ICS 35.040
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0093—2020

证书与密钥交换格式规范

Certificate and key exchange format specification

2020-12-28 发布　　2021-07-01 实施

国家密码管理局　　发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：北京信安世纪科技股份有限公司、格尔软件股份有限公司、北京数字认证股份有限公司、长春吉大正元信息技术股份有限公司、兴唐通信科技有限公司、卫士通信息产业股份有限公司、国家信息安全工程技术研究中心、山东得安信息技术有限公司、北京创原天地科技有限公司、西安西电捷通无线网络通信股份有限公司。

本文件主要起草人：汪宗斌、刘婷、郑强、傅大鹏、赵丽丽、王妮娜、赵闪、罗俊、张旭、周淑静、张庆勇、焦靖伟、史晓峰、马洪富、杜志强。

引　言

本文件的内容参照个人信息交换语法(RFC7292 PKCS ＃12),按照我国相关密码政策和规范,结合我国实际应用需求及产品生产厂商的实践经验,定义了基于 SM2 密码算法的证书与密钥交换格式。

对于需要传递的证书与密钥等用户个人信息,涉及信息机密性和完整性保护方法。机密性保护使用加密技术来防止个人信息被暴露,完整性保护则防止个人信息被篡改。

本文件支持机密性保护方法和完整性保护方法的四种组合。

所述机密性保护,有以下两种方法。

——公钥机密性保护方法:在源平台上,使用已知可信的目标平台的加密公钥以数字信封的形式来封装用户个人信息。这个数字信封可以被对应的加密私钥打开。

——口令机密性保护方法:用从机密性口令派生的对称密钥加密用户个人信息。如果同时使用口令完整性保护方法,机密性保护口令和完整性保护口令可以相同也可以不相同。

所述完整性保护,有以下两种方法。

——公钥完整性保护方法:通过对 AuthenticatedSafe 内容的数字签名来保证完整性。在源平台上,使用签名私钥产生数字签名。在目标平台上,使用对应的签名公钥来验证签名。

——口令完整性保护方法:通过保密的完整性口令产生消息鉴别码(MAC)来保证完整性。如果口令方式的机密性保护方法被同时采用,机密性保护口令和完整性保护口令可以相同,也可以不同。

注意,这里讨论的密钥仅指用于传递用户个人信息的密钥。用户可能希望把个人密钥从一个平台传递到另外一个平台(可以保存在 PDU 中),但不要将这里讨论的用于传递用户个人信息的密钥与用户的个人密钥混淆。

本文件通过基于公钥的机密性和完整性保护方法提供高层次的安全防护,需要源平台和目标平台分别具有可用于数字签名和加密的可信密钥对;同时也支持略低的安全需求,基于口令的机密性和完整性保护方法,用于不能提供可信密钥对的环境。

证书与密钥交换格式规范

1 范围

本文件规定了证书与密钥等信息的传递语法，包括私钥、证书、证书撤销列表、各种形式的秘密值及其扩展的标准化封装。

本文件适用于个人的 SM2 算法证书与密钥等信息在不同平台之间迁移的应用场景。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 15852.2—2012 信息技术 安全技术 消息鉴别码 第 2 部分：采用专用杂凑函数的机制

GB/T 35275—2017 信息安全技术 SM2 密码算法加密签名消息语法规范

GB/T 35276—2017 信息安全技术 SM2 密码算法使用规范

GB/T 33560—2017 信息安全技术 密码应用标识规范

GM/T 0091—2020 基于口令的密钥派生规范

GM/Z 4001 密码术语

3 术语和定义

GM/Z 4001 和 GM/T 0091—2020 界定的术语和定义适用于本文件。

3.1

属性 attribute

一个 ASN.1 类型，标识一个属性类型（通过一个对象标识符）及其相关的属性值。

3.2

平台 platform

机器硬件、操作系统和应用软件组成的集合。

注：用户在这些环境中行使他的个人标识。在这里，应用是指使用用户个人信息的软件。如果机器硬件不同，或者软件不同，即为平台不同。在多用户系统中，每个用户至少有一个平台。

3.3

源平台 source platform

最终要传递到目标平台上的个人信息的起源平台。

3.4

目标平台 target platform

源平台上产生的个人信息要传递到的最终目的平台。

3.5

目的加密密钥对 destination encryption key pair

用于公钥机密性保护方法中的特定平台的密钥对。

注：公钥被称作 DesEncPubKey(当需要强调是可信的公钥时，称为 TDesEncPubKey)，私钥称为 DesEncPrivKey。

3.6

源签名密钥对　source signature key pair

用于公钥完整性保护方法中的特定平台的签名密钥对。

注：公钥称为 SrcSigPubKey(当需要强调是可信的公钥时，称为 TSrcSigPubKey)，私钥称为 SrcSigPrivKey。

3.7

包裹　shrouded

对私钥的加密，防止以明文方式保存的密钥被非法接口查看。

4　缩略语

下列缩略语适用于本文件。

ASN.1：抽象语法符号(Abstract Syntax Notation One，X.208)

CKX：证书与密钥交换格式(Certificate and Key Exchange Format)。本文件中为存储和传输用户或服务器私钥、公钥和证书指定的一个可移植的格式。它是一种二进制格式，这些文件也称为 CKX 文件。

DER：确定性编码规则(Distinguished Encoding Rules，X.690)

HMAC：散列消息鉴别码(Hash Message Authentication Code)

PDU：协议数据单元(Protocol Data Unit)。一个协议中，格式与机器硬件无关，组成一个消息的字节序列。

5　OID 定义

本文件对 6 个对象 keyBag、shroudedKeyBag、certBag、crlBag、secretBag、safeContentBag 和相关属性的标识符进行了定义，详见表 1。

表 1　对象标识符

对象标识符 OID	对象标识符定义
1.2.156.10197.6.1.4.1.12	证书与密钥交换格式规范 ckx
1.2.156.10197.6.1.4.1.12.10.1.1	密钥包 keyBag
1.2.156.10197.6.1.4.1.12.10.1.2	包裹的密钥包 shroudedKeyBag
1.2.156.10197.6.1.4.1.12.10.1.3	证书包 certBag
1.2.156.10197.6.1.4.1.12.10.1.4	证书废止列表包 crlBag
1.2.156.10197.6.1.4.1.12.10.1.5	秘密包 secretBag
1.2.156.10197.6.1.4.1.12.10.1.6	安全内容包 safeContentBag
1.2.156.10197.6.1.4.1.9.20	昵称(友好名称) friendlyName
1.2.156.10197.6.1.4.1.9.21	密钥标识 localKeyId
1.2.156.10197.6.1.4.1.9.22	证书类型 certTypes
1.2.156.10197.6.1.4.1.9.23	证书废止列表类型 crlTypes
1.2.156.10197.6.1.4.1.9.216	用户自定义扩展 ckxUser

6 基本类型定义

6.1 CKX 类型

CKX 是本文件中定义的最高层交换数据单元，通过对 AuthenticatedSafe 内容进行完整性保护，以实现将 CKX 数据单元从一个平台安全地传递到另一个平台。

AuthenticatedSafe 可以经过数字签名(公钥完整性保护方法)或者 MAC(口令完整性保护方法)产生一个 CKX 数据单元。

如果 AuthenticatedSafe 经过数字签名，则 CKX 由在源平台的签名私钥 SrcSigPrivKey 产生的数字签名和对应的可信签名公钥 TSrcSigPubKey 组成。TSrcSigPubKey 应和 CKX 一起传递到目标平台，这样用户才可以验证公钥是否可信并验证 AuthenticatedSafe 的签名。如果 AuthenticatedSafe 经过 MAC，则 CKX 由 AuthenticatedSafe 内容和一个从保密的完整性口令、盐值、迭代次数产生的消息鉴别码组成。

CKX 类型结构定义如下：(下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行)

```
CKX ::= SEQUENCE {
    Version INTEGER (1),
    authSafe ContentInfo,
    macData MacData OPTIONAL
}
```

authSafe 是交换内容，其类型为 ContentInfo。ContentInfo 结构定义按 GB/T 35275—2017 中的 6.10。现摘录如下：

```
ContentInfo ::=SEQUENCE{
    contentType ContentType,
    content [0]EXPLICIT ANY DEFINED BY contentType OPTIONAL
}
ContentType ::= OBJECT IDENTIFIER
```

ContentType 内容类型是一个对象标识符，其定义按 GB/T 35275—2017 的第 5 章。

MacData 类型定义如下：(下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行)

```
MacData ::= SEQUENCE {
    mac DigestInfo,
    macSalt OCTET STRING,
    iterations INTEGER DEFAULT 1024
}
DigestInfo ::= SEQUENCE {
    digestAlgorithm DigestAlgorithmIdentifier,
    digest Digest
}
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
Digest ::= OCTET STRING
```

DigestAlgorithmIdentifier 类型标识一个密码杂凑算法，本文件为有密钥使用的 SM3 密码杂凑算法，其 OID 按 GB/T 33560—2017 执行。

CKX 数据类型组成见表 2。

表 2　CKX 数据类型

字段名称	数据类型	含义
version	INTEGER	语法的版本号
authSafe	ContentInfo	交换内容。ContentInfo 结构定义，按 GB/T 35275—2017 的 6.10
macData	MacData	消息鉴别码

在公钥完整性保护方法下，authSafe 的 ContentInfo 域是 SignedData 类型，以 AuthenticatedSafe 作为被签名内容的 SignedData 类型的 BER 编码。SignedData 类型定义按 GB/T 35275—2017 的第 8 章。

在口令完整性保护方法下，authSafe 的 ContentInfo 域是 Data 类型，以 AuthenticatedSafe 作为 Data 内容的 BER 编码。Data 类型定义按 GB/T 35275—2017 的第 7 章。

macData 是可选项，在公钥完整性保护方法下省略，在口令完整性保护方法下可选。

在口令完整性保护方法下，authSafe 域的 content 内容的 BER 编码作为明文内容，通过 HMAC-SM3 算法生成消息鉴别码 MAC，HMAC 算法遵循 GB/T 15852.2—2012 的规定，其中密码杂凑算法推荐使用 SM3。MAC 的密钥由保密的完整性口令、盐值 macSalt 和迭代次数 iterations 派生得到，密钥派生函数按 GM/T 0091—2020 执行。口令应为 BMPString，且每个字符都以高位有效字节在前的方式以2 字节进行编码，最后附带 2 个字节 0x00。

6.2　AuthenticatedSafe 类型

authSafe 的 contentType 域是 Data 类型或者 SignedData 类型。authSafe 的 content 域直接（类型为 Data 时）或间接（类型为 SignedData 时）的包含一个 AuthenticateSafe 类型的 BER 编码。

AuthenticatedSafe 类型定义如下：

AuthenticatedSafe ::= SEQUENCE OF ContentInfo

ContentInfo 定义见 6.1。

一个 AuthenticatedSafe 包括一系列有序的 ContentInfo 实例。这些 ContentInfo 域的 content 域包含明文的、加密的或经过数字信封加密的数据。

如果是明文，则 ContentInfo 域是 Data 类型，Data 类型定义按 GB/T 35275—2017 的第 7 章。

如果是口令机密性保护方法，则 ContentInfo 域是 EncryptedData 类型，EncryptedData 类型定义按 GB/T 35275—2017 的第 11 章；

如果是公钥机密性保护方法，则 ContentInfo 域是 EnvelopedData 类型，EnvelopedData 类型定义按 GB/T 35275—2017 的第 9 章。

其中，EncryptedData 和 EnvelopedData 的保护明文是一个或多个 SafeContents 实例的 BER 编码。7.1 描述了 AuthenticatedSafe 类型各个域值的解释。

可以按照用户的要求，AuthenticatedSafe 的每个 ContentInfo 包含私钥、已加密的私钥、证书、证书撤销列表（CRLs）或其他需要保护的秘密信息等的任意集合，存储为 SafeContents 类型数据。

6.3　SafeContents 类型

SafeContents 类型由多个 SafeBag 构成。每个 SafeBag 包含密钥、证书或其他信息，由对象标识符区分。

类型定义如下：（下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行）

SafeContents ::= SEQUENCE OF SafeBag

```
SafeBag ::= SEQUENCE {
    bagId BAG-TYPE.&id ({CKXBagSet})
    bagValue      [0] EXPLICIT BAG-TYPE.&Type({CKXBagSet}{@bagId}),
    bagAttributes          SET OF CKXAttribute OPTIONAL
}
CKXAttribute ::= SEQUENCE {
    attrId ATTRIBUTE.&id ({CKXAttrSet}),
    attrValues SET OF ATTRIBUTE.&Type ({CKXAttrSet}{@attrId})
} -- 这个类型与 X.500 类型 Attribute 兼容

CKXAttrSet ATTRIBUTE ::= {
    friendlyName | localKeyId, -- 其他属性也允许
}
```

可选的 bagAttributes 域允许用户指定昵称和密钥标识等，并且允许通过可视化工具向用户显示某种有意义的字符串。

6.4 SafeBag 类型

6.4.1 概述

本文件定义了 SafeBag 的 6 种类型，分别对应着 KeyBag、ShroudedKeyBag、CertBag、CRLBag、SecretBag 和 SafeContents 类型。

类型定义如下：(下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行)

```
bagtypes OBJECT IDENTIFIER ::= {ckx 10 1}

BAG-TYPE ::= TYPE-IDENTIFIER
keyBag            BAG-TYPE ::= {KeyBag IDENTIFIED BY            {bagtypes 1}}
shroudedKeyBag    BAG-TYPE ::= {ShroudedKeyBag IDENTIFIED BY    {bagtypes 2}}
certBag           BAG-TYPE ::= {CertBag IDENTIFIED BY           {bagtypes 3}}
crlBag            BAG-TYPE ::= {CRLBag IDENTIFIED BY            {bagtypes 4}}
secretBag         BAG-TYPE ::= {SecretBag IDENTIFIED BY         {bagtypes 5}}
safeContentsBag   BAG-TYPE ::={SafeContents IDENTIFIED BY       {bagtypes 6}}

CKXBagSet BAG-TYPE ::= {
    keyBag | shroudedKeyBag | certBag | crlBag | secretBag | safeContentsBag,
    ... -- 预留将来扩展
}
```

6.4.2 KeyBag 类型

本文件的 KeyBag 是一个 SM2 私钥类型 ECPrivateKey。

类型定义如下：(下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行)

```
KeyBag ::= ECPrivateKey
```

ECPrivateKey 结构定义按 GB/T 35275—2017 的 13.3。现摘录如下：

```
ECPrivateKey{CURVES:IOSet}::= SEQUENCE{
```

```
    version INTEGER{ecPrivkeyVer1(1)}( ecPrivkeyVer1),
    privateKey SM2PrivateKey,
    parameters [0] Parameters{{IOSet}} OPTIONAL,
    publicKey [1] SM2PublicKey
}
```

注：一个 KeyBag 中只包含一个 SM2 私钥信息。

6.4.3 ShroudedKeyBag 类型

ShroudedKeyBag 是一个加密的 SM2 私钥结构。ShroudedKeyBag 类型定义如下：(下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行)

```
ShroudedKeyBag ::= SM2EnvelopedKey
```

SM2EnvelopedKey 结构定义按 GB/T 35276—2017 中的 7.4。现摘录如下：

```
SM2EnvelopedKey ::= SEQUENCE{
    symAlgID AlgorithmIdentifier,          --对称密码算法标识
    symEncryptedKey SM2Cipher,      --对称密钥密文
    Sm2PublicKey    SM2PublicKey,  --SM2 公钥
    Sm2EncryptedPrivateKey      BITSTRING     --SM2 私钥密文
}
```

其中，

——symAlgID 域为 SM4-CBC 的算法标识，其 OID 按 GB/T 33560—2017 的附录 A。

——symEncryptedKey 域为应用程序中的外部 SM2 公钥加密对称密钥得到的对称密钥密文，这里的外部 SM2 公钥是指用户的个人密钥，与平台密钥无关。请注意不要与用于传递用户个人信息的密钥，即目标平台 SM2 公钥(称为 TDesEncPubKey)混淆。SM2Cipher 类型定义按 GB/T 35276—2017 中的 7.2。

——Sm2PublicKey 域为被保护的 SM2 私钥对应公钥的 BER 编码，SM2PublicKey 类型定义按 GB/T 35276—2017 中的 7.1。

——Sm2EncryptedPrivateKey 域为 SM2 私钥密文的 BER 编码，SM2 私钥密文由对称密钥使用 AlgorithmIdentifier 域标识的算法加密得到。

注：ShroudedKeyBag 中只包含一个加密的 SM2 私钥信息。

6.4.4 CertBag 类型

CertBag 包含一个证书类型。对象标识用来区分不同类型的证书。

类型定义如下：(下面仅列出部分 ASN.1，完整的 ASN.1 定义按附录 A 执行)

```
CertBag ::= SEQUENCE {
    certId      BAG-TYPE.&id     ({CertTypes}),
    certValue [0] EXPLICIT BAG-TYPE.&Type ({CertTypes}{@certId})
}
x509Certificate BAG-TYPE ::={OCTET STRING IDENTIFIED BY {certTypes 1}}
            -- DER 编码的 X.509 证书用 OCTET STRING 存储
CertTypes BAG-TYPE ::= { x509Certificate, . -- 未来扩展}
```

6.4.5 CrlBag 类型

CRLBag 包含一个证书撤销列表(CRL)。使用对象标识来区分不同类型的 CRL。

类型定义如下:(下面仅列出部分 ASN.1,完整的 ASN.1 定义按附录 A 执行)

```
CRLBag ::= SEQUENCE {
    crlId       BAG-TYPE.&id ({CRLTypes}),
    crlValue    [0] EXPLICIT BAG-TYPE.&Type ({CRLTypes}{@crlId})
}
x509CRL BAG-TYPE ::={OCTET STRING IDENTIFIED BY {certTypes 1}
        -- DER 编码的 X.509 CRL 用 OCTET STRING 格式存储
CRLTypes BAG-TYPE ::= { x509CRL, -- 为了未来的扩展}
```

6.4.6 SecretBag 类型

SecretBag 表示其他类型的秘密信息,其值由对象标识符定义。

类型定义如下:(下面仅列出部分 ASN.1,完整的 ASN.1 定义按附录 A 执行)

```
SecretBag ::= SEQUENCE {
    secretTypeId      BAG-TYPE.&id ({SecretTypes}),
    secretValue   [0] EXPLICIT BAG-TYPE.&Type ({SecretTypes}{secretTypeId})
}
SecretTypes BAG-TYPE ::= { ... -- 留作扩展 }
```

注:一个 SecretBag 只能包含一个秘密信息。

6.4.7 SafeContents 类型

SafeBag 第六种类型是 SafeContents,允许以递归的方式实现多个 KeyBag、EncryptedPrivateKeyBag、CertBag、CRLBag 和 SecretBags 在上层 SafeContents 下的任意嵌套。

7 证书与密钥交换基本流程

7.1 创建 CKX 数据单元

创建 CKX 数据单元的步骤如下所述。

a) 根据 ASN.1 语法创建 SafeContents 的多个实例,每个实例包含 SafeBag 的多个(可能嵌套)实例。假设 SafeContents 的多个实例分别为 SC1、SC2、...、SCn。注意,在一个 CKX PDU 中可能有 SafeContents 的一个或多个实例。正如 b)中所述,每个实例可以被单独加密(或者不加密)。附录 B 给出了包含 2 个实例的示例供实现者参考。

b) 对每个 SCi ,根据所选择的不同加密选项。

   1) 如果 SCi 不被加密,建立一个 Data 类型的 ContentInfo 实例 CIi。Data 八位位组流内容是 SCi 的 BER 编码(包括标签、长度和值的八位位组流)。

   2) 如果 SCi 使用口令加密,建立一个 EncryptedData 类型的 ContentInfo 实例 CIi。设置 CIi 的 encryptedContentInfo 域的 contentType 域为 data 类型,并设置 encryptedContent 域是加密过的 SCi 的 BER 编码(注意,应包括标签和长度字节的八位位组流)。

   3) 如果 SCi 采用公钥加密,建立一个 EnvelopedData 类型的 ContentInfo 实例 CIi ,本质上与 b) 2)中的 EncryptedData ContentInfo 是同样的方式。

c) 在 SEQUENCE 中排列 CIi's,生成一个 AuthenticatedSafe 的实例。

d) 生成内容类型为 Data 的 ContentInfo 实例 T。Data 八位位组流的内容是 AuthenticatedSafe 值的 BER 编码(包括标签、长度和值的八位位组流)。

e) 对于完整性保护。

1) 如果 CKX PDU 使用数字签名进行认证，生成一个 SignedData 类型的 ContentInfo 实例 C。C 中 SignedData 的 contentInfo 域是 T。C 是顶层 CKX 结构中 ContentInfo 域的实例。

2) 如果 CKX PDU 使用 HMAC 进行认证，对 T 中 Data 内容使用 SM3 密码杂凑算法进行计算，得到 HMAC(也就是说，在八位位组流中排除标签和长度字节)。如果使用公钥认证，这也是 e) 1)中最初要计算杂凑值的内容。

7.2 从一个 CKX 数据单元中导入密钥和证书等

从 CKX 中导入与创建一个 CKX 的过程相反。大体来讲，当一个应用从 CKX 导入密钥等，应忽略所有无关的对象标识。有时候，需要提醒用户提供某些对象标识。

8 扩展属性

本文件提供了证书与密钥的交换格式。当这些信息被存储在一个目录服务中时，应使用 userCKX 属性。

类型定义如下：

```
userCKX ATTRIBUTE ::= {
        WITH SYNTAX CKX
        ID ckx-at-userCKX
    }
```

附 录 A
（规范性）
ASN.1 语法标记

本附录给出了本文件中定义的所有的 ASN.1 类型、值和对象。

```
-- 本模块已经通过 OSS ASN.1 工具进行的一致性检验
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- 全部导出
-- 导出这个模块中定义的所有类型和值都到其他 ASN.1 模块中使用
IMPORTS
informationFramework
    FROM UsefulDefinitions {joint-iso-itu-t(2) ds(5) module(1) usefulDefinitions(0) 3}
ATTRIBUTE
    FROM InformationFramework informationFramework
ContentInfo, DigestInfo

-- Object identifiers
cstc
OBJECT IDENTIFIER ::= {iso(1) member-body(2) cn(156) cstc(10197)}
ckx             OBJECT IDENTIFIER ::= { cstc ss(6) bc(1) sm(4) 12}
ckxPbeIds               OBJECT IDENTIFIER ::= {ckx 1}
pbeWithSM3ANDSM4_CBC                    OBJECT IDENTIFIER ::= {ckxPbeIds 8}
bagtypes        OBJECT IDENTIFIER ::= {ckx 10 1}
ckx-at-userCKX          OBJECT IDENTIFIER ::= {cstc ss(6) bc(1) sm(4) 1 9 216}
-- The CKX PDU
CKX ::= SEQUENCE {
        version                 INTEGER(1),
        authSafe                ContentInfo,
        macData         MacData OPTIONAL
}
MacData ::= SEQUENCE {
        mac             DigestInfo,
        macSalt         OCTET STRING,
        iterations      INTEGER DEFAULT 1024
}
AuthenticatedSafe ::= SEQUENCE OF ContentInfo
    -- Data if unencrypted
    -- EncryptedData if password-encrypted
    -- EnvelopedData if public key-encrypted
SafeContents ::= SEQUENCE OF SafeBag
SafeBag ::= SEQUENCE {
```

```
        bagId           BAG-TYPE.&id ({CKXBagSet }),
        bagValue        [0] EXPLICIT BAG-TYPE.&Type({CKXBagSet }{@bagId}),
    bagAttributes       SET OF CKXAttribute OPTIONAL
}

-- Bag types
keyBag          BAG-TYPE ::={KeyBag             IDENTIFIED BY {bagtypes 1}}
shroudedKeyBag  BAG-TYPE ::={ShroudedKeyBag     IDENTIFIED BY {bagtypes 2}}
certBag         BAG-TYPE ::={CertBag            IDENTIFIED BY {bagtypes 3}}
crlBag          BAG-TYPE ::={CRLBag             IDENTIFIED BY {bagtypes 4}}
secretBag       BAG-TYPE ::={SecretBag          IDENTIFIED BY {bagtypes 5}}
safeContentsBag BAG-TYPE ::={SafeContents       IDENTIFIED BY {bagtypes 6}}
CKXBagSet BAG-TYPE ::= {
    keyBag              |
    shroudedKeyBag      |
    certBag             |
    crlBag              |
    secretBag           |
    safeContentsBag,
    ... -- For future extensions
}
BAG-TYPE ::= TYPE-IDENTIFIER

-- KeyBag
KeyBag ::= ECPrivateKey

-- Shrouded KeyBag
ShroudedKeyBag ::= SM2EnvelopedKey

-- CertBag
CertBag ::= SEQUENCE {
    certId              BAG-TYPE.&id ({CertTypes}),
    certValue           [0]EXPLICIT BAG-TYPE.&Type ({CertTypes}{@certId})
}
x509Certificate BAG-TYPE ::=      {OCTET STRING IDENTIFIED BY {certTypes 1}}
    -- DER-encoded X.509 certificate stored in OCTET STRING
CertTypes BAG-TYPE ::= {   x509Certificate
    ... -- For future extensions
}

-- CRLBag
CRLBag ::= SEQUENCE {
    crlId           BAG-TYPE.&id ({CRLTypes}),
```

```
    crltValue       [0]EXPLICIT BAG-TYPE.&Type ({CRLTypes}{@crlId})
}
x509CRL BAG-TYPE ::=      {OCTET STRING IDENTIFIED BY {certTypes 1}}
    -- DER-encoded X.509 CRL stored in OCTET STRING
CRLTypes BAG-TYPE ::= {   x509CRL,
    ... -- For future extensions
}

-- Secret Bag
SecretBag ::= SEQUENCE {
    secretTypeId            BAG-TYPE.&id ({SecretTypes}),
    secretValue             [0]EXPLICIT BAG-TYPE.&Type ({SecretTypes}{@secretTypeId})
}
SecretTypes BAG-TYPE ::= {
    ... -- For future extensions
}

-- Attributes
CKXAttribute ::= SEQUENCE {
    attrId          ATTRIBUTE.&id ({CKXAttrSet}),
    attrValues      SET OF ATTRIBUTE.&Type ({CKXAttrSet}{@attrId})
} -- This type is compatible with the X.500 type 'Attribute'
CKXAttrSet ATTRIBUTE ::= {
    friendlyName    |
    localKeyId,
    ... -- Other attributes are allowed
}
userCKX ATTRIBUTE ::= {
            WITH SYNTAX CKX
            ID      ckx-at-userCKX
    }
END
```

附 录 B
（资料性）
双证书及私钥导入导出示例

B.1 需求说明

为了将用户个人信息如双证书（签名证书和加密证书）及对应的私钥从源平台传递到目标平台，满足双证书及私钥同步的迁移需求，本文件可提供类似智能密码钥匙密钥容器的功能，密钥容器示意图见图 B.1。

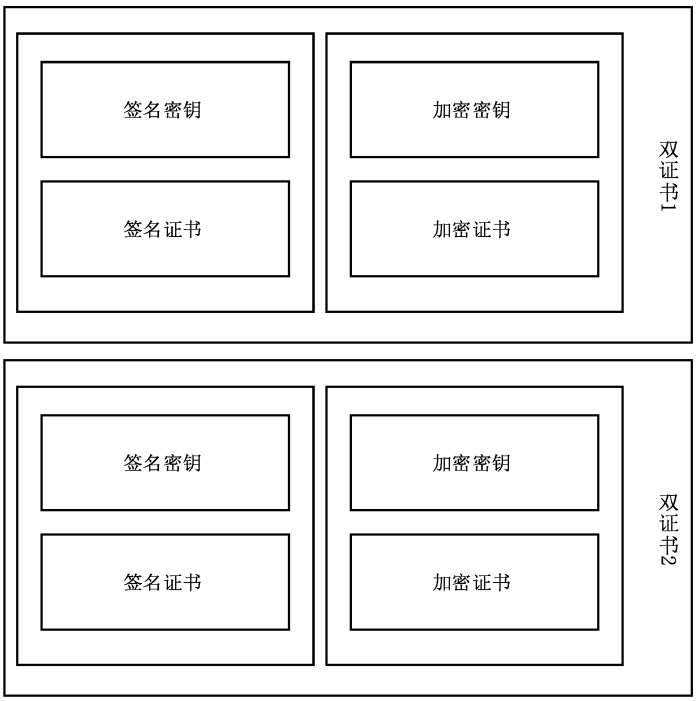

图 B.1 智能密码钥匙的容器示意图

本附录将详细列举从双证书及私钥导出 CKX PDU 以及从 CKX PDU 导入双证书及私钥的过程。以实际应用场景中，常用的口令机密性和口令完整性保护方法为例，说明本文件的用法以及 CKX 语法格式。

一般来说，口令方式的机密性保护方法和完整性保护方法若被同时采用，机密性保护口令和完整性保护口令可以相同，也可以不相同。本附录中机密性保护和完整性保护假设设置相同的口令。

B.2 说明了从双证书及私钥导出 CKX PDU 的过程，B.3 说明了从 CKX PDU 导入双证书及私钥的过程，B.4 从外层到内层阐述了 CKX PDU 的语法格式。

B.2 从双证书及私钥导出 CKX PDU

以创建双证书及对应私钥的 CKX PDU 为例，从内层到外层，步骤如下。

a) ASN.1 中建立 SafeContents 的两个实例 SC_sign、SC_enc。每个实例都包含 SafeBag 类型的两个实例。SC_sign 的两个 SafeBag 类型实例设为 SB_sign_cert、SB_sign_ shroudedKey，其中，SB_sign_cert 是 CertBag 类型，SB_sign_ shroudedKey 是 ShroudedKeyBag 类型。SC_enc 的两个实例设为 SB_enc_cert、SB_enc_shroudedKey，其中 SB_enc_cert 是 CertBag 类型，SB_

enc_shroudedKey 是 ShroudedKeyBag 类型。注意，本示例中的 CKX PDU 包含 SafeContents 的两个实例 SC_sign、SC_enc。

b) 对于 SC_sign 或 SC_enc，分别根据选择的不同加密选项。

1) SC_sign 使用口令方式加密，建立一个 EncryptedData 类型的 ContentInfo 实例 CI_sign。CI_sign 的 encryptedContentInfo 域的 contentType 为 Data 类型，并设置 encryptedContent 域是对加密过的 SC_sign 的 BER 编码(注意，应包括标签和长度字节的八位位组流)。其中，SC_sign 是 SB_sign_cert 和 SB_sign_ shroudedKey 的有序序列的 BER 编码。

2) SC_enc 使用口令方式加密，建立一个 EncryptedData 类型的 ContentInfo 实例 CI_enc。CI_enc 的 encryptedContentInfo 域的 contentType 为 Data 类型，并设置 encryptedContent 域是对加密过的 SC_enc 的 BER 编码(注意，应包括标签和长度字节的八位位组流)。其中，SC_enc 是 SB_enc_cert 和 SB_enc_ shroudedKey 的有序序列的 BER 编码。

c) 在 SEQUENCE 中排列 CI_sign 和 CI_enc，生成一个 AuthenticatedSafe 的实例。

d) 生成内容类型为 Data 的 ContentInfo 实例 T。Data 八位位组流的内容是 AuthenticatedSafe 值的 BER 编码(包括标签、长度和值的八位位组流)。

e) 对于完整性保护，CKX PDU 通过 HMAC 认证，对 T 中的 Data 内容使用 SM3 算法进行计算，得到 HMAC(也就是，在八位位组流中排除标签和长度字节)。

B.3 从 CKX PDU 导入双证书及私钥

从 CKX 中导入双证书和私钥，基本上和建立一个 CKX 的过程相反。从外层到内层，其步骤如下。

a) 对于完整性保护，从 CKX PDU 中取出 ContentInfo 实例 T 和 MacData 实例，对 T 中的 Data 内容(在八位位组流中排除标签和长度字节)使用 SM3 算法计算 HMAC；将 HMAC 与 MacData 实例的 digest 内容进行比对。若结果一致，进行 b)；若结果不一致，终止操作。

b) 取出 T 中的 Data 内容(在八位位组流中排除标签和长度字节)，即得到 AuthenticatedSafe 实例。

c) AuthenticatedSafe 实例是由两个 ContentInfo 实例组成的有序集合，将有序集合拆开成两个部分，并分别在八位位组流中排除标签和长度字节，得到 ContentInfo 实例 CI_sign 和 CI_enc。

1) 对 EncryptedData 类型的 ContentInfo 实例 CI_sign 的密文内容进行解密，得到 SafeContents 实例 SC_sign(排除标签和长度字节的八位位组流)。SC_sign 是两个 SafeBag 实例 SB_sign_cert 和 SB_sign_ shroudedKey 有序序列的 BER 编码。

i) SafeBag 实例 SB_sign_cert 根据 bagID 域得到是一个 CertBag 类型，根据 bagValue 和 bagAttributes 域导出签名证书。

ii) SafeBag 实例 SB_sign_ shroudedKey 根据 bagID 域得到是一个 shroudedKey 类型，根据 bagValue 和 bagAttributes 域导出签名证书对应的私钥。

2) 对 EncryptedData 类型的 ContentInfo 实例 CI_enc 的密文内容进行解密，得到 SafeContents 实例 SC_enc(排除标签和长度字节的八位位组流)。SC_enc 是 SB_enc_cert 和 SB_enc_ shroudedKey 的有序序列的 BER 编码。

i) SafeBag 实例 SB_enc_cert 根据 bagID 域得到是一个 CertBag 类型，根据 bagValue 和 bagAttributes 域导出加密证书。

ii) SafeBag 实例 SB_enc_ shroudedKey 根据 bagID 域得到是一个 shroudedKey 类型，根据 bagValue 和 bagAttributes 域导出加密证书对应的私钥。

B.4 双证书及私钥的 CKX PDU 语法

B.4.1 CKX 语法

以下从外层到内层，对双证书及私钥的 CKX PDU 的具体语法进行示例。

```
CKX ::= SEQUENCE {
    version INTEGER{v1(1)},
    authSafe ContentInfo,
    macData MacData OPTIONAL
}
```

ContentInfo 实例 T 的语法结构如下：

```
    ContentInfo::=SEQUENCE{
    contentType ContentType,--Data 类型 OID:1.2.156.10197.6.1.4.2.1
    content [0]EXPLICIT ANY DEFINED BY contentType OPTIONAL
    -- AuthenticatedSafe 实例的 BER 编码(包括标签、长度和值的八位位组流)
}
ContentType ::=OBJECT IDENTIFIER

MacData ::= SEQUENCE {
    mac      DigestInfo,
    macSalt      OCTET STRING,
    iterations      INTEGER DEFAULT 1024
}
DigestInfo ::= SEQUENCE {
    digestAlgorithm DigestAlgorithmIdentifier,
    --本文件为有密钥的 SM3 密码杂凑算法，其 OID 按 GB/T 33560—2017 执行。
    Digest Digest
}
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
Digest ::= OCTET STRING
```

B.4.2 AuthenticatedSafe 语法

AuthenticatedSafe 实例由两个 ContentInfo 实例 CI_sign 和 CI_enc 的有序集合构成，其语法结构如下：

```
AuthenticatedSafe ::= SEQUENCE OF ContentInfo
```

以下是 EncryptedData 类型的 ContentInfo 实例 CI_sign 的语法结构：

```
ContentInfo::=SEQUENCE{
    contentType      ContentType,--EncryptedData 类型的 OID 为 1.2.156.10197.6.1.4.2.5
    content   [0]EXPLICIT ANY DEFINED BY contentType OPTIONAL
}
```

其中，contentType 域是 EncryptedData 类型，content 域是对 SafeContents 实例 SC_sign 加密封装成 EncryptedData 类型的 BER 编码(包括标签、长度和值的八位位组流)。

B.4.3 SafeContents 语法

B.4.3.1 SC_sign 语法

SafeContents 实例 SC_sign 的语法结构如下：

```
SafeContents ::= SEQUENCE OF SafeBag
```

SC_sign 由两个 SafeBag 实例 SB_sign_cert 和 SB_sign_shroudedKey 有序序列构成。

以下是 SafeBag 实例 SB_sign_cert，是一个 CertBag 类型：

```
SafeBag ::= SEQUENCE {
    bagId BAG-TYPE.&id ({CKXBagSet}),
    -- CertBag 类型 OID 为 1.2.156.10197.6.1.4.1.12.3
    bagValue      [0] EXPLICIT BAG-TYPE.&Type({CKXBagSet}{@bagId}),
    --内容是 CertBag 实例 SM2 签名证书
    bagAttributes          SET OF CKXAttribute OPTIONAL
}
CertBag ::= SEQUENCE {
    certId      BAG-TYPE.&id ({CertTypes}),
    certValue   [0] EXPLICIT BAG-TYPE.&Type ({CertTypes}{@certId})
}
x509Certificate BAG-TYPE ::={OCTET STRING IDENTIFIED BY {certTypes 1}}
    -- DER 编码的 X.509 证书用 OCTET STRING 存储
```

以下是 SafeBag 实例 SB_sign_shroudedKey，是一个 ShroudedKeyBag 类型：

```
SafeBag ::= SEQUENCE {
    bagId BAG-TYPE.&id ({CKXBagSet}),
    --ShroudedKeyBag 类型 OID 为 1.2.156.10197.6.1.4.1.12.2
    bagValue      [0] EXPLICIT BAG-TYPE.&Type({CKXBagSet}{@bagId}),
    --内容是 ShroudedKeyBag 类型的数字信封保护的 SM2 签名证书对应的私钥
    bagAttributes          SET OF CKXAttribute OPTIONAL
}
ShroudedKeyBag ::= SM2EnvelopedKey
SM2EnvelopedKey ::= SEQUENCE{
    symAlgID AlgorithmIdentifier,          --对称密码算法标识
    symEncryptedKey SM2Cipher,         --对称密钥密文
    sm2PublicKey SM2PublicKey,         --SM2 公钥
    sm2EncryptedPrivateKey BIT STRING      --SM2 私钥密文
}
```

B.4.3.2 SC_enc 语法

SafeContents 实例 SC_enc 的语法结构如下：

```
SafeContents ::= SEQUENCE OF SafeBag
```

SC_enc 由两个 SafeBag 实例 SB_enc_cert 和 SB_enc_shroudedKey 有序序列构成。

以下是 SafeBag 实例 SB_enc_cert，是一个 CertBag 类型：

```
SafeBag ::= SEQUENCE {
```

```
        bagId BAG-TYPE.&id ({CKXBagSet}),
        -- CertBag 类型 OID 为 1.2.156.10197.6.1.4.1.12.3
        bagValue      [0] EXPLICIT BAG-TYPE.&Type({CKXBagSet}{@bagId}),
        --内容是 CertBag 实例 SM2 加密证书
        bagAttributes         SET OF CKXAttribute OPTIONAL
    }
    CertBag ::= SEQUENCE {
        certId      BAG-TYPE.&id ({CertTypes}),
        certValue [0] EXPLICIT BAG-TYPE.&Type ({CertTypes}{@certId})
    }
    x509Certificate BAG-TYPE ::={OCTET STRING IDENTIFIED BY {certTypes 1}}
        -- DER 编码的 X.509 证书用 OCTET STRING 存储
    以下是 SafeBag 实例 SB_sign_shroudedKey,是一个 ShroudedKeyBag 类型:
    SafeBag ::= SEQUENCE {
        bagId BAG-TYPE.&id ({CKXBagSet}),
        --ShroudedKeyBag 类型 OID 为 1.2.156.10197.6.1.4.1.12.2
        bagValue      [0] EXPLICIT BAG-TYPE.&Type({CKXBagSet}{@bagId}),
        --内容是 ShroudedKeyBag 类型的数字信封保护的 SM2 加密证书对应的私钥
        bagAttributes         SET OF CKXAttribute OPTIONAL
    }
    ShroudedKeyBag ::= SM2EnvelopedKey
    SM2EnvelopedKey ::= SEQUENCE{
        symAlgID AlgorithmIdentifier,          --对称密码算法标识
        symEncryptedKey SM2Cipher,        --对称密钥密文
        Sm2PublicKey SM2PublicKey,        --SM2 公钥
        Sm2EncryptedPrivateKey BIT STRING      --SM2 私钥密文
    }
```

参 考 文 献

[1] RFC7292 PKCS #12：Personal Information Exchange Syntax Version 1.0

中华人民共和国密码
行业标准
证书与密钥交换格式规范
GM/T 0093—2020
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室:(010)68533533 发行中心:(010)51780238
读者服务部:(010)68523946
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 1.5 字数 41 千字
2021年4月第一版 2021年4月第一次印刷
*
书号:155066·2-35944 定价 26.00 元

GM/T 0093-2020

码上扫一扫 正版服务到