比特。

SM3 算法

一种杂凑算法，输出长度为 256 比特。

SM4 算法

一种分组密码算法，分组长度为 128 比特，密钥长度为 128 比特。

交互密钥

在本规范中，交互密钥特指由申请者产生的非对称密钥对，用于保护加密证书私钥。

4. PKCS#11 接口版本

厂商实现 P11 库时使用的头文件版本，必须使用 CFCA 修订的版本，请联系 CFCA 索取。

5. 容器、证书、密钥区分

CFCA 使用 CKA_LABEL 和 CKA_ID 两个属性字段来区分不同的容器、证书、密钥：

1. CFCA 在产生 PKCS#10 请求时会生成一串 UUID，并以该 UUID 作为容器名称来产生密钥对，并将该容器名称返回给上层应用。
2. 将容器名称作为 CKA_LABEL 属性设置到对应的证书、密钥上。
3. 将容器名称附加“#1”作为 CKA_ID 属性来标识签名证书、密钥；
将容器名称附加“#2”作为 CKA_ID 属性来标识加密证书、密钥。

6. SM2 数字证书申请调用 PKCS#11 接口规范

6.1 SM2 数字证书申请流程

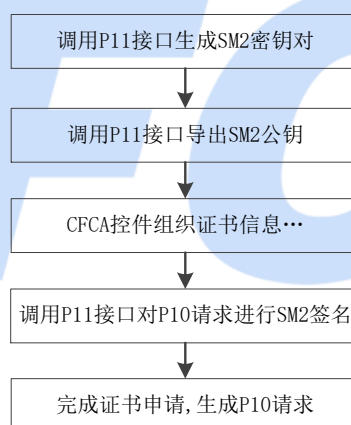


图 1 SM2 数字证书申请流程

SM2 数字证书申请流程如下：

- 1、调用 PKCS#11 接口，生成 SM2 密钥对。若为双证书请求，则需生成两对密钥对。
本步骤中需调用的接口：C_GeneratekeyPair。
- 2、调用 PKCS#11 接口，导出 SM2 公钥。若为双证书请求，应导出两个公钥。
本步骤中需调用的接口：C_GetAttributeValue。
- 3、由 CFCA 控件，组织生成证书请求信息。
- 4、调用 PKCS#11 接口，对 PKCS#10 请求进行 SM2 签名。
本步骤中需调用的接口：C_SignInit、C_Sign。
- 5、由 CFCA 控件完成证书申请，生成 PKCS#10 请求。

6.2 PKCS#11 接口描述

6.2.1 C_GenerateKeyPair

```
CK_RV C_GenerateKeyPair( CK_SESSION_HANDLE hSession,
                          CK_MECHANISM_PTR pMechanism,
                          CK_ATTRIBUTE_PTR pPublicKeyTemplate,
                          CK_ULONG ulPublicKeyAttributeCount,
                          CK_ATTRIBUTE_PTR pPrivateKeyTemplate,
                          CK_ULONG ulPrivateKeyAttributeCount,
                          CK_OBJECT_HANDLE_PTR phPublicKey,
                          CK_OBJECT_HANDLE_PTR phPrivateKey )
```

描述：产生密钥对。

特殊参数取值说明：pMechanism：机制类型为：CKM_SM2_KEY_PAIR_GEN。
pPublicKeyTemplate：创建 SM2 密钥对时，使用的公钥模板。
pPrivateKeyTemplate：创建 SM2 密钥对时，使用的私钥模板。

备注：

1、创建 SM2 密钥对时的公钥模板：

```
CKO_OBJECT_CLASS ulPubkeyClass = CKO_PUBLIC_KEY;
CK_BBOOL bTrue = TRUE;
CK_KEY_TYPE ulKeyType = CKK_SM2;
CK_UTF8CHAR sm2CurveName[] = "ChinaSM2Curve";
{CKA_CLASS, &ulPubkeyClass, sizeof(ulPubkeyClass)},
{CKA_TOKEN, &bTrue, sizeof(bTrue)},
{CKA_KEY_TYPE, &ulKeyType, sizeof(ulKeyType)},
{CKA_MODIFIABLE, &bTrue, sizeof(bTrue)},
{CKA_ID, byCkID, strlen((char *)byCkID)},
{CKA_LABEL, pszContainerName, strlen((char *)pszContainerName)},
{CKA_ENCRYPT, &bEncryDecry, sizeof(bEncryDecry)},
{CKA_VERIFY, &bVerifySign, sizeof(bVerifySign)},
{CKA_WRAP, &bTrue, sizeof(bTrue)},
{CKA_PRIVATE, &bFalse, sizeof(bFalse)},
{CKA_SM2_CURVE_NAME, sm2CurveName, sizeof(sm2CurveName)-1}
```

其中：

CKA_ID：取值为“容器名#1”或“容器名#2”，

例如：9125758C-60F3-45B7-8552-7BD1FDFA2B8F#1

CKA_LABEL：取值为容器名，即 UUID，例如：9125758C-60F3-45B7-8552-7BD1FDFA2B8F

CKA_ENCRYPT：设置密钥用途，需根据签名、加密公钥类型不同，设置为对应的值。

CKA_VERIFY: 设置密钥用途, 需根据签名、加密公钥类型不同, 设置为对应的值。

2、创建 SM2 密钥对时的私钥模板:

```
CKO_OBJECT_CLASS      ulPrikeyClass      = CKO_PRIVATE_KEY;
CK_BBOOL              bTrue              = TRUE;
CK_KEY_TYPE           ulKeyType          = CKK_SM2;
CK_UTF8CHAR           sm2CurveName[]    = "ChinaSM2Curve";
{CKA_CLASS,           &ulPrikeyClass,    sizeof(ulPrikeyClass)},
{CKA_TOKEN,           &bTrue,             sizeof(bTrue)},
{CKA_KEY_TYPE,        &ulKeyType,        sizeof(ulKeyType)},
{CKA_MODIFIABLE,      &bTrue,             sizeof(bTrue)},
{CKA_ID,              byCkID,            strlen((char *)byCkID)},
{CKA_LABEL,           pszContainerName,   strlen((char *)pszContainerName)},
{CKA_SENSITIVE,       &bTrue,             sizeof(bTrue)},
{CKA_SIGN,            &bSignVerify,      sizeof(bSignVerify)},
{CKA_DECRYPT,         &bEncryDecry,      sizeof(bEncryDecry)},
{CKA_UNWRAP,         &bTrue,             sizeof(bTrue)},
{CKA_PRIVATE,         &bTrue,             sizeof(bTrue)},
{CKA_SM2_CURVE_NAME,  sm2CurveName,      sizeof(sm2CurveName)-1}
```

其中:

CKA_ID: 取值为“容器名#1”或“容器名#2”,

例如: 9125758C-60F3-45B7-8552-7BD1FDFA2B8F#1

CKA_LABEL: 取值为容器名, 即 UUID, 例如: 9125758C-60F3-45B7-8552-7BD1FDFA2B8F

CKA_DECRYPT: 设置密钥用途, 需根据签名、加密公钥类型不同, 设置为对应的值。

CKA_SIGN: 设置密钥用途, 需根据签名、加密公钥类型不同, 设置为对应的值。

6.2.2 C_GetAttributeValue

```
CK_RV  C_GetAttributeValue( CK_SESSION_HANDLE  hSession,
                             CK_OBJECT_HANDLE   hObject,
                             CK_ATTRIBUTE_PTR    pTemplate,
                             CK_ULONG           ulCount )
```

描述: 获取对象属性。

特殊参数取值说明: hObject: 导出公钥时, 该参数为公钥句柄。

pTemplate: 获取参数的模板。

备注:

1、导出公钥的 X 分量时, 模板为: {CKA_SM2_X, NULL_PTR, 0}。

2、导出公钥的 Y 分量时, 模板为: {CKA_SM2_Y, NULL_PTR, 0}。

6.2.3 C_SignInit

CK_RV C_SignInit(CK_SESSION_HANDLE hSession,
CK_MECHANISM_PTR pMechanism,
CK_OBJECT_HANDLE hKey)

描述： 签名初始化。

特殊参数取值说明： pMechanism： 在对 PKCS#10 申请签名时，
其机制为 CKM_SM2_SIGN_VERIFY（带默认 Z 值）。

6.2.4 C_Sign

CK_RV C_Sign(CK_SESSION_HANDLE hSession,
CK_BYTE_PTR pData,
CK_ULONG ulDataLen,
CK_BYTE_PTR pSignature,
CK_ULONG_PTR pulSignatureLen)

描述： 签名。

特殊参数取值说明： pSignature： 为返回签名值结果，签名结果请参照章节 8.2。

7.SM2 数字证书导入调用 PKCS#11 接口规范

7.1 SM2 签名证书导入

SM2 签名证书导入时，仅需导入签名公钥证书，只涉及一个 PKCS#11 接口： C_CreateObject。

7.1.1 PKCS#11 接口描述

7.1.1.1 C_CreateObject

CK_RV C_CreateObject(CK_SESSION_HANDLE hSession,
CK_ATTRIBUTE_PTR pTemplate,
CK_ULONG ulCount,
CK_OBJECT_HANDLE_PTR phObject)

描述： 创建对象。

特殊参数取值说明： pTemplate： 对象模板。

备注：

1、 模板取值如下：

CK_OBJECT_CLASS	ulCertificateClass	= CKO_CERTIFICATE;
CK_BBOOL	bTrue	= TRUE;

CK_BBOOL	bFalse	= FALSE;
CK_CERTIFICATE_TYPE	ulCertificateType	= CKC_X_509;
{CKA_CLASS,	&ulCertificateClass,	sizeof(ulCertificateClass)},
{CKA_TOKEN,	&bTrue,	sizeof(bTrue)},
{CKA_LABEL,	pszContainerName,	strlen((char *)pszContainerName)},
{CKA_MODIFIABLE,	&bTrue,	sizeof(bTrue)},
{CKA_ID,	byCkID,	strlen((char *)byCkID)},
{CKA_CERTIFICATE_TYPE,	&ulCertificateType,	sizeof(ulCertificateType)},
{CKA_PRIVATE,	&bFalse,	sizeof(bFalse)},
{CKA_VALUE,	pbyCertificate,	ulCertificateSize}

其中：

CKA_LABEL：在导入 SM2 公钥证书时，需按章节 5 中的规则设定。

CKA_ID：在导入 SM2 公钥证书时，需按章节 5 中的规则设定。

7.2 SM2 加密证书及私钥导入流程

SM2 加密证书私钥以密文形式导入 KEY 中，流程如下：

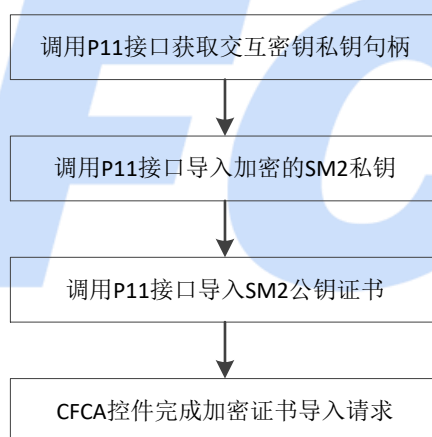


图 2 SM2 加密证书导入流程

SM2 加密证书导入过程中，需将公钥证书及对应的私钥同时导入。在本文档中，SM2 加密证书私钥必须以密文形式导入 KEY 中。导入流程图见图 2，详细说明如下：

- 1、调用 PKCS#11 接口，根据 CKA_CLASS 和 CKA_ID 查找并获取 SM2 交互密钥私钥句柄。

本步骤中需调用到的接口：C_FindObjectInit、C_FindObjects、C_FindObjectsFinal。

- 2、调用 PKCS#11 接口，导入加密后的 SM2 私钥。

本步骤中需调用到的接口：C_UnwrapKey。

- 3、调用 PKCS#11 接口，导入 SM2 加密证书。

本步骤中需调用到的接口：C_CreateObject。

- 4、由 CFCA 控件完成 SM2 加密证书导入请求。

7.2.1 PKCS#11 接口描述

7.2.1.1 C_FindObjectsInit

```
CK_RV C_FindObjectsInit( CK_SESSION_HANDLE    hSession,
                          CK_ATTRIBUTE_PTR      pTemplate,
                          CK_ULONG              ulCount )
```

描述： 查找对象初始化。

特殊参数取值说明： pTemplate： 查找私钥对象时，通过 CKA_CLASS、CKA_ID 来查找。

7.2.1.2 C_FindObjects

```
CK_RV C_FindObjects( CK_SESSION_HANDLE    hSession,
                     CK_OBJECT_HANDLE_PTR  phObject,
                     CK_ULONG              ulMaxObjectCount,
                     CK_ULONG_PTR          pulObjectCount )
```

描述： 查找对象。

特殊参数取值说明： 无。

7.2.1.3 C_FindObjectsFinal

```
CK_RV C_FindObjectsFinal( CK_SESSION_HANDLE hSession )
```

描述： 结束查找对象。

特殊参数取值说明： 无。

7.2.1.4 C_UnwrapKey

```
CK_RV C_UnwrapKey( CK_SESSION_HANDLE    hSession,
                   CK_MECHANISM_PTR      pMechanism,
                   CK_OBJECT_HANDLE       hUnwrappingKey,
                   CK_BYTE_PTR            pWrappedKey,
                   CK_ULONG               ulWrappedKeyLen,
                   CK_ATTRIBUTE_PTR       pTemplate,
                   CK_ULONG               ulAttributeCount,
                   CK_OBJECT_HANDLE_PTR   phKey )
```

描述： 通过 UnwrapKey 导入加密私钥。

特殊参数取值说明： pMechanism： CKM_SM2_CRYPT。

hUnwrappingKey： 交互私钥句柄。

pWrappedKey: 使用交互密钥公钥加密的加密证书私钥，
加密证书私钥结构请参照章节 8。

ulWrappedKeyLen: 加密后的私钥长度。

pTemplate: 导入模板。

备注:

- 1、SM2 私钥包含公钥值，创建私钥对象时须同时创建对应的公钥对象（请按照 CFCA 制定的 CKA_LABEL、CKA_ID 和 CKA_SM2_CURVE_NAME 规则来创建）。
- 2、创建对应的公钥、私钥对象时请先删除原有的交互密钥公钥、私钥对象（因交互密钥对与待导入的加密密钥对的 CKA_LABEL 与 CKA_ID 一样，会造成混淆）。
- 3、SM2 私钥以密文形式导入，导入私钥时，需指定 CKA_KEY_TYPE 为 CKO_PRIVATE_KEY，且置 CKA_TOKEN 属性为 TRUE，具体私钥值由 KEY 内部解密出后写入，导入模板如下：

```
CK_OBJECT_CLASS      ulPrikeyClass      = CKO_PRIVATE_KEY;
CK_KEY_TYPE          ulKeyType          = CKK_SM2;
CK_BBOOL             bTrue              = TRUE;
CK_BBOOL             bFalse             = FALSE;
CK_UTF8CHAR          sm2CurveName[]    = "ChinaSM2Curve";
{CKA_CLASS,          &ulPrikeyClass,    sizeof(ulPrikeyClass)},
{CKA_KEY_TYPE,       &ulKeyType,        sizeof(ulKeyType)},
{CKA_TOKEN,          &bTrue,            sizeof(bTrue)},
{CKA_LABEL,          pszContainerName,   strlen((char *)pszContainerName)},
{CKA_ID,             byCkID,            strlen((char *)byCkID)},
{CKA_MODIFIABLE,     &bTrue,            sizeof(bTrue)},
{CKA_SENSITIVE,      &bTrue,            sizeof(bTrue)},
{CKA_UNWRAP,         &bTrue,            sizeof(bTrue)},
{CKA_PRIVATE,        &bTrue,            sizeof(bTrue)},
{CKA_DECRYPT,         &bTrue,            sizeof(bTrue)},
{CKA_SIGN,           &bFalse,           sizeof(bFalse)},
{CKA_SM2_CURVE_NAME, sm2CurveName,      sizeof(sm2CurveName)-1}},
```

- 4、导入加密证书请参照章节 7.1.1.1。

8. SM2 私钥及签名结构

8.1 SM2 导入私钥结构

SM2 加密证书私钥需以加密形式导入，私钥结构为自定义的 ASN.1 格式（DER 编码）：

TAG 为：0x04；

内容为： OCTET STRING EncryptedPrivateKey

备注：

1、加密私钥密文 EncryptedPrivateKey 存在两种格式：

一种为 C1||C2||C3（国密老的标准），另一种为 C1||C3||C2（国密最新标准）。

P11 需要能够自动兼容上述两种私钥密文格式。

2、解密后的 SM2 密钥对为 x||y||d，其中 x，y 是 32 字节的公钥坐标点，d 是 32 字节的私钥。

8.2 SM2 签名结构

在做 PKCS#10 请求及签名接口时，需要 Key 做 SM2 签名并返回签名结果。SM2 签名结果为 ASN.1 格式（DER 编码）结构如下：

```
SM2Signature ::= SEQUENCE {
    R    INTEGER,
    S    INTEGER
}
```

其中：

1、R：取值为签名 r 值。

2、S：取值为签名 s 值。

9. PIN 码框

为保证用户数据安全，PIN 码输入框应由 P11 库弹出。

在 C_Login 时，CFCA 会向 P11 库传入默认的 PIN 码：

```
unsigned char byPin[] = {0x01, 0x08, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x00};
```

P11 库检测到此 PIN 码，则自动弹出 PIN 码框，要求用户输入 PIN 码。

10. 代码签名

为保证 P11 库文件不被恶意修改，厂商应使用自己的代码签名证书对库文件进行文件签名，CFCA 在调用 P11 库前会首先验证此签名。

厂商需提供以下几项内容：

1、P11 库文件名称及其安装路径；例如：/usr/lib/xxx.dylib

2、P11 库文件的代码签名文件；

文件内容：P11 库文件的 RSA PKCS#7 分离式签名结果（Base64 编码）。

命名方法：P11 库文件名.sig，例如：xxx.sig

安装路径：与 P11 库文件的安装路径相同。

3、验签 P11 库文件代码签名时，所需的证书链。