ICS 35.030
CCS L 80

# 中华人民共和国国家标准

GB/T 41389—2022

# 信息安全技术
# SM9 密码算法使用规范

**Information security technology—**
**SM9 cryptographic algorithm application specification**

2022-04-15 发布　　2022-11-01 实施

国家市场监督管理总局
国家标准化管理委员会　发布

# 目　　次

前言 …… Ⅰ
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 1
4　符号和缩略语 …… 2
5　SM9 的密钥对 …… 2
5.1　生成元 …… 2
5.2　SM9 主私钥 …… 2
5.3　SM9 主公钥 …… 2
5.4　SM9 用户私钥 …… 3
5.5　SM9 用户公钥 …… 3
6　技术要求 …… 3
6.1　数据格式 …… 3
6.2　预处理 …… 5
6.3　计算过程 …… 7
7　证实方法 …… 11
7.1　数据格式 …… 11
7.2　预处理 …… 11
7.3　计算过程 …… 12
附录 A（规范性）　数据格式编码测试用例 …… 14

# 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。

本文件由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本文件起草单位：北京国脉信安科技有限公司、上海信息安全工程技术研究中心、深圳奥联信息安全技术有限公司、无锡华正天网信息安全系统有限公司、国网区块链科技(北京)有限公司。

本文件主要起草人：袁峰、王晓春、封维端、张立圆、王学进、药乐、蒋楠、程朝辉、蔡先勇、王一曲、王栋。

# 信息安全技术
# SM9 密码算法使用规范

## 1 范围

本文件规定了 SM9 密码算法的使用要求，描述了密钥、加密与签名的数据格式。

本文件适用于 SM9 密码算法的正确和规范使用，以及指导 SM9 密码算法的设备和系统的研发和检测。

## 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 16262.1—2006 信息技术 抽象语法记法一(ASN.1) 第 1 部分：基本记法规范

GB/T 17964 信息安全技术 分组密码算法的工作模式

GB/T 32905 信息安全技术 SM3 密码杂凑算法

GB/T 32907 信息安全技术 SM4 分组密码算法

GB/T 32915 信息安全技术 二元序列随机性检测方法

GB/T 35276—2017 信息安全技术 SM2 密码算法使用规范

GB/T 38635.1—2020 信息安全技术 SM9 标识密码算法 第 1 部分：总则

GB/T 38635.2—2020 信息安全技术 SM9 标识密码算法 第 2 部分：算法

## 3 术语和定义

下列术语和定义适用于本文件。

3.1

**SM9 算法 SM9 algorithm**

一种基于身份标识的椭圆曲线公钥密码算法。

3.2

**签名主密钥 signature master key**

密钥管理基础设施的根签名密钥对。

注：包括签名主私钥和签名主公钥，用于进行数字签名、验签和为用户生成用户签名密钥。

3.3

**加密主密钥 encryption master key**

密钥管理基础设施的根加密密钥对。

注：包括加密主私钥和加密主公钥，用于进行数据加密、解密和为用户生成用户加密密钥。

3.4

**用户签名密钥 user signature key**

用户的签名密钥对。

注：包括用户签名私钥和签名公钥，其中私钥由密钥管理基础设施产生并下发给用户，用于数字签名和验签。

3.5

**用户加密密钥　user encryption key**

用户的加密密钥对。

注：包括用户加密私钥和加密公钥，其中私钥由密钥管理基础设施产生并下发给用户，用于加密、解密、密钥封装和密钥交换。

## 4　符号和缩略语

下列符号和缩略语适用于本文件。

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

ID：用户身份标识(Identity)

KGC：密钥生成中心(Key Generation Center)

$e$：从 $G_1 \times G_2$ 到 $G_T$ 的双线性对

$F_q$：包含 $q$ 个元素的有限域

$F_{q^m}$：有限域 $F_q$ 的 $m$ 次扩域

$G_T$：阶为素数 $N$ 的乘法循环群

$G_1$、$G_2$：阶为素数 $N$ 的加法循环群

$N$：循环群 $G_1$、$G_2$ 和 $G_T$ 的阶

$P_1$：$G_1$ 的生成元

$P_2$：$G_2$ 的生成元

$q$：有限域 $F_q$ 中元素的数目

## 5　SM9 的密钥对

### 5.1　生成元

$G_1$ 上的生成元 $P_1$ 点，由横坐标和纵坐标两个分量来表示，记为($x_{P_1}$，$y_{P_1}$)，其中 $x_{P_1}$，$y_{P_1}$ 的长度都为 256 位，其值符合 GB/T 38635.1—2020 中 A.1 的规定。

$G_2$ 上的生成元 $P_2$ 点，由横坐标和纵坐标两个分量来表示，记为($x_{P_2}$，$y_{P_2}$)，其中 $x_{P_2}$，$y_{P_2}$ 是有限域 $F_{q^2}$ 中元素，$x_{P_2}$ 的高维和低维分量记作 $X_1$ 和 $X_2$，$y_{P_2}$ 的高维和低维分量记作 $Y_1$ 和 $Y_2$，其中 $X_1$、$X_2$、$Y_1$ 和 $Y_2$ 的长度都为 256 位，其值符合 GB/T 38635.1—2020 中 A.1 的规定。

### 5.2　SM9 主私钥

包括 SM9 签名主私钥 $ks$ 和加密主私钥 $ke$，都是一个大于或等于 1 且小于 $N$ 的整数($N$ 是循环群 $G_1$、$G_2$ 和 $G_T$ 的阶，其值符合 GB/T 38635.1—2020 中 A.1 的规定)，长度为 256 位。

### 5.3　SM9 主公钥

包括 SM9 签名主公钥 $P_{\text{pub-s}}$ 和加密主公钥 $P_{\text{pub-e}}$。分别是 $G_2$ 和 $G_1$ 上的点，记为($x_{\text{pub-s}}$，$y_{\text{pub-s}}$)和($x_{\text{pub-e}}$，$y_{\text{pub-e}}$)。其中签名主公钥的横坐标 $x_{\text{pub-s}}$ 的高维和低维分量记作 $X_1$ 和 $X_2$，纵坐标 $y_{\text{pub-s}}$ 的高维和低维分量记作 $Y_1$ 和 $Y_2$，每个分量的长度为 256 位。加密主公钥 $x_{\text{pub-e}}$ 和 $y_{\text{pub-e}}$ 坐标值长度都是 256 位。

### 5.4 SM9 用户私钥

包括 SM9 用户签名私钥 $ds$ 和用户加密私钥 $de$，分别是 $G_1$ 和 $G_2$ 上的点，记为（$x_{pri\text{-}s}$，$y_{pri\text{-}s}$）和（$x_{pri\text{-}e}$，$y_{pri\text{-}e}$）。其中用户签名私钥的横坐标 $x_{pri\text{-}s}$ 和纵坐标 $y_{pri\text{-}s}$ 的长度都是 256 位。用户加密私钥的 $x_{pri\text{-}e}$ 的高维和低维分量记作 $X_1$ 和 $X_2$，$y_{pri\text{-}e}$ 的高维和低维分量记作 $Y_1$ 和 $Y_2$，每个分量的长度为 256 位。

### 5.5 SM9 用户公钥

在 IBC 技术中，KGC 的签名主密钥（加密主密钥）确定后，用户身份标识可唯一确定用户的签名公钥（加密公钥），应用中以此代表公钥。用户签名公钥与签名主公钥结构相同，记为 $Q_D$。用户加密公钥与加密主公钥结构相同，记为 $Q_E$。

用户公钥的计算方法如下（按照 GB/T 38635.2—2020 中 6.4，7.2，8.2，9.2）。

输入：算法函数 $H_1$、ID、hid，主公钥 $P_{pub\text{-}s}$（或 $P_{pub\text{-}e}$），生成元 $P_1$、$P_2$。

输出：用户公钥 $Q_D$、$Q_E$。

计算方法：

$Q_D=[H_1(ID \| hid, N)]\ P_2+P_{pub\text{-}s}=(X_{QD}, Y_{QD})$，签名公钥用于数字签名验证；

$Q_E=[H_1(ID \| hid, N)]\ P_1+P_{pub\text{-}e}=(X_{QE}, Y_{QE})$，加密公钥用于密钥交换、密钥封装、加密/解密。

## 6 技术要求

### 6.1 数据格式

#### 6.1.1 综述

本文件按照 GB/T 16262.1—2006 的特定编码规则（DER）对 SM9 算法中的各项数据进行编码。

位串与 8 位字节串、整数与 8 位字节串的数据转换按照 GB/T 35276—2017 的第 6 章。

#### 6.1.2 生成元数据格式

$G_1$ 上的生成元 $P_1$ 点数据格式的 ASN.1 定义为：

SM9P1 ::= BIT STRING

SM9P1 为 BIT STRING 类型，内容应为：

——04 ‖ X ‖ Y，其中，X 和 Y 分别为 $P_1$ 点的 $x_{P_1}$ 分量和 $y_{P_1}$ 分量，每个分量长度为 256 位；

——03 ‖ X，其中，X 为 $P_1$ 点的 $x_{P_1}$ 分量，长度为 256 位，并且 $y_{P_1}$ 分量最右边的位为 1；

——02 ‖ X，其中，X 为 $P_1$ 点的 $x_{P_1}$ 分量，长度为 256 位，并且 $y_{P_1}$ 分量最右边的位为 0。

$G_2$ 上的生成元 $P_2$ 点数据格式的 ASN.1 定义为：

SM9P2 ::= BIT STRING

SM9P2 为 BIT STRING 类型，内容应为：

——04 ‖ $X_1$ ‖ $X_2$ ‖ $Y_1$ ‖ $Y_2$，其中，$X_1$、$X_2$ 是 $P_2$ 点的 $x_{P_2}$ 的高维和低维分量，$Y_1$、$Y_2$ 是 $y_{P_2}$ 的高维和低维分量，其长度各为 256 位；

——03 ‖ $X_1$ ‖ $X_2$，其中，$X_1$、$X_2$ 是 $x_{P_2}$ 的高维和低维分量，其长度各为 256 位，并且 $y_{P_2}$ 的低维分量 $Y_2$ 最右边的位为 1；

——02 ‖ $X_1$ ‖ $X_2$，其中，$X_1$、$X_2$ 是 $x_{P_2}$ 的高维和低维分量，其长度各为 256 位，并且 $y_{P_2}$ 的低维分量 $Y_2$ 最右边的位为 0。

### 6.1.3 密钥数据格式

密钥类型分为签名主密钥、加密主密钥、用户签名密钥和用户加密密钥。

SM9 算法签名主私钥数据格式的 ASN.1 定义为：

SM9SignMasterPrivateKey ::= SM9PrivateKey

SM9PrivateKey ::= INTERGER

SM9 算法签名主公钥数据格式的 ASN.1 定义为：

SM9SignMasterPublicKey ::= SM9KeyBlob2

SM9KeyBlob2 ::= BIT STRING

SM9KeyBlob2 为 BIT STRING 类型，内容应为：

——04 ‖ $X_1$ ‖ $X_2$ ‖ $Y_1$ ‖ $Y_2$，其中，$X_1$、$X_2$ 是 $x_{pub\text{-}s}$ 的高维和低维分量，$Y_1$、$Y_2$ 是 $y_{pub\text{-}s}$ 的高维和低维分量，其长度各为 256 位；

——03 ‖ $X_1$ ‖ $X_2$，其中，$X_1$、$X_2$ 是 $x_{pub\text{-}s}$ 的高维和低维分量，其长度各为 256 位，并且 $y_{pub\text{-}s}$ 的低维分量 $Y_2$ 最右边的位为 1；

——02 ‖ $X_1$ ‖ $X_2$，其中，$X_1$、$X_2$ 是 $x_{pub\text{-}s}$ 的高维和低维分量，其长度各为 256 位，并且 $y_{pub\text{-}s}$ 的低维分量 $Y_2$ 最右边的位为 0。

SM9 算法加密主私钥数据格式的 ASN.1 定义为：

SM9EncryptMasterPrivateKey ::= SM9PrivateKey

SM9 算法加密主公钥数据格式的 ASN.1 定义为：

SM9EncryptMasterPublicKey ::= SM9KeyBlob1

SM9KeyBlob1::=BIT STRING

SM9KeyBlob1 为 BIT STRING 类型，内容应为：

——04 ‖ X ‖ Y，其中，X 和 Y 分别为 $P_{pub\text{-}e}$ 点的 $x_{pub\text{-}e}$ 分量和 $y_{pub\text{-}e}$ 分量，每个分量长度为 256 位；

——03 ‖ X，其中，X 为 $P_{pub\text{-}e}$ 点的 $x_{pub\text{-}e}$ 分量，长度为 256 位，并且 $y_{pub\text{-}e}$ 分量最右边的位为 1；

——02 ‖ X，其中，X 为 $P_{pub\text{-}e}$ 点的 $x_{pub\text{-}e}$ 分量，长度为 256 位，并且 $y_{pub\text{-}e}$ 分量最右边的位为 0。

SM9 算法用户签名密钥包括用户签名私钥和用户签名公钥，其中用户签名私钥数据格式的 ASN.1 定义为：

SM9SignPrivateKey ::= SM9KeyBlob1

SM9 算法用户加密密钥包括用户加密私钥和用户加密公钥，其中用户加密私钥数据格式的 ASN.1 定义为：

SM9EncryptPrivateKey ::= SM9KeyBlob2

### 6.1.4 签名数据格式

SM9 算法签名数据格式的 ASN.1 定义为：

```
SM9Signature ::= SEQUENCE{
H     OCTET STRING,      ——杂凑分量，算法是 H2（符合 GB/T 38635.2—2020）
S     SM9KeyBlob1        ——签名结果（符合 GB/T 38635.2—2020）
}
```

### 6.1.5 加密数据格式

SM9 算法加密数据格式的 ASN.1 定义为：

```
SM9Cipher ::= SEQUENCE{
EnType   INTEGER,        ——加密方式
```

C1　　SM9KeyBlob1，　——密文第一部分 $C_1$（符合 GB/T 38635.2—2020）
C3　　OCTET STRING，——密文杂凑值
C2　　OCTET STRING ——密文
}

EnType 为加密的方式，内容为 0 代表 M ⊕ K1 序列密码加密，内容为 1、2、4、8 分别代表 ECB、CBC、OFB、CFB 分组密码算法工作模式，分组密码算法工作模式和初始值Ⅳ遵守 GB/T 17964 的规定。分组密码加密的算法按照 GB/T 32907。

C1 在 GB/T 38635.2—2020 的 9.2 中被称为 $C_1$。

C3 在 GB/T 38635.2—2020 的 9.2 中被称为 $C_3$，为按照 GB/T 32905 算法对明文数据运算得到的杂凑值，其长度固定为 256 位。

C2 在 GB/T 38635.2—2020 的 9.2 中被称为 $C_2$，为加密密文。

#### 6.1.6　密钥封装数据格式

用户 A 将一个随机数封装成 C 后，并传递给用户 B，以便计算出密钥 K。

密钥封装数据格式的 ASN.1 定义为：

SM9KeyPackage ::= SEQUENCE{
K　　OCTET STRING，　——生成的密钥
C　　SM9KeyBlob1　　——封装的交换密文
}

K 作为用户 A 保留的密钥。C 作为交换密文传递给 B 用户，B 用户利用 C 可以生成 K。

### 6.2　预处理

#### 6.2.1　预处理杂凑函数 $\mathbf{H}_1$

验签、加密时应按 GB/T 38635.2—2020 中 5.3.2.2 进行预处理计算。

输入：
DATA　比特串　——数据
N　　整型　　——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶
输出：
h1　　整型　　——长度为 256 位，且 $1 \leqslant h1 \leqslant N-1$

#### 6.2.2　预处理杂凑函数 $\mathbf{H}_2$

签名时应按 GB/T 38635.2—2020 中 5.3.2.3 进行预处理计算。

输入：
DATA　比特串　——数据
N　　整型　　——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶
输出：
h2　　整型　　——长度为 256 位，且 $1 \leqslant h2 \leqslant N-1$

#### 6.2.3　预处理对运算 1

预处理对运算 1 是指使用 KGC 的签名主公钥和 $G_1$ 的生成元 $P_1$ 点，计算 $G_T$ 中的元素 $g_1$。该结果用于 SM9 数字签名。

预处理对运算 1 应按 GB/T 38635.1—2020 附录 A 规定的双线性对进行计算。

输入：

| | | |
|---|---|---|
| $P_1$ | SM9P1 | ——生成元 $P_1$ |
| $P_{pub-s}$ | SM9SignMasterPublicKey | ——签名主公钥 |

输出：

| | | |
|---|---|---|
| $g_1$ | 比特串 | ——群 $G_T$ 中元素，长度为 3 072 位 |

计算公式为：

$g_1 = e(P_1, P_{pub-s})$

双线性对 $e$ 的详细计算过程按照 GB/T 38635.1—2020 中 C.6。

#### 6.2.4 预处理对运算 2

预处理对运算 2 是指使用 KGC 的加密主公钥和 $G_2$ 的生成元 $P_2$ 点，计算 $G_T$ 中的元素 $g_2$。该结果用于 SM9 密钥交换、密钥封装和加密。

预处理对运算 2 应按 GB/T 38635.1—2020 附录 A 规定的双线性对进行计算。

输入：

| | | |
|---|---|---|
| $P_2$ | SM9P2 | ——生成元 $P_2$ |
| $P_{pub-e}$ | SM9EncryptMasterPublicKey | ——加密主公钥 |

输出：

| | | |
|---|---|---|
| $g_2$ | 比特串 | ——群 $G_T$ 中元素，长度为 3 072 位 |

计算公式为：

$g_2 = e(P_{pub-e}, P_2)$

双线性对 $e$ 的详细计算过程按照 GB/T 38635.1—2020 中 C.6。

#### 6.2.5 预处理用户验签公钥 $Q_D$

预处理用户验签公钥 $Q_D$ 是指将身份标识字符串变换为群 $G_2$ 上的点 $Q_D$，该结果用于数字签名验证计算过程中。

预处理用户验签公钥 $Q_D$ 应按 GB/T 38635.2—2020 中 6.4 的 B5 和 B6 进行计算。

输入：

| | | |
|---|---|---|
| ID | 字节串 | ——用户身份标识 |
| hid | 整型 | ——KGC 私钥生成函数识别符，取值为 1 |
| N | 整型 | ——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶 |
| $P_2$ | SM9P2 | ——生成元 $P_2$ |
| $P_{pub-s}$ | SM9SignMasterPublicKey | ——签名主公钥 |

输出：

| | | |
|---|---|---|
| $Q_D$ | SM9KeyBlob2 | ——群 $G_2$ 上的点 |

计算公式为：

$Q_D = [h1]P_2 + P_{pub-s}$

其中 h1 的计算见 6.2.1。

#### 6.2.6 预处理用户加密公钥 $Q_E$

预处理用户加密公钥 $Q_E$ 是指将身份标识 ID 字符串变换为群 $G_1$ 上的点 $Q_E$，该结果用于密钥交换、密钥封装和加密的计算过程中。

预处理用户加密公钥 $Q_E$ 应按 GB/T 38635.2—2020 中 7.2 的 B1（或 GB/T 38635.2—2020 中 8.2 的 A1，或 GB/T 38635.2—2020 中 9.2 的 A1）进行计算。

输入：

| | | |
|---|---|---|
| ID | 字节串 | ——用户身份标识串 |
| hid | 整型 | ——KGC 私钥生成函数识别符，取值为 3 |
| N | 整型 | ——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶 |
| $P_1$ | SM9P1 | ——生成元 $P_1$ |
| $P_{pub-e}$ | SM9EncryptMasterPublicKey | ——加密主公钥 |

输出：

| | | |
|---|---|---|
| $Q_E$ | SM9KeyBlob1 | ——群 $G_1$ 上的点 |

计算公式为：

$Q_E=[h1]P_1+P_{pub-e}$

其中 h1 的计算见 6.2.1。

## 6.3 计算过程

### 6.3.1 生成密钥

密钥生成是指生成 SM9 算法的主私/公钥、用户私钥的过程。

a) 主私钥生成

SM9 签名主私钥和加密主私钥是由 KGC 产生的随机数生成，并且大于或等于 1 且小于 $N$，其中 $N$ 为群 $G_1$、$G_2$ 和 $G_T$ 的阶。

KGC 产生的随机数应符合 GB/T 32915 的要求。

输入：

无

输出：

| | | |
|---|---|---|
| k | SM9PrivateKey | ——SM9 主私钥，如果为签名主私钥标识为 $k_S$，加密主私钥标识为 $k_E$ |

b) 主公钥生成

SM9 签名主公钥和加密主公钥分别由相对应的主私钥与生成元 $P_1$ 和 $P_2$ 的点乘生成。

签名主公钥生成过程应为：

输入：

| | | |
|---|---|---|
| $k_S$ | SM9SignMasterPrivateKey | ——签名主私钥 |
| $P_2$ | SM9P2 | ——群 $G_2$ 的生成元 |

输出：

| | | |
|---|---|---|
| $P_{pub-s}$ | SM9SignMasterPublicKey | ——签名主公钥 |

加密主公钥生成过程应为：

输入：

| | | |
|---|---|---|
| $k_E$ | SM9EncryptMasterPrivateKey | ——加密主私钥 |
| $P_1$ | SM9P1 | ——群 $G_1$ 的生成元 |

输出：

| | | |
|---|---|---|
| $P_{pub-e}$ | SM9EncryptMasterPublicKey | ——加密主公钥 |

详细的计算过程见 GB/T 38635.2—2020 中 9.1。

c) 用户私钥生成

用户私钥生成是指生成用户私钥的过程，用户私钥分用户签名私钥和用户加密私钥，分别与相对应的主私钥与 $G_1$ 的生成元 $P_1$ 和 $G_2$ 的生成元 $P_2$ 有关。

用户签名私钥生成过程应为：

输入：

| | | |
|---|---|---|
| $k_S$ | SM9SignMasterPrivateKey | ——签名主私钥 |
| ID | 字节串 | ——用户身份标识串 |
| hid | 整型 | ——KGC 私钥生成函数标识符，取值为 1 |
| N | 整型 | ——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶 |
| $P_1$ | SM9P1 | ——群 $G_1$ 的生成元 |

输出：

| | | |
|---|---|---|
| $d_S$ | SM9SignPrivateKey | ——SM9 用户签名私钥 |

用户加密私钥生成过程应为：

输入：

| | | |
|---|---|---|
| $k_E$ | SM9EncryptMasterPrivateKey | ——加密主私钥 |
| ID | 字节串 | ——用户身份标识串 |
| hid | 整型 | ——KGC 私钥生成函数标识符，取值为 3 |
| N | 整型 | ——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶 |
| $P_2$ | SM9P2 | ——群 $G_2$ 的生成元 |

输出：

| | | |
|---|---|---|
| $d_E$ | SM9EncryptPrivateKey | ——SM9 用户加密私钥 |

详细的计算过程按照 GB/T 38635.2—2020 中 9.1。

### 6.3.2 数字签名

SM9 签名是指使用预处理对运算 1 的结果和签名者私钥，通过签名计算得到签名结果的过程。

数字签名过程应为：

输入：

| | | |
|---|---|---|
| $g_1$ | 比特串 | ——预处理对运算 1 的结果 |
| N | 整型 | ——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶 |
| M | 比特串 | ——待签名数据 |
| $d_S$ | SM9SignPrivateKey | ——签名者私钥 |

输出：

| | | |
|---|---|---|
| sign | SM9Signature | ——签名值 |

详细的计算过程按照 GB/T 38635.2—2020 中 6.2。

### 6.3.3 签名验证

SM9 签名验证是指使用预处理对运算 1 的结果、签名者标识、签名值、被签名数据，通过验签计算确定签名是否通过验证的过程。

签名验证过程应为：

输入：

| | | |
|---|---|---|
| ID | 字节串 | ——签名者标识 |
| hid | 整型 | ——私钥生成函数标识符 |
| N | 整型 | ——循环群 $G_1$、$G_2$ 和 $G_T$ 的阶 |
| $g_1$ | 比特串 | ——预处理对运算 1 的结果 |
| $Q_D$ | SM9KeyBlob2 | ——预处理用户验签公钥 $Q_D$ 的结果 |
| M | 比特串 | ——被签名数据 |
| sign | SM9Signature | ——签名值 |

输出：

为“真”表示“验证通过”，为“假”表示“验证不通过”。

详细的计算过程按照 GB/T 38635.2—2020 中 6.4。

#### 6.3.4 密钥封装

SM9 密钥封装是指使用对方身份标识通过双线性对运算生成密钥，并产生基于对方加密公钥的密文即加密封装的密文。

密钥封装过程应为：

输入：

| | | |
|---|---|---|
| KeyLen | 整型 | ——生成密钥的长度 |
| ID | 字节串 | ——对方身份标识 |
| $g_2$ | 比特串 | ——预处理对运算 2 的结果 |
| $Q_E$ | SM9KeyBlob1 | ——对方的预处理用户加密公钥 $Q_E$ 的结果 |

输出：

| | | |
|---|---|---|
| KeyC | SM9KeyPackage | ——封装密文 |

详细的计算过程按照 GB/T 38635.2—2020 中 8.2。

#### 6.3.5 密钥解封装

SM9 密钥解封装是指用户收到封装密文后，对密文进行解封装的计算过程。

密钥解封装过程应为：

输入：

| | | |
|---|---|---|
| ID | 字节串 | ——解封装方身份标识 |
| $d_e$ | SM9EncryptPrivateKey | ——解封装方加密私钥 |
| C | SM9KeyBlob1 | ——封装密文 |

输出：

| | | |
|---|---|---|
| Key | 比特串 | ——密钥 |

详细的计算过程按照 GB/T 38635.2—2020 中 8.4。

#### 6.3.6 加密

SM9 加密是指使用指定公开密钥对明文进行特定的加密计算，生成相应密文的过程。该密文只能由该指定公开密钥对应的私钥解密。

加密过程应为：

输入：

| | | |
|---|---|---|
| M | 字节串 | ——明文 |
| EnType | 整型 | ——加密方式，见 6.1.5 |
| $g_2$ | 比特串 | ——预处理对运算 2 的结果 |
| $Q_E$ | SM9KeyBlob1 | ——对方的预处理用户加密公钥 $Q_E$ 的结果 |

输出：

| | | |
|---|---|---|
| C | SM9Cipher | ——密文 |

详细的计算过程按照 GB/T 38635.2—2020 中 9.2。

#### 6.3.7 解密

SM9 解密是指使用指定私钥对密文进行解密计算，还原对应明文的过程。

解密过程应为：

输入：

| | | |
|---|---|---|
| ID | 字节串 | ——解密方身份标识 |
| EnType | 整型 | ——加密方式，见 6.1.5 |
| $d_e$ | SM9EncryptPrivateKey | ——解密方加密私钥 |
| C | SM9Cipher | ——密文 |

输出：

| | | |
|---|---|---|
| M | 字节串 | ——明文 |

详细的计算过程按照 GB/T 38635.2—2020 中 9.4。

### 6.3.8 密钥交换

密钥交换是在两个用户之间建立一个共享秘密密钥的协商过程，通过这种方式能够确定一个共享秘密密钥的值。

设密钥交换双方身份标识为 $ID_A$、$ID_B$，分别计算预处理用户加密 $Q_E$ 的结果为 $Q_{EA}$ 和 $Q_{EB}$，其用户私钥分别为 $d_A$ 和 $d_B$，双方需要获得的密钥数据的比特长度为 klen。密钥交换协议分为两个阶段。

密钥交换过程应为：

第一阶段：产生临时密钥对

用户 A：

输入：

| | | |
|---|---|---|
| $Q_{EB}$ | SM9KeyBlob1 | ——用户 B 的预处理用户加密公钥 $Q_E$ 的结果 |

输出：

| | | |
|---|---|---|
| $R_A$ | SM9KeyBlob1 | ——用户 A 的临时公钥 |

用户 B：

输入：

| | | |
|---|---|---|
| $Q_{EA}$ | SM9KeyBlob1 | ——用户 A 的预处理用户加密公钥 $Q_E$ 的结果 |

输出：

| | | |
|---|---|---|
| $R_B$ | SM9KeyBlob1 | ——用户 B 的临时公钥 |

详细的计算过程按照 GB/T 38635.2—2020 中 7.1。

第二阶段：计算共享秘密密钥

用户 B：

输入：

| | | |
|---|---|---|
| $R_A$ | SM9KeyBlob1 | ——用户 A 的临时公钥 |
| $R_B$ | SM9KeyBlob1 | ——用户 B 的临时公钥 |
| $ID_A$ | 字节串 | ——用户 A 的身份标识 |
| $ID_B$ | 字节串 | ——用户 B 的身份标识 |
| $P_{pub-e}$ | SM9EncryptMasterPublicKey | ——加密主公钥 |
| $P_2$ | SM9P2 | ——生成元 $P_2$ |
| $d_B$ | SM9EncryptPrivateKey | ——用户 B 加密私钥 |
| klen | 整型 | ——需要输出的密钥数据的比特长度 |

输出：

| | | |
|---|---|---|
| $S_B$ | 比特串 | ——可选项，校验值，用于用户 A 校验 $S_1$ |
| $S_2$ | 比特串 | ——可选项，用于对比 $S_A$ 的校验值 |
| $SK_B$ | 比特串 | ——位长为 klen 的密钥数据 |

详细的计算过程按照 GB/T 38635.2—2020 中 7.2。

用户 A：

输入：

| | | |
|---|---|---|
| $R_A$ | SM9KeyBlob1 | ——A 临时密钥 |
| $R_B$ | SM9KeyBlob1 | ——B 临时密钥 |
| $ID_A$ | 字节串 | ——A 的身份标识 |
| $ID_B$ | 字节串 | ——B 的身份标识 |
| $P_{pub-e}$ | SM9EncryptMasterPublicKey | ——加密主公钥 |
| $P_2$ | SM9P2 | ——生成元 $P_2$ |
| $d_A$ | SM9EncryptPrivateKey | ——用户 A 加密私钥 |
| klen | 整型 | ——需要输出的密钥数据的比特长度 |

输出：

| | | |
|---|---|---|
| $S_A$ | 比特串 | ——可选项，校验值，用于用户 B 校验 $S_2$ |
| $S_1$ | 比特串 | ——可选项，用于对比 $S_B$ 的校验值 |
| $SK_A$ | 比特串 | ——位长为 klen 的密钥数据 |

详细的计算过程按照 GB/T 38635.2—2020 中 7.2。

## 7 证实方法

### 7.1 数据格式

#### 7.1.1 生成元数据结构

$G_1$ 上的生成元 $P_1$ 点和 $G_2$ 上的生成元 $P_2$ 点数据结构按 GB/T 16262.1—2006 进行编码，使用 A.2 的测试用例测试通过。

#### 7.1.2 密钥数据结构

签名主私钥数据结构、签名主公钥数据结构、加密主私钥数据结构、加密主公钥数据结构、用户签名私钥数据结构和用户加密私钥数据结构按 GB/T 16262.1—2006 进行编码，使用 A.3 的测试用例测试通过。

#### 7.1.3 签名数据结构

签名数据结构按 GB/T 16262.1—2006 进行编码，使用 A.4 的测试用例测试通过。

#### 7.1.4 加密数据结构

加密数据结构按 GB/T 16262.1—2006 进行编码，使用 A.5 的测试用例测试通过。

#### 7.1.5 密钥封装数据结构

密钥封装数据结构按 GB/T 16262.1—2006 进行编码，使用 A.6 的测试用例测试通过。

### 7.2 预处理

#### 7.2.1 预处理杂凑函数 $H_1$

预处理杂凑函数 $H_1$ 按 GB/T 38635.2—2020 中 5.3.2.2 进行计算。

计算实例见 GB/T 38635.2—2020 中 A.2。

#### 7.2.2 预处理杂凑函数 $H_2$

预处理杂凑函数 $H_2$ 按 GB/T 38635.2—2020 中 5.3.2.3 进行计算。

计算实例见 GB/T 38635.2—2020 中 A.2。

#### 7.2.3 预处理对运算 1

预处理对运算 1 计算按 GB/T 38635.1—2020 中 A.1 和 C.6.2 进行计算。

计算实例见 GB/T 38635.2—2020 中 A.2。

#### 7.2.4 预处理对运算 2

预处理对运算 2 计算按 GB/T 38635.1—2020 中 A.1 和 C.6.2 进行计算。

计算实例见 GB/T 38635.2—2020 中 A.3、A.4 或 A.5。

#### 7.2.5 预处理用户验签公钥 $Q_D$

预处理用户验签公钥 $Q_D$ 按 GB/T 38635.2—2020 中 6.4 的 B5 和 B6 进行计算。

计算实例见 GB/T 38635.2—2020 中 A.3、A.4 或 A.5。

#### 7.2.6 预处理用户加密公钥 $Q_E$

预处理用户加密公钥 $Q_E$ 按 GB/T 38635.2—2020 中 7.2 的 B1(或 GB/T 38635.2—2020 中 8.2 的 A1,或 GB/T 38635.2—2020 中 9.2 的 A1)进行计算。

计算实例见 GB/T 38635.2—2020 中 A.3、A.4 或 A.5。

### 7.3 计算过程

#### 7.3.1 生成密钥

生成密钥包括主密钥生成、主公钥生成和用户私钥生成。

a) 主私钥生成

KGC 的随机数发生器按照 GB/T 32915 进行验证。

签名主私钥生成按 GB/T 38635.2—2020 中 6.1 进行计算,计算实例见 GB/T 38635.2—2020 中 A.2。

加密主私钥生成按 GB/T 38635.2—2020 中 7.1、8.1 或 9.1 进行计算,计算实例见 GB/T 38635.2—2020 中 A.3、A.4 或 A.5。

b) 主公钥生成

签名主公钥生成按 GB/T 38635.2—2020 中 6.1 进行计算,计算实例见 GB/T 38635.2—2020 中 A.2。

加密主公钥生成按 GB/T 38635.2—2020 中 7.1、8.1 或 9.1 进行计算,计算实例见 GB/T 38635.2—2020 中 A.3、A.4 或 A.5。

c) 用户私钥生成

用户签名私钥生成按 GB/T 38635.2—2020 中 6.1 进行计算,计算实例见 GB/T 38635.2—2020 中 A.2。

用户加密私钥生成按 GB/T 38635.2—2020 中 7.1、8.1 或 9.1 进行计算,计算实例见 GB/T 38635.2—2020 中 A.3、A.4 或 A.5。

#### 7.3.2 数字签名

数字签名按 GB/T 38635.2—2020 中 6.2 进行计算，计算实例见 GB/T 38635.2—2020 中 A.2。

#### 7.3.3 签名验证

签名验证按 GB/T 38635.2—2020 中 6.4 进行计算，计算实例见 GB/T 38635.2—2020 中 A.2。

#### 7.3.4 密钥封装

密钥封装按 GB/T 38635.2—2020 中 8.2 进行计算，计算实例见 GB/T 38635.2—2020 中 A.4。

#### 7.3.5 密钥解封装

密钥解封装按 GB/T 38635.2—2020 中 8.4 进行计算，计算实例见 GB/T 38635.2—2020 中 A.4。

#### 7.3.6 加密

加密按 GB/T 38635.2—2020 中 9.2 进行计算，计算实例见 GB/T 38635.2—2020 中 A.5。

#### 7.3.7 解密

解密按 GB/T 38635.2—2020 中 9.4 进行计算，计算实例见 GB/T 38635.2—2020 中 A.5。

#### 7.3.8 密钥交换

密钥交换按 GB/T 38635.2—2020 中 7.2 进行计算，计算实例见 GB/T 38635.2—2020 中 A.3。

# 附 录 A
（规范性）
数据格式编码测试用例

## A.1 概述

本附录给出数据格式的编码测试用例。

本附录中，所有用16进制表示的数，左边为高位，右边为低位。

本附录中的数据来源见GB/T 38635.2—2020中附录A。

## A.2 生成元数据格式编码测试用例

群 $G_1$ 的生成元 $P_1=(x_{P_1},y_{P_1})$

坐标 $x_{P_1}$：93DE051D 62BF718F F5ED0704 487D01D6 E1E40869 09DC3280 E8C4E481 7C66DDDD

坐标 $y_{P_1}$：21FE8DDA 4F21E607 63106512 5C395BBC 1C1C00CB FA602435 0C464CD7 0A3EA616

群 $G_2$ 的生成元 $P_2=(x_{P_2},y_{P_2})$

坐标 $x_{P_2}$：(85AEF3D0 78640C98 597B6027 B441A01F F1DD2C19 0F5E93C4 54806C11 D8806141,
37227552 92130B08 D2AAB97F D34EC120 EE265948 D19C17AB F9B7213B AF82D65B)

坐标 $y_{P_2}$：(17509B09 2E845C12 66BA0D26 2CBEE6ED 0736A96F A347C8BD 856DC76B 84EBEB96,
A7CF28D5 19BE3DA6 5F317015 3D278FF2 47EFBA98 A71A0811 6215BBA5 C999A7C7)

$P_1$ 的编码如下。

```
03 42:   BIT STRING
      :     00 04 93 DE 05 1D 62 BF 71 8F F5 ED 07 04 48 7D
      :     01 D6 E1 E4 08 69 09 DC 32 80 E8 C4 E4 81 7C 66
      :     DD DD 21 FE 8D DA 4F 21 E6 07 63 10 65 12 5C 39
      :     5B BC 1C 1C 00 CB FA 60 24 35 0C 46 4C D7 0A 3E
      :     A6 16
```

$P_2$ 的编码如下。

```
03 81 82:   BIT STRING
      :     00 04 85 AE F3 D0 78 64 0C 98 59 7B 60 27 B4 41
      :     A0 1F F1 DD 2C 19 0F 5E 93 C4 54 80 6C 11 D8 80
      :     61 41 37 22 75 52 92 13 0B 08 D2 AA B9 7F D3 4E
      :     C1 20 EE 26 59 48 D1 9C 17 AB F9 B7 21 3B AF 82
      :     D6 5B 17 50 9B 09 2E 84 5C 12 66 BA 0D 26 2C BE
      :     E6 ED 07 36 A9 6F A3 47 C8 BD 85 6D C7 6B 84 EB
      :     EB 96 A7 CF 28 D5 19 BE 3D A6 5F 31 70 15 3D 27
      :     8F F2 47 EF BA 98 A7 1A 08 11 62 15 BB A5 C9 99
      :     A7 C7
```

## A.3 密钥数据格式编码测试用例

签名主私钥 $ks$：0130E7 8459D785 45CB54C5 87E02CF4 80CE0B66 340F319F 348A1D5B 1F2DC5F4

签名主公钥 $P_{\text{pub-s}}=[ks]P_2=(x_{P_{\text{pub-s}}},y_{P_{\text{pub-s}}})$

坐标 $x_{P_{\text{pub-s}}}$：(9F64080B 3084F733 E48AFF4B 41B56501 1CE0711C 5E392CFB 0AB1B679 1B94C408,
29DBA116 152D1F78 6CE843ED 24A3B573 414D2177 386A92DD 8F14D656 96EA5E32)
坐标 $y_{P_{\text{pub-s}}}$：(69850938 ABEA0112 B57329F4 47E3A0CB AD3E2FDB 1A77F335 E89E1408 D0EF1C25,
41E00A53 DDA532DA 1A7CE027 B7A46F74 1006E85F 5CDFF073 0E75C05F B4E3216D)
加密主私钥 $ke$：02E65B 0762D042 F51F0D23 542B13ED 8CFA2E9A 0E720636 1E013A28 3905E31F
加密主公钥 $P_{\text{pub-e}} = [ke]P_1 = (x_{P_{\text{pub-e}}}, y_{P_{\text{pub-e}}})$
坐标 $x_{P_{\text{pub-e}}}$：91745426 68E8F14A B273C094 5C3690C6 6E5DD096 78B86F73 4C435056 7ED06283
坐标 $y_{P_{\text{pub-e}}}$：54E598C6 BF749A3D ACC9FFFE DD9DB686 6C50457C FC7AA2A4 AD65C316 8FF74210
实体 A 的标识 $\text{ID}_\text{A}$：Alice
$\text{ID}_\text{A}$ 的 16 进制表示：416C6963 65
用户签名私钥 $ds_\text{A} = [t_2]P_1 = (x_{ds_\text{A}}, y_{ds_\text{A}})$
坐标 $x_{ds_\text{A}}$：A5702F05 CF131530 5E2D6EB6 4B0DEB92 3DB1A0BC F0CAFF90 523AC875 4AA69820
坐标 $y_{ds_\text{A}}$：78559A84 4411F982 5C109F5E E3F52D72 0DD01785 392A727B B1556952 B2B013D3
用户加密私钥 $de_\text{A}=[t_2]P_2= (x_{de_\text{A}}, y_{de_\text{A}})$：
坐标 $x_{de_\text{A}}$：(4C5EC9C8 CA8DEBA2 38CC3E50 0458F514 7911F225 1A4BD0AA 903BB5F8 D5FD23B4,
0360DBBD D69A0573 0775BB3F 8AD799CC 571DCB88 3D417B8D 239302BD 90097C6B)
坐标 $y_{de_\text{A}}$：(21F05A64 F6592874 00F2D202 72329F2A 80EB6076 7C9FF9D2 3CE8046A F5C950D0,
68AFFFD5 03C768A7 65731F62 FC3CB7B7 705456D4 0830E868 CC17A7F9 51855678)
签名主私钥 $ks$ 的编码如下。

```
02 1F:   INTERGER
      :     01 30 E7 84 59 D7 85 45 CB 54 C5 87 E0 2C F4 80
      :     CE 0B 66 34 0F 31 9F 34 8A 1D 5B 1F 2D C5 F4
```

签名主公钥 $P_{\text{pub-s}}$ 的编码如下。

```
03 81 82:   BIT STRING
      :     00 04 9F 64 08 0B 30 84 F7 33 E4 8A FF 4B 41 B5
      :     65 01 1C E0 71 1C 5E 39 2C FB 0A B1 B6 79 1B 94
      :     C4 08 29 DB A1 16 15 2D 1F 78 6C E8 43 ED 24 A3
      :     B5 73 41 4D 21 77 38 6A 92 DD 8F 14 D6 56 96 EA
      :     5E 32 69 85 09 38 AB EA 01 12 B5 73 29 F4 47 E3
      :     A0 CB AD 3E 2F DB 1A 77 F3 35 E8 9E 14 08 D0 EF
      :     1C 25 41 E0 0A 53 DD A5 32 DA 1A 7C E0 27 B7 A4
      :     6F 74 10 06 E8 5F 5C DF F0 73 0E 75 C0 5F B4 E3
      :     21 6D
```

加密主私钥 $ke$ 的编码如下。

```
02 1F:   INTERGER
      :     02 E6 5B 07 62 D0 42 F5 1F 0D 23 54 2B 13 ED 8C
      :     FA 2E 9A 0E 72 06 36 1E 01 3A 28 39 05 E3 1F
```

加密主公钥 $P_{\text{pub-e}}$ 的编码如下。

```
03 42:   BIT STRING
      :     00 04 91 74 54 26 68 E8 F1 4A B2 73 C0 94 5C 36
      :     90 C6 6E 5D D0 96 78 B8 6F 73 4C 43 50 56 7E D0
      :     62 83 54 E5 98 C6 BF 74 9A 3D AC C9 FF FE DD 9D
```

：　B6 86 6C 50 45 7C FC 7A A2 A4 AD 65 C3 16 8F F7
：　42 10

用户签名私钥的编码如下。

03 42：　BIT STRING
：　00 04 A5 70 2F 05 CF 13 15 30 5E 2D 6E B6 4B 0D
：　EB 92 3D B1 A0 BC F0 CA FF 90 52 3A C8 75 4A A6
：　98 20 78 55 9A 84 44 11 F9 82 5C 10 9F 5E E3 F5
：　2D 72 0D D0 17 85 39 2A 72 7B B1 55 69 52 B2 B0
：　13 D3

用户加密私钥的编码如下。

03 81 82：　BIT STRING
：　00 04 4C 5E C9 C8 CA 8D EB A2 38 CC 3E 50 04 58
：　F5 14 79 11 F2 25 1A 4B D0 AA 90 3B B5 F8 D5 FD
：　23 B4 03 60 DB BD D6 9A 05 73 07 75 BB 3F 8A D7
：　99 CC 57 1D CB 88 3D 41 7B 8D 23 93 02 BD 90 09
：　7C 6B 21 F0 5A 64 F6 59 28 74 00 F2 D2 02 72 32
：　9F 2A 80 EB 60 76 7C 9F F9 D2 3C E8 04 6A F5 C9
：　50 D0 68 AF FF D5 03 C7 68 A7 65 73 1F 62 FC 3C
：　B7 B7 70 54 56 D4 08 30 E8 68 CC 17 A7 F9 51 85
：　56 78

## A.4　签名数据格式编码测试用例

签名主私钥 $ks$：0130E7 8459D785 45CB54C5 87E02CF4 80CE0B66 340F319F 348A1D5B 1F2DC5F4

签名主公钥 $P_{\text{pub-s}} = [ks]P_2 = (x_{P_{\text{pub-s}}}, y_{P_{\text{pub-s}}})$

坐标 $x_{P_{\text{pub-s}}}$：(9F64080B 3084F733 E48AFF4B 41B56501 1CE0711C 5E392CFB 0AB1B679 1B94C408,
29DBA116 152D1F78 6CE843ED 24A3B573 414D2177 386A92DD 8F14D656 96EA5E32)

坐标 $y_{P_{\text{pub-s}}}$：(69850938 ABEA0112 B57329F4 47E3A0CB AD3E2FDB 1A77F335 E89E1408 D0EF1C25,
41E00A53 DDA532DA 1A7CE027 B7A46F74 1006E85F 5CDFF073 0E75C05F B4E3216D)

实体 A 的标识 $\text{ID}_\text{A}$：Alice

$\text{ID}_\text{A}$ 的 16 进制表示：416C6963 65

签名私钥 $ds_\text{A} = [t_2]P_1 = (x_{ds_\text{A}}, y_{ds_\text{A}})$

坐标 $x_{ds_\text{A}}$：A5702F05 CF131530 5E2D6EB6 4B0DEB92 3DB1A0BC F0CAFF90 523AC875 4AA69820

坐标 $y_{ds_\text{A}}$：78559A84 4411F982 5C109F5E E3F52D72 0DD01785 392A727B B1556952 B2B013D3

待签名消息 $M$：Chinese IBS standard

$M$ 的 16 进制表示：4368696E 65736520 49425320 7374616E 64617264

消息 $M$ 的签名为$(h, S)$

$h$：　823C4B21 E4BD2DFE 1ED92C60 6653E996 66856315 2FC33F55 D7BFBB9B D9705ADB

$S$：04 73BF9692 3CE58B6A D0E13E96 43A406D8 EB98417C 50EF1B29 CEF9ADB4 8B6D598C
856712F1 C2E0968A B7769F42 A99586AE D139D5B8 B3E15891 827CC2AC ED9BAA05

签名的编码如下。

30 66：　SEQUENCE{
04 20：　OCTET STRING

```
:     82 3C 4B 21 E4 BD 2D FE 1E D9 2C 60 66 53 E9 96
:     66 85 63 15 2F C3 3F 55 D7 BF BB 9B D9 70 5A DB
03 42:  BIT STRING
:     00 04 73 BF 96 92 3C E5 8B 6A D0 E1 3E 96 43 A4
:     06 D8 EB 98 41 7C 50 EF 1B 29 CE F9 AD B4 8B 6D
:     59 8C 85 67 12 F1 C2 E0 96 8A B7 76 9F 42 A9 95
:     86 AE D1 39 D5 B8 B3 E1 58 91 82 7C C2 AC ED 9B
:     AA 05
:   }
```

## A.5 加密数据格式编码测试用例

加密主私钥 $ke$：01EDEE 3778F441 F8DEA3D9 FA0ACC4E 07EE36C9 3F9A0861 8AF4AD85 CEDE1C22

加密主公钥 $P_{\text{pub-e}} = [ke]P_1 = (x_{P_{\text{pub-e}}}, y_{P_{\text{pub-e}}})$

坐标 $x_{P_{\text{pub-e}}}$：787ED7B8 A51F3AB8 4E0A6600 3F32DA5C 720B17EC A7137D39 ABC66E3C 80A892FF

坐标 $y_{P_{\text{pub-e}}}$：769DE617 91E5ADC4 B9FF85A3 1354900B 20287127 9A8C49DC 3F220F64 4C57A7B1

加密私钥生成函数识别符 hid：0x03

实体 B 的标识 $ID_B$：Bob

$ID_B$ 的 16 进制表示：426F62

用户加密私钥 $de_B=[t_2]P_2=(x_{de_B}, y_{de_B})$

坐标 $x_{de_B}$：(94736ACD 2C8C8796 CC4785E9 38301A13 9A059D35 37B64141 40B2D31E ECF41683,
115BAE85 F5D8BC6C 3DBD9E53 42979ACC CF3C2F4F 28420B1C B4F8C0B5 9A19B158)

坐标 $y_{de_B}$：(7AA5E475 70DA7600 CD760A0C F7BEAF71 C447F384 4753FE74 FA7BA92C A7D3B55F,
27538A62 E7F7BFB5 1DCE0870 4796D94C 9D56734F 119EA447 32B50E31 CDEB75C1)

待加密消息 $M$ 为：Chinese IBE standard

消息 $M$ 的 16 进制表示为：4368696E 65736520 49424520 7374616E 64617264

加密明文的方法为基于密钥派生函数的序列密码算法：

计算 $C_1$：(24454711 64490618 E1EE2052 8FF1D545 B0F14C8B CAA44544 F03DAB5D AC07D8FF,
42FFCA97 D57CDDC0 5EA405F2 E586FEB3 A6930715 532B8000 759F1305 9ED59AC0)

计算 $C_2$：1B5F5B0E 95148968 2F3E64E1 378CDD5D A9513B1C

计算 $C_3$：BA672387 BCD6DE50 16A158A5 2BB2E7FC 429197BC AB70B25A FEE37A2B 9DB9F367

密文数据结构的编码如下。

```
30 7F:  SEQUENCE{
02 01:  INTEGER
          00
03 42:  BIT STRING
:     00 04 24 45 47 11 64 49 06 18 E1 EE 20 52 8F F1
:     D5 45 B0 F1 4C 8B CA A4 45 44 F0 3D AB 5D AC 07
:     D8 FF 42 FF CA 97 D5 7C DD C0 5E A4 05 F2 E5 86
:     FE B3 A6 93 07 15 53 2B 80 00 75 9F 13 05 9E D5
:     9A C0
04 20:  OCTET STRING
:     BA 67 23 87 BC D6 DE 50 16 A1 58 A5 2B B2 E7 FC
```

```
        :     42 91 97 BC AB 70 B2 5A FE E3 7A 2B 9D B9 F3 67
    04 14:    OCTET STRING
        :     1B 5F 5B 0E 95 14 89 68 2F 3E 64 E1 37 8C DD 5D
        :     A9 51 3B 1C
        :   }
```

## A.6 密钥封装数据格式编码测试用例

加密主私钥 $ke$：01EDEE 3778F441 F8DEA3D9 FA0ACC4E 07EE36C9 3F9A0861 8AF4AD85 CEDE1C22

加密主公钥 $P_{\text{pub-e}} = [ke]P_1 = (x_{P_{\text{pub-e}}}, y_{P_{\text{pub-e}}})$

坐标 $x_{P_{\text{pub-e}}}$：787ED7B8 A51F3AB8 4E0A6600 3F32DA5C 720B17EC A7137D39 ABC66E3C 80A892FF

坐标 $y_{P_{\text{pub-e}}}$：769DE617 91E5ADC4 B9FF85A3 1354900B 20287127 9A8C49DC 3F220F64 4C57A7B1

实体 B 的标识 $ID_B$：Bob

$ID_B$ 的 16 进制表示：426F62

用户加密私钥 $de_B=(x_{de_B}, y_{de_B})$：

坐标 $x_{de_B}$：(94736ACD 2C8C8796 CC4785E9 38301A13 9A059D35 37B64141 40B2D31E ECF41683,
115BAE85 F5D8BC6C 3DBD9E53 42979ACC CF3C2F4F 28420B1C B4F8C0B5 9A19B158)

坐标 $y_{de_B}$：(7AA5E475 70DA7600 CD760A0C F7BEAF71 C447F384 4753FE74 FA7BA92C A7D3B55F,
27538A62 E7F7BFB5 1DCE0870 4796D94C 9D56734F 119EA447 32B50E31 CDEB75C1)

封装密钥的长度：0100

封装密文 $C$：（1EDEE2C3 F4659144 91DE44CE FB2CB434 AB02C308 D9DC5E20 67B4FED5 AAAC8A0F，
1C9B4C43 5ECA35AB 83BB7341 74C0F78F DE81A533 74AFF3B3 602BBC5E 37BE9A4C）

密钥 $K$：4FF5CF86 D2AD40C8 F4BAC98D 76ABDBDE 0C0E2F0A 829D3F91 1EF5B2BC E0695480

封装密文数据结构的编码如下。

```
  03 42:    BIT STRING
        :     00 04 1E DE E2 C3 F4 65 91 44 91 DE 44 CE FB 2C
        :     B4 34 AB 02 C3 08 D9 DC 5E 20 67 B4 FE D5 AA AC
        :     8A 0F 1C 9B 4C 43 5E CA 35 AB 83 BB 73 41 74 C0
        :     F7 8F DE 81 A5 33 74 AF F3 B3 60 2B BC 5E 37 BE
        :     9A 4C
```

密钥封装数据结构的编码如下。

```
  30 66:    SEQUENCE{
  04 20:    OCTET STRING
        :     4F F5 CF 86 D2 AD 40 C8 F4 BA C9 8D 76 AB DB DE
        :     0C 0E 2F 0A 82 9D 3F 91 1E F5 B2 BC E0 69 54 80
  03 42:    BIT STRING
        :     00 04 1E DE E2 C3 F4 65 91 44 91 DE 44 CE FB 2C
        :     B4 34 AB 02 C3 08 D9 DC 5E 20 67 B4 FE D5 AA AC
        :     8A 0F 1C 9B 4C 43 5E CA 35 AB 83 BB 73 41 74 C0
        :     F7 8F DE 81 A5 33 74 AF F3 B3 60 2B BC 5E 37 BE
        :     9A 4C
        :   }
```