ICS 35.040
L 80
备案号:

中 华 人 民 共 和 国 国 家 标 准

GM/T 0057-2018

基于 IBC 技术的身份鉴别规范

Identity authentication specifications based on IBC technology

（报批稿）

2018-05-02 发布　　2018-05-02 实施

国家密码管理局　发布

目　　次

前　言

本标准依据 GB/T 1.1-2009 给出的规则起草。

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这些专利的责任。

本标准由密码行业标准化技术委员会提出并归口。

本标准起草单位：上海信息安全工程技术研究中心、北京国脉信安科技有限公司、西安工业大学、无锡江南信息安全工程技术研究中心、中油瑞飞信息技术有限责任公司。

本标准起草人：袁峰、药乐、容晓峰、杜志强、王一曲、蒋楠、王建、崔广印、金一、万进。

本标准凡涉及密码算法的相关内容，按国家有关法规实施；凡涉及到采用密码技术解决保密性、完整性、真实性、不可否认性需求的须遵循密码相关国家标准和行业标准。

引　言

本标准是IBC（Identity-Based Cryptography）基于标识的密码技术系列标准之一，本标准依托于GM/T 0044 SM9标识密码算法标准，面向应用系统中基于IBC技术和SM9算法进行的身份鉴别时涉及到的鉴别需求。

鉴于标识密码技术的特点，规定了两种单向身份鉴别要求和一个双向身份鉴别要求。本标准还在附录中给出了利用IBC技术进行鉴别时需要访问公开参数服务（PPS）的基本流程和相关密码数据结构，用于公开参数和标识状态查询。

基于 IBC 技术的身份鉴别规范

1 范围

本标准规定了使用基于标识的密码技术的身份鉴别要求。

本标准适用于使用基于标识的密码技术的身份鉴别领域。

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，仅所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（包括所有的修改单）适用于本文件。

GM/T 0044 SM9标识密码算法

3 术语和定义

下列术语和定义适用于本文件。

3.1

标识 identity

可确定一个对象身份的唯一信息，例如电子邮箱地址、手机号码、指纹数据等。

3.2

SM9密码算法 SM9 algorithm

由 GM/T 0044 定义的一种算法。

3.3

公开参数服务 public parameter service

用于发布基于标识的密码技术中公开参数、私钥生成策略、用户标识信息和状态等数据的应用服务。

4 符号和缩略语

下列缩略语适用于本文件。

ASN.1：抽象语法标记（Abstract Syntax Notation One）

IBC： 基于标识的密码技术（Identity-Based Cryptography）

IRI： 国际化资源标识符（Internationalized Resource Identifiers）

OID： 对象标识符（Object identifier）

PKG： 私钥生成（Private Key Generation）

PPS： 公开参数服务（Public Parameter Service）

URI： 统一资源标识符（Uniform Resource Identifier）

5 标识结构

标识的ASN.1数据格式定义为：标识数据格式的ASN.1定义为：

```
Identifier::= SEQUENCE{
        ibcType              OBJECT IDENTIFIER,
        identityData         OCTET STRING,
        validStart           UTCTIME,
        validEnd        [0] UTCTIME OPTIONAL,
        idExtensions    [1] Extensions OPTION
}
```

其中：

identityType 身份类别是一个对象标识符号 OID，定义所应用算法或数据域的编码。

identityData 身份数据是八位串，身份的具体描述。

validStart 有效期的起始日期。

validEnd 有效期的终止日期，可选项，如果该项不存在表示标识长期有效。

idExtensions 扩展项是 Extensions 类型，标识信息扩展项。

```
Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension
Extension  ::=  SEQUENCE {
     extnID      OBJECT IDENTIFIER,
     critical    BOOLEAN DEFAULT FALSE,
     extnValue   OCTET STRING
}
```

其中：

extnID 表示一个扩展元素的 OID。

critical 表示这个扩展元素是否极重要。

extnValue 表示这个扩展元素的值，字符串类型。

6 用户身份鉴别规范

6.1 描述

用户身份鉴别过程包含了鉴别双方的流程和具体协议，在进行身份鉴别时各方都可以通过访问公开参数服务PPS获取相关鉴别体系的关键参数，例如IBC的公开参数；还可以通过访问PPS获取对方标识的相关信息，例如标识信息的状态，完整的标识数据等，如图1所示。

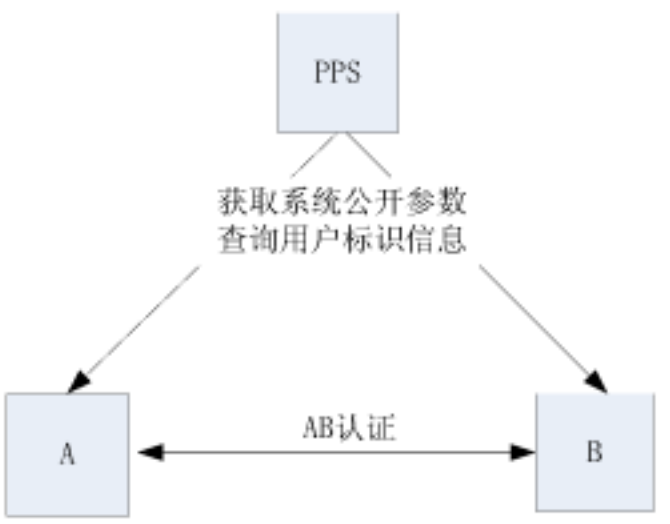

图 1 身份鉴别结构

对PPS的查询协议见附录A.4。

6.2 单向用户身份鉴别

6.2.1 接收者鉴别发起者身份

a) 一次传递鉴别

在接收者环境基本可信的情况下适用于一次传递鉴别。其特点是接收者B直接验证发起者A的身份，见图2。

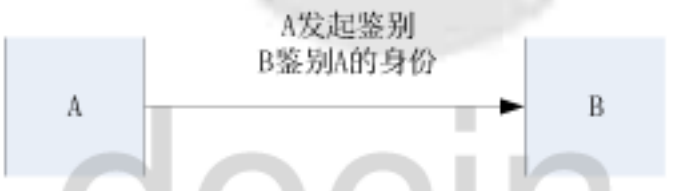

图 2 一次传递鉴别关系

发起者A发起鉴别过程，由接收者B对其身份进行鉴别。提交的用于验证的数据记作令牌Token，其唯一性/时效性通过产生并验证时间戳或者随机数进行控制。

具体内容如下：

A：A向B发送TokenAB,

A to B：$ID_A \| TokenAB$, 也可表示为{ ID_A, TokenAB }。

发起者A发送给验证者B的令牌记作TokenAB。

$TokenAB= ID_B \| r_A \| Text1 \| sign_A(ID_B \| r_A \| Text1)$,也可表示为，

$TokenAB=\{ID_B, r_A, Text1, sign_A(ID_B, r_A, Text1)\}$ 。

其中：

ID_B为验证者B的唯一性标识。

r_A为表示唯一性的时间或者由发起者产生的随机数。

$sign_A$为发起者A的签名，格式为SM9Signature。

Text1为可选项，其他需要传递的信息。如果有信息需要加密传输，则应以数字信封方式进行封装。

请求被鉴别的格式如下：

```
RequestAuthenticated::=SEQUENCE{
        userIDA         Identifier,
        authToken       AuthToken
}
AuthToken::=SEQUENCE{
        userIDB         Identifier,
        verifier        Verifier,
        text1           [0] OCTET STRING Option,
        userSignature   SM9Signature
}
Verifier::= CHOICE{
        generalTime         GeneralizedTime,
        serialNumber        INTEGER
}
```

userSignature 令牌签名SM9Signature类型，参加附录B。

B：验证A的鉴别信息

B接收A发来的含有TokenAB的消息后，进行身份鉴别，见图2。

具体内容如下：

1）B根据A的标识生成A的公钥，如果B没有生成A公钥的公共参数，应从公开参数服务系统PPS中获取；

2）检验包含在令牌中的A的签名信息

- 检验 TokenAB 中 ID_B标识段的值，是否等于实体 B 的标识符；
- 验证数字签名的正确性；
- 检验唯一性/时效性；
- 解析 Text1。

若任一验证结果不正确，则停止通联，且反馈：鉴别失败信息。

若验证正确，则完成身份鉴别。

b)　两次传递鉴别

在接收者环境基本可信的情况下适用于两次传递鉴别。当采用两次传递鉴别方式时r^A代表一个随机数，具体内容如图3：

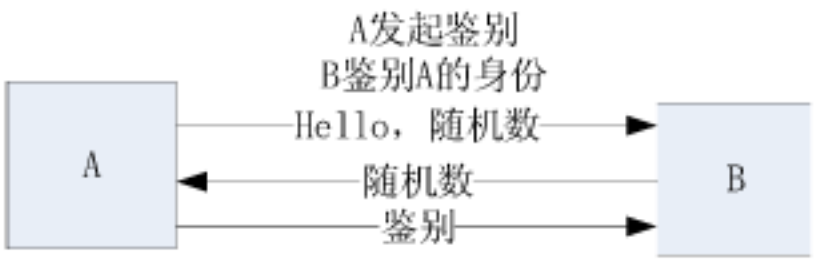

图 3 两次传递鉴别关系

A：A向B发送请求，附带随机数r^A。

A发送给B：$ID_A \| r_A$

请求被鉴别的格式如下：

```
RequestHello::=SEQUENCE{
        userIDA          Identifier,
        randeom          INTGER
}
```

B：回复A随机数r_B。

B发送给A：r_B

请求被鉴别的格式如下：

```
ResponseHello::=SEQUENCE{
        randeomB              INTGER
}
```

A：A向B发送TokenAB。

A发送给B：$ID_A \| TokenAB$，也可表示为$\{ID_A, TokenAB\}$

$TokenAB = ID_B \| Text1 \| sign_A(ID_B \| r \| Text1)$

$r=r_A \| r_B$。

Text1为需要传送的数据，可选项。

请求被鉴别的格式如下：

```
RequestAuthenticated::=SEQUENCE{
        userIDA          Identifier,
        authToken        AuthToken
}
```

B：验证A的鉴别信息

B接收A发来的含有TokenAB的消息后，进行身份鉴别。

6.2.2 发起者鉴别接收者身份

在接收者环境基本可信的情况下适用于单向鉴别。其特点是发起者A要求接收者B向A证明自己的身份，要求获取B的签名并验证，见图4。

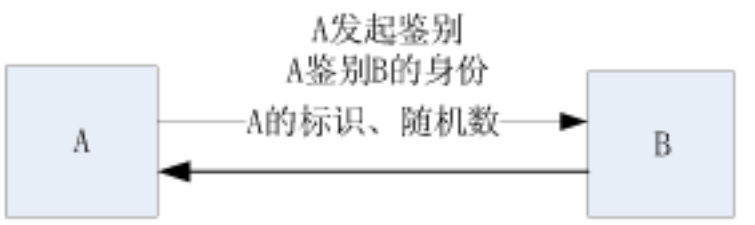

图 4 单向鉴别关系

发起者A发起鉴别过程，接收者B回应A的要求并将自身的身份信息返回给A，由A对其身份进行鉴别。提交的用于验证的数据记作令牌Token，其唯一性/时效性通过产生并验证时间戳或者随机数进行控制。

具体内容如下：

A：A向B发送获取验证B身份信息的要求。

A发送给B：$ID_A \| r_A \| Text1$，也可表示为$\{ID_A, r_A, Text1\}$。

其中：

ID_A为发起者A的唯一性标识。

r_A为表示唯一性的时间或者由发起者产生的随机数。

Text1为可选项，其他需要传递的信息。

要求鉴别对方的格式如下：

```
RequireAuthenticated::=SEQUENCE{
      userIDA          Identifier,
      verifier         Verifier,
      text1            OCTET STRING Option
}
```

B：B向A返回确认其身份的信息。

B发送给A：$ID_B \| TokenBA$，也可表示为$\{ID_B, TokenBA\}$。

接收者B发送给发起者A的令牌记作TokenBA。

$TokenBA= ID_A \| r \| Text2 \| sign_B(ID_A \| r \| Text2)$，也可表示为：

$TokenBA=\{ID_A, r, Text2, sign_B(ID_A, r, Text2)\}$ 。

其中：

ID_A为发起者A的唯一性标识。

ID_A为可选项，由于A对该值已知，B在向A返回时可以不再附带之。

$r=r_A \| r_B$。

r_B为B产生的随机数。

$Sign_B$为接收者B的签名。

Text2为可选项，其他需要传递的信息。如果有信息需要加密传输，则应以数字信封方式进行封装。

回应的格式ResponseAuthentication：

```
ResponseAuthenticated::=SEQUENCE{
      userIDB     Identifier,
      authToken        AuthToken
}
AuthToken::=SEQUENCE{
      userIDA          Identifier,
      verifier         Verifier,
      text2            OCTET STRING Option,
      userSignature    SM9Signature
}
Verifier::= CHOICE{
      generalTime      GeneralizedTime,
      serialNumber          INTEGER
}
```

A：验证B的鉴别信息。

发起者A接收B发来的含有TokenBA的消息后，进行身份鉴别。

具体内容如下：

a) A根据B的标识生成B的公钥，如果A没有生成B公钥的公共参数，应从公开参数服务系统PPS中获取；

b) 检验包含在令牌中的B的签名信息；

1) 检验 TokenBA 中 ID_A标识段的值，是否等于实体 A 的标识符，该步骤可选；

2) 验证数字签名的正确性；

3) 鉴别 r_A 是不是以前的那个；

4) 检验唯一性/时效性；

5) 解析 Text2。

若任一验证结果不正确，则停止通联，且反馈：鉴别失败信息。

若验证正确，则完成身份鉴别。

6.3 三次传递鉴别

其特点是接收者B验证发起者A的身份，接收者B也向A证明自己的身份，见图5。

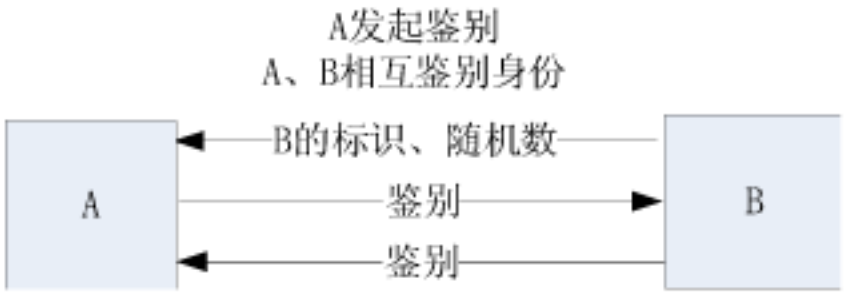

图 5 三次传递鉴别关系

A签名，将签名值发给B；

B直接验证；

B签名，将签名值发给A；

A直接验证。

发起者A发起鉴别过程，先由接收者B对其身份进行鉴别。接收者B再将自身的身份信息返回给A，由A对其身份进行鉴别。提交的用于验证的数据记作令牌Token，其唯一性/时效性通过产生并验证时间戳或者随机数进行控制。

具体内容如下：

B：B向A发送请求附带随机数r_B。

B发送给A：$ID_B \| r_B$，也可表示为$\{ID_B, r_B\}$。

其中：

ID_B为发起者B的唯一性标识。

r_B为表示唯一性的时间或者由发起者产生的随机数。

要求鉴别对方的格式如下：

```
RequireAuthenticated::=SEQUENCE{
        userIDB          Identifier,
        randeom          INTGER,
}
```

A：A向B发送TokenAB。

A发送给B：$ID_A \| TokenAB$，也可表示为$\{ ID_A, TokenAB \}$。

发起者A发送给验证者B的令牌记作TokenAB。

$TokenAB= ID_B \| r_A \| r_B \| Text1 \| sign_A(ID_B \| r_A \| r_B \| Text1)$，也可表示为，

$TokenAB=\{ID_B, r_A, r_B, Text1, sign_A(ID_B, r_A, r_B, Text1)\}$ 。

其中：

r_A为表示唯一性的时间或者由发起者产生的随机数、随机数。

$sign_A$为发起者A的签名。

ID_B为验证者B的唯一性标识。

Text1为可选项，其他需要传递的信息。如果有信息需要加密传输，则应以数字信封方式进行封装。

采用RequireAuthentication定义。

B：验证A的鉴别信息

B接收A发来的含有TokenAB的消息后，进行身份鉴别。

具体内容如下：

a) B根据A的标识生成A的公钥，如果B没有生成A公钥的公共参数，应从公开参数服务系统PPS中获取；

b) 检验包含在令牌中的信息；

1) 检验 TokenAB 中的 r_B是否与发送给之前发送给 A 的值相等；

2) 检验 TokenAB 中 IDB 标识段的值，是否等于实体 B 的标识符；

3) 验证数字签名的正确性；

4) 检验唯一性/时效性；

5) 解析 Text1。

若任一验证结果不正确，则停止通联，且反馈：鉴别失败信息。

若验证正确，则完成身份鉴别。

B：B向A返回确认其身份的信息。

B发送给A：$ID_B \| TokenBA$，也可表示为{ ID_B，TokenBA }。

接收者B发送给发起者A的令牌记作TokenBA。

$TokenBA= ID_A \| r_A \| Text2 \| sign_B(ID_A \| r_A \| Text2)$,也可表示为，

$TokenBA=\{ID_A, r_A, Text2, sign_B(ID_A, r_A, Text2)\}$ 。

其中：

ID_A为发起者A的唯一性标识。

r_A为表示唯一性的时间或者由发起者产生的随机数、随机数。

$Sign_B$为接收者B的签名。

Text2为可选项，其他需要传递的信息。如果有信息需要加密传输，则应以数字信封方式进行封装。

采用RequestAuthenticated定义。

A：验证B的鉴别信息

发起者A接收B发来的含有TokenAB的消息后，进行身份鉴别。

具体内容如下：

a) A根据B的标识生成B的公钥，如果A没有生成B公钥的公共参数，应从公开参数服务系统PPS中获取；

b) 检验包含在令牌中的信息；

1) 检验 TokenBA 中的 r_A是否与发送给之前发送给 B 的值相等；

2) 检验 TokenBA 中 ID_A标识段的值，是否等于实体 A 的标识符；

3) 验证数字签名的正确性,验证签名时要按照 TokenBA 的验证包按照 $ID_A \| \Delta r_A \| Text2$ 的格式进行组织；

4) 检验唯一性/时效性；

5) 解析 Text2。

若任一验证结果不正确，则停止通联，且反馈：鉴别失败信息。
若验证正确，则完成身份鉴别。

附　录　A
（规范性附录）
公共参数查询协议

A.1　描述

本附录定义了与公开参数服务系统（PPS）进行信息查询的相关协议，协议格式基于ASN.1格式规范进行编写。

A.2　获取PPS服务信息

用于获取PPS支持的IBC密钥管理基础设施或者IBC密钥管理系统的数量和类型。

```
PPSInfoRequest::= SEQUENCE{
version                 INTEGER{v1(1)},
id                      Identifier,
time                    GeneralizedTime
}
```

其中：

version为版本号项，本文中定义为1。

id为标识项，为查询者的身份标识。

time为时间项，返回的时间，采用格林威治格式。

A.3　获取PPS服务信息应答

用于PPS基本信息的回复。

```
PPSInfoResponse::= SEQUENCE{
responseCode            INTEGER,
ppsInfo                 PPSInfo,
signInfo                IBCSignInfo
}
IBCSignInfo::= SEQUENCE{
signData                IBCSignData,
algorithm               OBJECT IDENTIFIER
}
PPSInfo::= SEQUENCE{
version                 INTEGER{v1(1)},
id                      Identifier
responseKgsItem ::= SET OFKgsInfo,
time                    GeneralizedTime,
algorithm               OBJECT IDENTIFIER
}
```

```
KgsInfo ::=SEQUENCE{
kgsName                 OCTET STRING,
kgsIDInfo               Identifier,
algorithm               OBJECT IDENTIFIER
}
```

其中：

PPSInfoResponse 表示PPS的应答信息。

responseCode 返回码项，表示应答的返回码，0表示正确，其他标识错误。

PPSInfo 表示PPS的相关信息。

version 版本号项，本文中定义为1。

id 标识项，为PPS的身份标识。

responseKgsItem 返回密钥生成服务器信息的集合，代表该PPS中支持那些密钥生成服务器。

KgsInfo 密钥生成服务器信息项，如果查询成功将返回kgsName、kgsIDInfo项。

kgsName 密钥生成服务器名称项，PPS所服务的密钥生成服务器或IBC系统（仅限支持一组主密钥）的名称。

kgsIDInfo 密钥生成服务器标识项，PPS所服务的密钥生成服务器或IBC系统（仅限支持一组主密钥）的ID标识。

time 时间项，返回的时间，采用格林威治格式。

IBCSignInfo 签名信息。

signData 签名项，PPS的签名信息，内容包括PPSInfo。

algorithm 算法项，签名用的算法标识。

A.4 公开参数信息查询

用于向PPS查询IBC系统公开参数的请求。

```
PublicParameterRequest::= SEQUENCE{
version                 INTEGER{v1(1)},
id                      Identifier,
kgsIDInfo               Identifier,
time                    GeneralizedTime
}
```

其中：

version 版本号项，本文中定义为1。

id 标识项，为查询者的身份标识。

kgsIDInfo 查询条件，查询密钥生成服务器的标识信息，以PPS基本信息之一作为获取PPS中某组公开参数的查询条件。

Time 时间项，查询时间，采用格林威治格式。

A.5 公开参数信息查询应答

用于PPS对公开参数查询的回复。

```
PublicParameterResponse::= SEQUENCE{
responseCode            INTEGER,
```

```
publicParameter        PublicParameter,
signInfo               IBCSignInfo
}
IBCSignInfo::= SEQUENCE{
signData               IBCSignData,
algorithm              OBJECT IDENTIFIER
}
PublicParameter::= SEQUENCE{
version                INTEGER{v1(1)},
parameter              IBCSysParams,
id                     Identifier,
time                   GeneralizedTime,
algorithm              OBJECT IDENTIFIER
}
```

其中：

PublicParameterResponse 公共参数应答信息。

responseCode 应答码，表示应答的返回码 0标识正确其他标识错误。

publicParameter 公共参数项，是公共参数信息。

version 版本号项，本文中定义为1。

parameter 公开参数项，如果查询成功将返回公开参数内容IBCSysParams。

id 标识项，为PPS的身份标识。

time 时间项，返回的时间，采用格林威治格式。

algorithm 算法标识项，表示该kgs支持的算法。

IBCSignInfo 签名信息。

signData 签名项，PPS的签名信息，内容包括PublicParameter。

algorithm 算法项，签名用的算法标识。

A.6 用户标识查询

用于向PPS查询用户的标识参数的请求。

```
IBCUserInfoRequest ::= SEQUENCE{
version                INTEGER{v1(1)},
id                     OCTET STRING,
time                   GeneralizedTime
}
```

其中：

version 版本号项，本文中定义为1。

Id 用户ID标识项，被查询的用户标识。

time 时间项，查询时间，采用格林威治格式。

A.7 用户标识查询应答

应答内容包括：正确＋有效标识，无效＋有效标识，无效

```
IBCUserInfoResponse::= SEQUENCE{
responseCode            INTEGER,
ibcUserInfo             IBCUserInfo,
signInfo                IBCSignInfo
}
IBCSignInfo::= SEQUENCE{
algorithm               OBJECT IDENTIFIER,
signData                IBCSignData
}
IBCUserInfo ::= SEQUENCE{
version                 INTEGER{v1(1)},
usersInfo ::= SET OF UserInfo
time                    GeneralizedTime
}
UserInfo::=SEQUENCE{
userStatusCode          INTEGER,
userIDInfo              Identifier,
publishTime             GeneralizedTime
}
```

其中：

IBCUserInfoResponse 用户信息查询应答。

responseCode 返回码项，表示应答的返回码 0标识正确其他标识错误。

ibcUserInfo ibc 用户信息项。

version 版本号项，本文中定义为1。

usersInfo 用户基本信息的集合。

userStatusCode 标识状态项，表示当前用户标识信息的状态。

userIDInfo 表示用户标识信息。

publishTime 用户信息的发布时间。

time 时间项，返回的时间，采用格林威治格式。

algorithm 算法项，签名用的算法标识。

IBCSignInfo 签名信息。

signData 签名项，PPS的签名信息，内容包括IBCUserInfo。

A.8 IBC公共参数结构

```
IBCSysParams ::= SEQUENCE {
version                 INTEGER{v2(2)},
districtName            IA5String,
districtSerial          INTEGER,
validity                ValidityPeriod,
ibcPublicParameters     IBCPublicParameters,
ibcIdentityType         OBJECT IDENTIFIER,
ibcParamExtensions      IBCParamExtensions OPTIONAL
```

```
}
```

其中：

version 版本项，确定了 IBCSysParams 格式的版本。本文中提及的格式，必须设置为 2。

districtName 名称项，是一个必须以 URI 或者 IRI 编码的 IA5 字符串。

districtSerial 是一个代表了可用的唯一 IBC 公共参数(对于以 districtName 定义的 URI 或 IRI)设置的整数。如果为 districtName 公布一个新的参数，那么 districtSerial 的数值必须大于之前使用的 districtSerial 数值。

validity 有效期项，确定了一个具体 IBCSysParams 范例的寿命，并按照以下内容确定：

notBefore 与 notAfter 的数值必须以格林威治时间表示，并包含秒（如：时间表示为 YYYYMMDDHHMMSSZ)，即使是秒数为零，也要表示为最近的秒数。客户必须确认它使用的 IBC 公共参数的日期处于 IBC 公共参数的 notBefore 时间与 notAfter 时间之间，于此同时，如果日期没有处于这一区间时，不能使用用于 IBC 加密操作的参数。

当 ibcPublicParameters,ibcIdentityType 或者 ibcParamExtensions 的数值改变了一个区域时，IBC 公共参数必须重新生成与公布。客户必须在应用程序配置间隔内重新找回 IBC 公共参数，以确保参数的版本为最新。

IBCPublicParameters 公共参数项，是一个包含了公共参数（对应于 PKG 支持的 IBC 算法式）的结构。其定义如下：

```
IBCPublicParameters ::= SEQUENCE (1..MAX) OF IBCPublicParameter
IBCPublicParameter  ::= SEQUENCE {
    ibcAlgorithm          OBJECT IDENTIFIER,
    publicParameterData   OCTET STRING
}
```

其中：

ibcAlgorithm OID 确定了 IBC 算法式。两个 IBC 算法式的 OID 以及他们的 publicParameterData 结构。

publicParameterData 是一个 DER 编码结构，其包含了真实的加密参数。其具体结构取决于算法式。

ibcIdentityType 标识类型项，是一个确定在这一区域使用的身份类型的 OID。对于每一个 OID、所需要以及可选择的域都应为依赖于应用程序而存在。

IBCParamExtensions 扩散参数项，是一组用于确定特定操作所需额外参数的一组扩展。定义如下：

```
IBCParamExtensions ::= SEQUENCE OF IBCParamExtension
IBCParamExtension ::= SEQUENCE {
    ibcParamExtensionOID     OBJECT IDENTIFIER,
    ibcParamExtensionValue   OCTET STRING
}
```

其中：

ibcParamExtensionValue 的八位字符串内容由具体的 ibcParamExtensionOID 确定。一个域的 IBCParamExtensions 可能包含任何数量的扩展（包括零在内）。一个实际应用的扩展实例如下：它为电子邮件系统用户提供了一个 URI，在这里加密的信息可以被解密，同时对用户可见。另一个实例如下：它提供了商标信息以使得银行可以为处于不同业务部门的客户提供不同的用户界面。

```
ibcParamExt OBJECT IDENTIFIER ::= {
    ibcs ibcs3(3) parameter-extensions(2)
}
```

附　录　B
（规范性附录）
密钥与签名格式

B.1　密钥数据结构

密钥类型分为签名、加密主密钥和签名、加密用户密钥。

a)　SM9 算法签名主私钥数据格式的 ASN.1 定义为：

SM9SignMasterPrivateKey::=SM9MasterPrivateKey

SM9MasterPrivateKey ::=INTERGER

b)　SM9 算法签名主公钥数据格式的 ASN.1 定义为：

SM9SignMasterPublicKey::=BIT STRING

SM9SignMasterPublicKey 为 BIT STRING 类型，内容为：

04‖X_1‖X_2‖Y_1‖Y_2，其中，X_1、X_2和 Y_1、Y_2分别标识公钥的各个 x 分量和 y 分量，每个分量长度为 256 bit。或

03‖X_1‖X_2，其中，X_1、X_2分别标识公钥的各个 x 分量，每个分量长度为 256 bit。选取解压后的 Y 根值（Y_1‖Y_2）中最右边 bit 位为 1 的那个值。还原后 Y 根值取最右那个比特为 0 的值，否则 Y_1＝基域 q－*根* Y_1，Y_2＝基域 q－*根* Y_2。或

02‖X_1‖X_2，其中，X_1、X_2分别标识公钥的 2 个 x 分量，每个分量长度为 256 bit。选取解压后的 Y 根值（Y_1‖Y_2）中最右边 bit 位为 0 的选项值。还原后 Y 根值取最右一比特为 0 的选项值，否则 Y_1＝基域 q－*根* Y_1，Y_2＝基域 q－*根* Y_2。

c)　SM9 算法加密主私钥数据格式的 ASN.1 定义为：

SM9EncryptMasterPrivateKey::=SM9MasterPrivateKey

d)　SM9 算法加密主公钥数据格式的 ASN.1 定义为：

SM9EncryptMasterPublicKey::=BIT STRING

SM9EncryptMasterPublicKey 为 BIT STRING 类型，内容为：

04‖X‖Y，其中，X 和 Y 标识公钥的各个 x 分量和 y 分量，每个分量长度为 256 bit。

03‖X，其中，X 标识公钥的 x 分量，每个分量长度为 256 bit。选取解压后的 Y 根值中最右边 bit 位为 1 的那个值。还原后 Y 根值取最右那个比特为 0 的值，否则 Y＝基域 q－*根* Y。或

02‖X，其中，X 分别标识公钥的 x 分量，每个分量长度为 256 bit。选取解压后的 Y 根值中最右边 bit 位为 0 的选项值。还原后 Y 根值取最右一比特为 0 的选项值，否则 Y＝基域 q－*根* Y。

e)　SM9 算法用户签名私钥数据格式的 ASN.1 定义为：

SM9SignPrivateKey::=SM9EncryptMasterPublicKey

f)　SM9 算法用户加密私钥数据格式的 ASN.1 定义为：

SM9EncryptPrivateKey::= SM9SignMasterPublicKey

B.2　签名加密数据结构

SM9 算法签名、加密数据结构如下：

a)　签名数据结构

SM9 算法签名数据格式的 ASN.1 定义为：

```
SM9Signature ::= SEQENCE{
H               OCTET STRING,          /杂凑分量，算法是 H1（见 GM/T 0044.2）
S               SM9SignPrivateKey      /签名结果（见 GM/T 0044.2）
}
```

b) 加密数据结构

SM9 算法加密后的数据格式的 ASN.1 定义为：

```
SM9Cipher ::= SEQENCE{
EnType          INTEGER,                    /加密方式
C1              SM9SignPrivateKey,          /C1 第一部分
C3              OCTET STRING,               /明文数据杂凑值
CipherText      OCTET STRING                /密文
}
```

EnType 为加密的方式，定义 0 代表 M⊕K1 序列密码加密，1 代表分组密码加密。分组密码加密的算法为 GB/T 32907，加密模式为 ECB。加密数据应为分组长度的整倍数。加密密钥和加密运算的详细的计算过程参见 GM/T 0044.4。

C1,该部分在 GM/T 0044.4 分中被称为 C1。

C3 为 HASH，使用 GB/T 32905 算法对明文数据运算得到的杂凑值，其长度固定为 256bit。

CipherText，为加密密文，长度为 GB/T 32907 分组长度的整倍数。该部分在 GM/T 0044.4 中被称为 C2。采用序列算法模式时与明文等长。

参考文献

[1] GB/T 15843.1-2008 信息技术 安全技术 实体鉴别 第3部分:采用数字签名技术的机制

[2] GB/T 16262.1-2006 信息技术 抽象语法记法一（ASN.1）：基本记法规范(ISO/IEC 8824-1:2002, IDT)

[3] RFC5408 IETF Identity-Based Encryption Architecture and Supporting Data Structures January 2009

[4] RFC5409 IETF Using the Boneh-Franklin and Boneh-Boyen Identity-Based Encryption January 2009