



# 中华人民共和国密码行业标准

GM/T 0009—2023

代替 GM/T 0009—2012

## SM2 密码算法使用规范

SM2 cryptography algorithm application specification

2023-12-04 发布

2024-06-01 实施

国家密码管理局 发布



目 次

前言 ..... III

1 范围 ..... 1

2 规范性引用文件 ..... 1

3 术语和定义 ..... 1

4 缩略语 ..... 1

5 SM2 密钥对 ..... 1

6 数据转换 ..... 2

7 数据格式 ..... 3

8 预处理 ..... 4

9 计算过程 ..... 5

10 用户身份标识 ID 的默认值 ..... 7



## 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。

本文件代替 GM/T 0009—2012《SM2 密码算法使用规范》，与 GM/T 0009—2012 相比，除结构调整和编辑性改动外，主要技术变化如下：

- a) 更改了 SM2 私钥(见 5.1, 2012 年版的 5.1)；
- b) 更改了 SM2 公钥(见 5.2, 2012 年版的 5.2)；
- c) 更改了 SM2 公钥格式(见 7.1, 2012 年版的 7.1)；
- d) 更改了密钥对保护数据格式(见 7.4, 2012 年版的 7.4)；
- e) 更改了预处理 1(见 8.1, 2012 年版的 8.1)。

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：北京海泰方圆科技股份有限公司、北京信安世纪科技股份有限公司、北京小雷科技有限公司、中电科网络安全科技股份有限公司、北京国脉信安科技有限公司、无锡江南信息安全工程技术中心、兴唐通信科技有限公司、山东得安信息技术有限公司、格尔软件股份有限公司、山东大学。

本文件主要起草人：刘平、蒋红宇、柳增寿、曾宇波、袁峰、李元正、徐强、谭武征、孔凡玉、王妮娜、汪宗斌、安晓江、罗俊、徐明翼、郑强、马洪富。

本文件及其所代替文件的历次版本发布情况为：

- 2012 年首次发布为 GM/T 0009—2012；
- 本次为第一次修订。



# SM2 密码算法使用规范

## 1 范围

本文件定义了 SM2 密码算法的使用方法,也定义了相关的数据格式。  
本文件适用于 SM2 密码算法的使用,也适用于支持 SM2 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 32905—2016	信息安全技术	SM3 密码杂凑算法	
GB/T 32918.1—2016	信息安全技术	SM2 椭圆曲线公钥密码算法	第 1 部分:总则
GB/T 32918.2—2016	信息安全技术	SM2 椭圆曲线公钥密码算法	第 2 部分:数字签名算法
GB/T 32918.3—2016	信息安全技术	SM2 椭圆曲线公钥密码算法	第 3 部分:密钥交换协议
GB/T 32918.4—2016	信息安全技术	SM2 椭圆曲线公钥密码算法	第 4 部分:密钥加密算法
GB/T 32918.5—2017	信息安全技术	SM2 椭圆曲线公钥密码算法	第 5 部分:参数定义
GM/T 0006	密码应用标识规范		
GM/Z 4001	密码术语		

## 3 术语和定义

GM/Z 4001 界定的术语和定义适用于本文件。

## 4 缩略语

下列缩略语适用于本文件。  
ECB 电码本模式(Electronic Codebook)  
ECC 椭圆曲线密码算法(Elliptic Curve Cryptography)  
ID 用户身份标识(Identity)

## 5 SM2 密钥对

### 5.1 SM2 私钥

SM2 私钥是大于或等于 1 且小于  $n-1$  的整数( $n$  为 SM2 算法的阶,其值见 GB/T 32918.5—2017),简记为  $d$ 。

### 5.2 SM2 公钥

SM2 公钥是 SM2 曲线上的一个点,由横坐标和纵坐标两个分量来表示,记为  $(x,y)$ ,简记为  $Q$ 。公



钥值由其对应的私钥  $d$  与  $G$  进行点乘计算得到 ( $G$  为 SM2 椭圆曲线的基点,  $G$  的取值见 GB/T 32918.5—2017)。

## 6 数据转换

### 6.1 位串到 8 位字节串转换

位串长度若不是 8 的整数倍,应先在它的左边补 0,以保证它的长度为 8 的倍数,然后构造 8 位字节串,转换过程如下:

输入: 一个长度为  $\text{blen}$  的位串  $B$ 。

输出: 一个长度为  $\text{mlen}$  的字节串  $M$ ,其中  $\text{mlen}$  的取值为  $(\text{blen}+7)/8$  的整数部分。

动作: 将位串  $B=B_0B_1\cdots B_{\text{blen}-1}$  转换到 8 位字节串  $M=M_0M_1\cdots M_{\text{mlen}-1}$  采用如下方法:

从  $0 \leq i \leq \text{mlen}-1$ , 设置:  $M_i = B_{\text{blen}-8-8(\text{mlen}-1-i)} B_{\text{blen}-7-8(\text{mlen}-1-i)} \cdots B_{\text{blen}-1-8(\text{mlen}-1-i)}$ 。

对于  $M_0$ ,最左边  $8-\text{blen}\%8$  位设置为 0,右边设置为  $B_0B_1\cdots B_{7+\text{blen}-8\text{mlen}}$ ,其中  $\%$  表示模运算。

输出  $M$ 。

### 6.2 8 位字节串到位串转换

8 位字节串到位串转换过程如下:

输入: 一个长度为  $\text{mlen}$  的 8 位字节串  $M$ 。

输出: 一个长度为  $\text{blen}=(8 * \text{mlen})$  的位串  $B$ 。

动作: 将 8 位字节串  $M=M_0M_1\cdots M_{\text{mlen}-1}$  转换到 位串  $B=B_0B_1\cdots B_{\text{blen}-1}$  采用如下方法:

从  $0 \leq i \leq \text{mlen}-1$ , 设置:  $B_{8i}B_{8i+1}\cdots B_{8i+7} = M_i$ 。

输出  $B$ 。

### 6.3 整数到 8 位字节串转换

一个整数转换为 8 位字节串,基本方法是将其先使用二进制表达,然后把结果位串再转换为 8 位字节串。以下是转换流程:

输入: 一个非负整数  $x$ ,期望的 8 位字节串长度  $\text{mlen}$ 。基本限制为:  $2^{8(\text{mlen})} > x$ 。

输出: 一个长度为  $\text{mlen}$  的 8 位字节串  $M$ 。

动作: 将基于  $2^8=256$  的  $x$  值  $x = x_{\text{mlen}-1}2^{8(\text{mlen}-1)} + x_{\text{mlen}-2}2^{8(\text{mlen}-2)} + \cdots + x_12^8 + x_0$  转换为一个 8 位字节串  $M=M_0M_1\cdots M_{\text{mlen}-1}$  采用如下方法:

从  $0 \leq i \leq \text{mlen}-1$ , 设置:  $M_i = x_{\text{mlen}-1-i}$ 。

输出  $M$ 。

### 6.4 8 位字节串到整数的转换

可把 8 位字节串看成以 256 为基表示的整数,转换过程如下:

输入: 一个长度  $\text{mlen}$  的 8 位字节串  $M$ 。

输出: 一个整数  $x$ 。

动作: 将一个 8 位字节串  $M=M_0M_1\cdots M_{\text{mlen}-1}$  转换为整数  $x$  方法如下:

将  $M_i$  看作  $[0\sim 255]$  中的一个整数,则:  $x = \sum_{i=0}^{\text{mlen}-1} 2^{8(\text{mlen}-1-i)} M_i$ 。

输出  $x$ 。



## 7 数据格式

### 7.1 密钥数据格式

SM2 算法私钥数据格式的 ASN.1 定义为：

SM2PrivateKey ::= INTEGER

SM2 算法公钥数据格式的 ASN.1 定义为：

SM2PublicKey ::= BIT STRING

SM2PublicKey 为 BIT STRING 类型, 内容为 04 || X || Y, 02 || X 或者 03 || X, 其中 X 和 Y 均为 256 位, 分别为公钥的 x 分量和 y 分量。

04 || X || Y 为 SM2 算法公钥数据非压缩格式, 02 || X 和 03 || X 均为 SM2 算法公钥数据压缩格式。当该公钥的 y 分量的最低位为 0 时, 该公钥压缩格式为 02 || X。当该公钥的 y 分量的最低位为 1 时, 该公钥压缩格式为 03 || X。SM2 公钥格式应符合 GB/T 32918.1—2016 中 A.5 的规定。

### 7.2 加密数据格式

SM2 算法加密后的数据格式的 ASN.1 定义为：

```
SM2Cipher ::= SEQUENCE {
    xCoordinate      INTEGER,           -- x 分量
    yCoordinate      INTEGER,           -- y 分量
    hash             OCTET STRING SIZE(32), -- 杂凑值
    cipherText       OCTET STRING       -- 密文
}
```

### 7.3 签名数据格式

SM2 算法签名数据格式的 ASN.1 定义为：

```
SM2Signature ::= SEQUENCE{
    r      INTEGER,           -- 签名值的第 1 部分
    s      INTEGER           -- 签名值的第 2 部分
}
```

### 7.4 密钥对保护数据格式

在 SM2 密钥对传递时, 应对 SM2 密钥对进行加密保护。具体的保护方法如下。

- 产生一个对称密钥。
- 按对称密码算法标识指定的算法对 SM2 私钥明文进行加密, 得到私钥的密文。私钥明文为高字节在前的 8 位字节串, 其长度固定为 32 字节, 由私钥整数的二进制形式进行高比特补 0 至 256 比特后转换得到。若对称算法为分组算法, 则其运算模式为 ECB。
- 使用外部 SM2 公钥加密对称密钥得到对称密钥密文。
- 将私钥密文、对称密钥密文封装到密钥对保护数据中。

SM2 密钥对的保护数据格式的 ASN.1 定义为：

```
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  
}

其中对称密码算法标识为 SGD\_SM4\_ECB,应符合 GM/T 0006 中的规定。

## 8 预处理

### 8.1 预处理 1

预处理 1 是指使用签名方的用户身份标识和签名方公钥,通过运算得到 Z 值的过程。Z 值用于预处理 2,也用于 SM2 密钥协商协议。

输入: ID      字节串      用户身份标识  
      Q      SM2PublicKey      用户的公钥  
输出: Z      字节串      预处理 1 的输出  
计算公式为:

$$Z = \text{SM3}(L \parallel S_{\text{ID}} \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$$

式中:

Z —— 预处理 1 的输出;

L —— 用户身份标识的长度(比特);

$S_{\text{ID}}$  —— 用户身份标识;

a —— SM2 椭圆曲线参数;

b —— SM2 椭圆曲线参数;

$x_G$  —— 基点的横坐标;

$y_G$  —— 基点的纵坐标;

$x_A$  —— 用户公钥的横坐标;

$y_A$  —— 用户公钥的纵坐标。

L 由其整数值的二进制形式进行高比特补 0 至 16 比特后转换得到。 $a$ 、 $b$ 、 $x_G$ 、 $y_G$ 、 $x_A$  和  $y_A$  分别由其整数值的二进制形式进行高比特补 0 至 256 比特后转换得到。

详细的计算过程应符合 GB/T 32918.2—2016 中 5.5 和 GB/T 32905—2016 中第 5 章的规定。

### 8.2 预处理 2

预处理 2 是指使用 Z 值和待签名消息,通过 SM3 运算得到杂凑值 H 的过程。杂凑值 H 用于 SM2 数字签名。

输入: Z      字节串      预处理 1 的输出  
      M      字节串      待签名消息  
输出: H      字节串      杂凑值  
计算公式为:

$$H = \text{SM3}(Z \parallel M)$$

式中:

H —— 杂凑值;

Z —— 预处理 1 的输出;

M —— 带签名的消息。

详细的计算过程应符合 GB/T 32905—2016 中第 5 章的规定。

## 9 计算过程

### 9.1 生成密钥

SM2 密钥生成是指生成 SM2 算法的密钥对的过程,该密钥对包括私钥和与之对应的公钥。其表示方法见 7.1。

输入: 无

输出: k      SM2PrivateKey      SM2 私钥  
       Q      SM2PublicKey      SM2 公钥

详细的计算过程应符合 GB/T 32918.1—2016 中第 6 章的规定。

### 9.2 加密

SM2 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。

输入: Q      SM2PublicKey      SM2 公钥  
       m      字节串              待加密的明文数据

输出: c      SM2Cipher          密文

其中:

输出参数 c 的格式应符合 7.2 中定义;

输出参数 c 的 xCoordinate、yCoordinate 为随机产生的公钥的 x 分量和 y 分量;

输出参数 c 中的 HASH 的计算公式为:

$$\text{HASH} = \text{SM3}(Q_x \parallel m \parallel Q_y)$$

其中,  $Q_x$ 、 $Q_y$  分别为 Q 的 x 分量和 y 分量的长度为 32 字节的 8 位字节串表示;

输出参数 c 中 cipherText 为加密密文,其长度等于明文的长度。

详细的计算过程应符合 GB/T 32918.4—2016 中第 6 章的规定。

### 9.3 解密

SM2 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。

输入: d      SM2PrivateKey      SM2 私钥  
       c      SM2Cipher          密文

输出: m      字节串              与密文对应的明文

m 为 SM2Cipher 经过解密运算得到的明文,该明文的长度与输入参数 c 中 CipherText 的长度相同。

详细的计算过程应符合 GB/T 32918.4—2016 中第 7 章的规定。

### 9.4 数字签名

SM2 签名是指使用预处理 2 的结果和签名者私钥,通过签名计算得到签名结果的过程。

输入: d      SM2PrivateKey      签名者私钥  
       H      字节串              预处理 2 的结果

输出: sign    SM2Signature          签名值

详细的计算过程应符合 GB/T 32918.2—2016 中第 6 章的规定。



## 9.5 签名验证

SM2 签名验证是指使用预处理 2 的结果、签名值和签名者的公钥,通过验签计算确定签名是否通过验证的过程。

输入: H      字节串      预处理 2 的结果  
       sign    SM2Signature    签名值  
       Q      PublicKey      签名者的公钥

输出: 为“真”表示“验证通过”,为“假”表示“验证不通过”。

详细的计算过程应符合 GB/T 32918.2—2016 中第 7 章的规定。

## 9.6 密钥协商

密钥协商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。

设密钥协商双方为 A、B,其密钥对分别为 $(d_A, Q_A)$ 和 $(d_B, Q_B)$ ,双方应获得的密钥数据的比特长度为 klen。密钥协商协议分为两个阶段。

第一阶段:产生临时密钥对

用户 A:

调用生成密钥算法产生临时密钥对 $(r_A, R_A)$ ,将 $R_A$ 和用户 A 的用户身份标识 $ID_A$ 发送给用户 B。

用户 B:

调用生成密钥算法产生临时密钥对 $(r_B, R_B)$ ,将 $R_B$ 和用户 B 的用户身份标识 $ID_B$ 发送给用户 A。

第二阶段:计算共享秘密密钥

用户 A:

输入参数:

$Q_A$	SM2PublicKey	用户 A 的公钥
$Q_B$	SM2PublicKey	用户 B 的公钥
$R_A$	SM2PublicKey	用户 A 的临时公钥
$ID_A$	OCTET STRING	用户 A 的用户身份标识
$R_B$	SM2PublicKey	用户 B 的临时公钥
$ID_B$	OCTET STRING	用户 B 的用户身份标识
$d_A$	SM2PrivateKey	用户 A 的私钥
$r_A$	SM2PrivateKey	用户 A 的临时私钥
klen	INTEGER	需要输出的密钥数据的比特长度(比特)

输出参数:

K      OCTET STRING    位长为 klen 的密钥数据

步骤:

- 用 $ID_A$ 和 $Q_A$ 作为输入参数,调用预处理 1 得到 $Z_A$ ;
- 用 $ID_B$ 和 $Q_B$ 作为输入参数,调用预处理 1 得到 $Z_B$ ;
- 以 klen、 $Z_A$ 、 $Z_B$ 、 $d_A$ 、 $r_A$ 、 $R_A$ 、 $Q_B$ 、 $R_B$ 为输入参数,进行运算得到 K。

用户 B:

输入参数:

$Q_B$	SM2PublicKey	用户 B 的公钥
$Q_A$	SM2PublicKey	用户 A 的公钥
$R_B$	SM2PublicKey	用户 B 的临时公钥

ID <sub>B</sub>	OCTET STRING	用户 B 的用户身份标识
R <sub>A</sub>	SM2PublicKey	用户 A 的临时公钥
ID <sub>A</sub>	OCTET STRING	用户 A 的用户身份标识
d <sub>B</sub>	SM2PrivateKey	用户 B 的私钥
r <sub>B</sub>	SM2PrivateKey	用户 B 的临时私钥
klen	INTEGER	需要输出的密钥数据的长度(比特)

输出参数:

K	OCTET STRING	位长为 klen 的密钥数据
---	--------------	----------------

步骤:

- a) 用 ID<sub>A</sub> 和 Q<sub>A</sub> 作为输入参数,调用预处理 1 得到 Z<sub>A</sub>;
- b) 用 ID<sub>B</sub> 和 Q<sub>B</sub> 作为输入参数,调用预处理 1 得到 Z<sub>B</sub>;
- c) 以 klen、Z<sub>A</sub>、Z<sub>B</sub>、d<sub>B</sub>、r<sub>B</sub>、R<sub>B</sub>、Q<sub>A</sub>、R<sub>A</sub> 为输入参数,进行运算得到 K。

详细的计算过程应符合 GB/T 32918.3—2016 中第 6 章的规定。

10 用户身份标识 ID 的默认值

无特殊约定的情况下,用户身份标识 ID 为字节串,其长度为 16 字节,其默认值从左至右依次为 0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38。

