

SM2椭圆曲线公钥密码算法 第1部分：总则

Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves

Part 1: General

国家密码管理局

2010年12月

目 次

前 言	IV
引 言	V
1 范围	1
2 符号和缩略语	1
3 域和椭圆曲线	2
3.1 有限域	2
3.1.1 概述	2
3.1.2 素域 F_p	2
3.1.3 二元扩域 F_{2^m}	2
3.2 有限域上的椭圆曲线	2
3.2.1 F_p 上的椭圆曲线	3
3.2.2 F_{2^m} 上的椭圆曲线	3
3.2.3 椭圆曲线群	3
3.2.4 椭圆曲线多倍点运算	4
3.2.5 椭圆曲线离散对数问题(ECDLP)	4
3.2.6 弱椭圆曲线	4
4 数据类型及其转换	5
4.1 数据类型	5
4.2 数据类型转换	5
4.2.1 整数到字节串的转换	5
4.2.2 字节串到整数的转换	6
4.2.3 比特串到字节串的转换	6
4.2.4 字节串到比特串的转换	6
4.2.5 域元素到字节串的转换	6
4.2.6 字节串到域元素的转换	6
4.2.7 域元素到整数的转换	6
4.2.8 点到字节串的转换	7
4.2.9 字节串到点的转换	7
5 椭圆曲线系统参数及其验证	8
5.1 一般要求	8
5.2 F_p 上椭圆曲线系统参数及其验证	8
5.2.1 F_p 上椭圆曲线系统参数	8
5.2.2 F_p 上椭圆曲线系统参数的验证	8
5.3 F_{2^m} 上椭圆曲线系统参数及其验证	8
5.3.1 F_{2^m} 上椭圆曲线系统参数	8
5.3.2 F_{2^m} 上椭圆曲线系统参数的验证	9
6 密钥对的生成与公钥的验证	9
6.1 密钥对的生成	9
6.2 公钥的验证	9
6.2.1 F_p 上椭圆曲线公钥的验证	9
6.2.2 F_{2^m} 上椭圆曲线公钥的验证	10

附录A (资料性附录) 关于椭圆曲线的背景知识	11
A.1 素域 F_p	11
A.1.1 素域 F_p 的定义	11
A.1.2 F_p 上椭圆曲线的定义	11
A.1.3 F_p 上椭圆曲线的阶	13
A.2 二元扩域 F_{2^m}	13
A.2.1 二元扩域 F_{2^m} 的定义	13
A.2.2 F_{2^m} 上椭圆曲线的定义	19
A.2.3 F_{2^m} 上椭圆曲线的阶	22
A.3 椭圆曲线多倍点运算	22
A.3.1 概述	22
A.3.2 椭圆曲线多倍点运算的实现	22
A.3.3 椭圆曲线多倍点运算复杂度估计	23
A.4 求解椭圆曲线离散对数问题的方法	24
A.4.1 椭圆曲线离散对数求解方法	24
A.4.2 安全椭圆曲线满足的条件	25
A.5 椭圆曲线上点的压缩	26
A.5.1 概述	26
A.5.2 F_p 上椭圆曲线点的压缩与解压缩方法	26
A.5.3 F_{2^m} 上椭圆曲线点的压缩与解压缩方法	26
附录B (资料性附录) 数论算法	27
B.1 有限域和模运算	27
B.1.1 有限域中的指数运算	27
B.1.2 有限域中的逆运算	27
B.1.3 Lucas序列的生成	27
B.1.4 模素数平方根的求解	28
B.1.5 迹函数和半迹函数	28
B.1.6 F_{2^m} 上二次方程的求解	28
B.1.7 整数模素数阶的检查	29
B.1.8 整数模素数阶的计算	29
B.1.9 模素数的阶为给定值的整数的构造	30
B.1.10 概率素性检测	30
B.1.11 近似素性检测	30
B.2 有限域上的多项式	31
B.2.1 最大公因式	31
B.2.2 F_2 上不可约多项式在 F_{2^m} 中根的求解	31
B.2.3 基的转换	31
B.2.4 F_2 上多项式不可约性的检测	33
B.3 椭圆曲线算法	33
B.3.1 椭圆曲线阶的计算	33
B.3.2 椭圆曲线上点的寻找	33
附录C (资料性附录) 曲线示例	35
C.1 一般要求	35
C.2 F_p 上椭圆曲线	35

C.3	F_{2^m} 上椭圆曲线	35
附录D	(资料性附录) 椭圆曲线方程参数的拟随机生成及验证	37
D.1	椭圆曲线方程参数的拟随机生成	37
D.1.1	F_p 上椭圆曲线方程参数的拟随机生成	37
D.1.2	F_{2^m} 上椭圆曲线方程参数的拟随机生成	37
D.2	椭圆曲线方程参数的验证	37
D.2.1	F_p 上椭圆曲线方程参数的验证	38
D.2.2	F_{2^m} 上椭圆曲线方程参数的验证	38
参考文献		39

前 言

《SM2椭圆曲线公钥密码算法》分为四个部分：

- 第1部分：总则
- 第2部分：数字签名算法
- 第3部分：密钥交换协议
- 第4部分：公钥加密算法

本部分为第1部分。

本部分的附录A、附录B、附录C和附录D为资料性附录。

引 言

N.Koblitz和V.Miller在1985年各自独立地提出将椭圆曲线应用于公钥密码系统。椭圆曲线公钥密码所基于的曲线性质如下：

- 有限域上椭圆曲线在点加运算下构成有限交换群，且其阶与基域规模相近；
- 类似于有限域乘法群中的乘幂运算，椭圆曲线多倍点运算构成一个单向函数。

在多倍点运算中，已知多倍点与基点，求解倍数的问题称为椭圆曲线离散对数问题。对于一般椭圆曲线的离散对数问题，目前只存在指数级计算复杂度的求解方法。与大数分解问题及有限域上离散对数问题相比，椭圆曲线离散对数问题的求解难度要大得多。因此，在相同安全程度要求下，椭圆曲线密码较其它公钥密码所需的密钥规模要小得多。

本部分描述必要的数学基础知识与一般技术，以帮助实现其它各部分所规定的密码机制。

SM2椭圆曲线公钥密码算法

第1部分：总则

1 范围

本部分给出了SM2椭圆曲线公钥密码算法涉及的必要数学基础知识与相关密码技术，以帮助实现其它各部分所规定的密码机制。

本部分适用于基域为素域和二元扩域的椭圆曲线公钥密码算法。

2 符号和缩略语

下列符号和缩略语适用于本部分。

a, b : F_q 中的元素，它们定义 F_q 上的一条椭圆曲线 E 。

B : MOV阈。正数 B ，使得求取 F_{q^p} 上的离散对数至少与求取 F_q 上的椭圆曲线离散对数一样困难。

$\deg(f)$: 多项式 $f(x)$ 的次数。

E : 有限域上由 a 和 b 定义的一条椭圆曲线。

$E(F_q)$: F_q 上椭圆曲线 E 的所有有理点(包括无穷远点 O)组成的集合。

$ECDLP$: 椭圆曲线离散对数问题。

F_p : 包含 p 个元素的素域。

F_q : 包含 q 个元素的素域。

F_q^* : 由 F_q 中所有非零元构成的乘法群。

F_{2^m} : 包含 2^m 个元素的二元扩域。

G : 椭圆曲线的一个基点，其阶为素数。

$\gcd(x, y)$: x 和 y 的最大公因子。

h : 余因子， $h = \#E(F_q)/n$ ，其中 n 是基点 G 的阶。

$LeftRotate()$: 循环左移运算。

l_{max} : 余因子 h 的最大素因子的上界。

m : 二元扩域 F_{2^m} 关于 F_2 的扩张次数。

$\text{mod}f(x)$: 模多项式 $f(x)$ 的运算。若 $f(x)$ 是二元域上的多项式，则所有系数执行模2运算。

$\text{mod}n$: 模 n 运算。例如， $23 \text{mod} 7 = 2$ 。

n : 基点 G 的阶(n 是 $\#E(F_q)$ 的素因子)。

O : 椭圆曲线上的一个特殊点，称为无穷远点或零点，是椭圆曲线加法群的单位元。

P : $P = (x_P, y_P)$ 是椭圆曲线上除 O 之外的一个点，其坐标 x_P, y_P 满足椭圆曲线方程。

$P_1 + P_2$: 椭圆曲线 E 上两个点 P_1 与 P_2 的和。

p : 大于3的素数。

q : 有限域 F_q 中元素的数目。

r_{min} : 基点 G 的阶 n 的下界。

$Tr()$: 迹函数。

x_P : 点 P 的 x 坐标。

$x^{-1} \text{mod} n$: 使得 $x \cdot y \equiv 1 \pmod{n}$ 成立的唯一整数 y ， $1 \leq y \leq n - 1$ ， $\gcd(x, n) = 1$ 。

$x \parallel y$: x 与 y 的拼接，其中 x 和 y 是比特串或字节串。

$x \equiv y \pmod{n}$: x 与 y 模 n 同余。亦即， $x \text{mod} n = y \text{mod} n$ 。

y_P : 点 P 的 y 坐标。

\tilde{y}_P : y_P 的点压缩表示。

Z_p : 整数模 p 的剩余类环。

$\langle G \rangle$: 基点 G 生成的循环群。

$[k]P$: 椭圆曲线上点 P 的 k 倍点, 即: $[k]P = \underbrace{P + P + \cdots + P}_{k \text{ 个}}$, 其中 k 是正整数。

$[x, y]$: 大于或等于 x 且小于或等于 y 的整数的集合。

$\lceil x \rceil$: 顶函数, 大于或等于 x 的最小整数。例如, $\lceil 7 \rceil = 7, \lceil 8.3 \rceil = 9$ 。

$\lfloor x \rfloor$: 底函数, 小于或等于 x 的最大整数。例如, $\lfloor 7 \rfloor = 7, \lfloor 8.3 \rfloor = 8$ 。

$\#E(F_q)$: $E(F_q)$ 上点的数目, 称为椭圆曲线 $E(F_q)$ 的阶。

\oplus : 长度相等的两个比特串按比特的异或运算。

3 域和椭圆曲线

3.1 有限域

3.1.1 概述

本条给出有限域 F_q 的描述及其元素的表示, q 是一个奇素数或者是2的方幂。当 q 是奇素数 p 时, 要求 $p > 2^{191}$; 当 q 是2的方幂 2^m 时, 要求 $m > 192$ 且为素数。

3.1.2 素域 F_p

当 q 是奇素数 p 时, 素域 F_p 中的元素用整数 $0, 1, 2, \dots, p-1$ 表示。

- 加法单位元是整数0;
- 乘法单位元是整数1;
- 域元素的加法是整数的模 p 加法, 即若 $a, b \in F_p$, 则 $a + b = (a + b) \bmod p$;
- 域元素的乘法是整数的模 p 乘法, 即若 $a, b \in F_p$, 则 $a \cdot b = (a \cdot b) \bmod p$ 。

3.1.3 二元扩域 F_{2^m}

当 q 是2的方幂 2^m 时, 二元扩域 F_{2^m} 可以看成 F_2 上的 m 维向量空间, 其元素可用长度为 m 的比特串表示。

F_{2^m} 中的元素有多种表示方法, 其中最常用的两种方法是多项式基(PB)表示(参见附录A.2.1.1)和正规基(NB)表示(参见附录A.2.1.3)。基的选择原则是使得 F_{2^m} 中的运算效率尽可能高。本文并不规定基的选择。下面以多项式基表示为例说明二元扩域 F_{2^m} 。

设 F_2 上 m 次不可约多项式 $f(x) = x^m + f_{m-1}x^{m-1} + \cdots + f_2x^2 + f_1x + f_0$ (其中 $f_i \in F_2, i = 0, 1, \dots, m-1$)是二元扩域 F_{2^m} 的约化多项式。 F_{2^m} 由 F_2 上所有次数低于 m 的多项式构成。多项式集合 $\{x^{m-1}, x^{m-2}, \dots, x, 1\}$ 是 F_{2^m} 在 F_2 上的一组基, 称为多项式基。 F_{2^m} 中的任意一个元素 $a(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0$ 在 F_2 上的系数恰好构成了长度为 m 的比特串, 用 $a = (a_{m-1}, a_{m-2}, \dots, a_1, a_0)$ 表示。

- 零元0用全0比特串表示;
- 乘法单位元1用比特串(00...001)表示;
- 两个域元素的加法为比特串的按比特异或运算;
- 域元素 a 和 b 的乘法定义如下: 设 a 和 b 对应的 F_2 上多项式为 $a(x)$ 和 $b(x)$, 则 $a \cdot b$ 定义为多项式 $(a(x)b(x)) \bmod f(x)$ 对应的比特串。

3.2 有限域上的椭圆曲线

有限域 F_q 上的椭圆曲线是由点组成的集合。在仿射坐标系下, 椭圆曲线上点 P (非无穷远点)的坐标表示为 $P = (x_P, y_P)$, 其中 x_P, y_P 为满足一定方程的域元素, 分别称为点 P 的 x 坐标和 y 坐标。在本文中, 称 F_q 为基域。

关于椭圆曲线更多的背景知识，参见附录A.1和A.2。

在本文中，如果不做特别说明，椭圆曲线上的点均采用仿射坐标表示。

3.2.1 F_p 上的椭圆曲线

定义在 F_p (p 是大于3的素数)上的椭圆曲线方程为：

$$y^2 = x^3 + ax + b, \quad a, b \in F_p, \text{ 且 } (4a^3 + 27b^2) \bmod p \neq 0. \quad \dots\dots\dots(1)$$

椭圆曲线 $E(F_p)$ 定义为：

$$E(F_p) = \{(x, y) | x, y \in F_p, \text{ 且满足方程(1)}\} \cup \{O\}, \text{ 其中 } O \text{ 是无穷远点。}$$

椭圆曲线 $E(F_p)$ 上的点的数目用 $\#E(F_p)$ 表示，称为椭圆曲线 $E(F_p)$ 的阶。

3.2.2 F_{2^m} 上的椭圆曲线

定义在 F_{2^m} 上的椭圆曲线方程为：

$$y^2 + xy = x^3 + ax^2 + b, \quad a, b \in F_{2^m}, \text{ 且 } b \neq 0. \quad \dots\dots\dots(2)$$

椭圆曲线 $E(F_{2^m})$ 定义为：

$$E(F_{2^m}) = \{(x, y) | x, y \in F_{2^m}, \text{ 且满足方程(2)}\} \cup \{O\}, \text{ 其中 } O \text{ 是无穷远点。}$$

椭圆曲线 $E(F_{2^m})$ 上的点的数目用 $\#E(F_{2^m})$ 表示，称为椭圆曲线 $E(F_{2^m})$ 的阶。

3.2.3 椭圆曲线群

3.2.3.1 F_p 上的椭圆曲线群

椭圆曲线 $E(F_p)$ 上的点按照下面的加法运算规则，构成一个交换群：

- a) $O + O = O$;
- b) $\forall P = (x, y) \in E(F_p) \setminus \{O\}, P + O = O + P = P$;
- c) $\forall P = (x, y) \in E(F_p) \setminus \{O\}, P$ 的逆元素 $-P = (x, -y), P + (-P) = O$;
- d) 两个非互逆的不同点相加的规则：

设 $P_1 = (x_1, y_1) \in E(F_p) \setminus \{O\}, P_2 = (x_2, y_2) \in E(F_p) \setminus \{O\},$ 且 $x_1 \neq x_2,$
 设 $P_3 = (x_3, y_3) = P_1 + P_2,$ 则

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = \lambda(x_1 - x_3) - y_1, \end{cases}$$

其中

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1};$$

- e) 倍点规则：

设 $P_1 = (x_1, y_1) \in E(F_p) \setminus \{O\},$ 且 $y_1 \neq 0, P_3 = (x_3, y_3) = P_1 + P_1,$ 则

$$\begin{cases} x_3 = \lambda^2 - 2x_1, \\ y_3 = \lambda(x_1 - x_3) - y_1, \end{cases}$$

其中

$$\lambda = \frac{3x_1^2 + a}{2y_1}.$$

3.2.3.2 F_{2^m} 上的椭圆曲线群

椭圆曲线 $E(F_{2^m})$ 上的点按照下面的加法运算规则，构成一个交换群：

- a) $O + O = O$;
- b) $\forall P = (x, y) \in E(F_{2^m}) \setminus \{O\}, P + O = O + P = P$;
- c) $\forall P = (x, y) \in E(F_{2^m}) \setminus \{O\}, P$ 的逆元素 $-P = (x, x+y), P + (-P) = O$;
- d) 两个非互逆的不同点相加的规则：

设 $P_1 = (x_1, y_1) \in E(F_{2^m}) \setminus \{O\}, P_2 = (x_2, y_2) \in E(F_{2^m}) \setminus \{O\},$ 且 $x_1 \neq x_2,$
 设 $P_3 = (x_3, y_3) = P_1 + P_2,$ 则

$$\begin{cases} x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \\ y_3 = \lambda(x_1 + x_3) + x_3 + y_1, \end{cases}$$

其中

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2};$$

- e) 倍点规则：

设 $P_1 = (x_1, y_1) \in E(F_{2^m}) \setminus \{O\},$ 且 $x_1 \neq 0, P_3 = (x_3, y_3) = P_1 + P_1,$ 则

$$\begin{cases} x_3 = \lambda^2 + \lambda + a, \\ y_3 = x_1^2 + (\lambda + 1)x_3, \end{cases}$$

其中

$$\lambda = x_1 + \frac{y_1}{x_1}.$$

3.2.4 椭圆曲线多倍点运算

椭圆曲线上同一个点的多次加称为该点的多倍点运算。设 k 是一个正整数， P 是椭圆曲线上的点，称点 P 的 k 次加为点 P 的 k 倍点运算，记为 $Q = [k]P = \underbrace{P + P + \cdots + P}_{k\text{个}}$ 。因为 $[k]P = [k-1]P + P$ ，所以 k 倍点可以递归求得。

多倍点运算的输出有可能是无穷远点 O 。

多倍点运算也可以通过一些技巧更有效地实现，参见附录A.3。

3.2.5 椭圆曲线离散对数问题(ECDLP)

已知椭圆曲线 $E(F_q)$ 、阶为 n 的点 $P \in E(F_q)$ 及 $Q \in \langle P \rangle$ ，椭圆曲线离散对数问题是指确定整数 $l \in [0, n-1]$ ，使得 $Q = [l]P$ 成立。

椭圆曲线离散对数问题关系到椭圆曲线密码系统的安全，因此必须选择安全的椭圆曲线。关于如何选择安全椭圆曲线，参见附录A.4。

3.2.6 弱椭圆曲线

若某椭圆曲线存在优于 $n^{1/2}$ 级(n 是基点的阶)计算复杂度的攻击方法，则称此曲线为弱椭圆曲线。在本文中禁止使用弱椭圆曲线。

F_q 上的超奇异曲线(有限域 F_q 的特征整除 $q+1 - \#E(F_q)$)和 F_p 上的异常(Anomalous)曲线($\#E(F_q) = p$)都是弱椭圆曲线。

4 数据类型及其转换

4.1 数据类型

在本文中，数据类型包括比特串、字节串、域元素、椭圆曲线上的点和整数。

比特串：有序的0和1的序列。

字节串：有序的字节序列，其中8比特为1个字节。

域元素：有限域 F_q 中的元素。

椭圆曲线上的点：椭圆曲线上的点 $P \in E(F_q)$ ，或者是一对域元素 (x_P, y_P) ，其中域元素 x_P 和 y_P 满足椭圆曲线方程，或者是无穷远点 O 。

点的字节串表示有多种形式，用一个字节 PC 加以标识。无穷远点 O 的字节串表示是单一的零字节 $PC = 00$ 。非无穷远点 $P = (x_P, y_P)$ 有如下三种字节串表示形式：

a) 压缩表示形式， $PC = 02$ 或 03 ；

b) 未压缩表示形式， $PC = 04$ ；

c) 混合表示形式， $PC = 06$ 或 07 。

注：混合表示形式既包含压缩表示形式又包含未压缩表示形式。在实现中，它允许转换到压缩表示形式或者未压缩表示形式。

对于椭圆曲线上点的压缩表示形式和混合表示形式，本文中定为可选形式。椭圆曲线上点的压缩表示形式参见附录A.5。

4.2 数据类型转换

图1提供了各种数据类型之间的转换关系，线上的标志是相应数据转换方法所在的条。

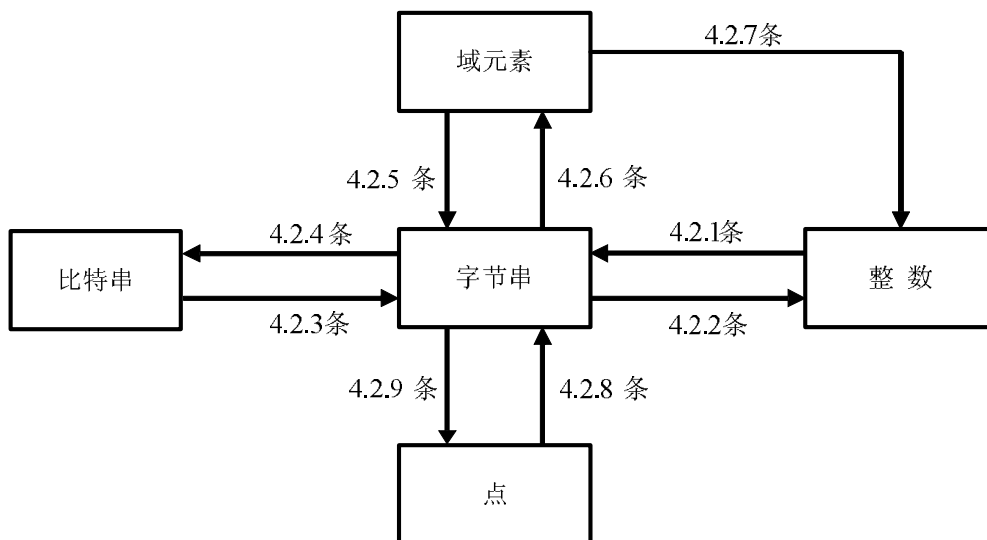


图1 数据类型和转换约定

4.2.1 整数到字节串转换

输入：非负整数 x ，以及字节串的目标长度 k （其中 k 满足 $2^{8k} > x$ ）。

输出：长度为 k 的字节串 M 。

a) 设 $M_{k-1}, M_{k-2}, \dots, M_0$ 是 M 的从最左边到最右边的字节；

b) M 的字节满足：

$$x = \sum_{i=0}^{k-1} 2^{8i} M_i。$$

4.2.2 字节串到整数的转换

输入：长度为 k 的字节串 M 。

输出：整数 x 。

- 设 $M_{k-1}, M_{k-2}, \dots, M_0$ 是 M 的从最左边到最右边的字节；
- 将 M 转换为整数 x ：

$$x = \sum_{i=0}^{k-1} 2^{8i} M_i。$$

4.2.3 比特串到字节串转换

输入：长度为 m 的比特串 s 。

输出：长度为 k 的字节串 M ，其中 $k = \lceil m/8 \rceil$ 。

- 设 $s_{m-1}, s_{m-2}, \dots, s_0$ 是 s 从最左边到最右边的比特；
- 设 $M_{k-1}, M_{k-2}, \dots, M_0$ 是 M 从最左边到最右边的字节，则

$$M_i = s_{8i+7} s_{8i+6} \cdots s_{8i+1} s_{8i}, \text{ 其中 } 0 \leq i < k, \text{ 当 } 8i+j \geq m, 0 < j \leq 7 \text{ 时, } s_{8i+j} = 0。$$

4.2.4 字节串到比特串转换

输入：长度为 k 的字节串 M 。

输出：长度为 m 的比特串 s ，其中 $m = 8k$ 。

- 设 $M_{k-1}, M_{k-2}, \dots, M_0$ 是 M 从最左边到最右边的字节；
- 设 $s_{m-1}, s_{m-2}, \dots, s_0$ 是 s 从最左边到最右边的比特，则 s_i 是 M_j 右起第 $i-8j+1$ 比特，其中 $j = \lfloor i/8 \rfloor$ 。

4.2.5 域元素到字节串转换

输入： F_q 中的元素 α 。

输出：长度 $l = \lceil t/8 \rceil$ 的字节串 S ，其中 $t = \lceil \log_2 q \rceil$ 。

- 若 q 为奇素数，则 α 必为区间 $[0, q-1]$ 中的整数，按4.2.1的细节把 α 转换成长度为 l 的字节串 S ；
- 若 $q = 2^m$ ，则 α 必为长度为 m 的比特串，按4.2.3的细节把 α 转换成长度为 l 的字节串 S ；

4.2.6 字节串到域元素的转换

输入：基域 F_q 的类型，长度为 $l = \lceil t/8 \rceil$ 的字节串 S ，其中 $t = \lceil \log_2 q \rceil$ 。

输出： F_q 中的元素 α 。

- 若 q 是奇素数，则按4.2.2的细节将 S 转换为整数 α ，若 $\alpha \notin [0, q-1]$ ，则报错；
- 若 $q = 2^m$ ，则按4.2.4的细节将 S 转换为长度为 m 的比特串 α 。

4.2.7 域元素到整数的转换

输入：域 F_q 中的元素 α 。

输出：整数 x 。

- 若 q 为奇素数，则 $x = \alpha$ (不需要转换)；
- 若 $q = 2^m$ ，则 α 必为长度为 m 的比特串，设 $s_{m-1}, s_{m-2}, \dots, s_0$ 是 α 的从最左边到最右边的比特，将 α 转化为整数 x ：

$$x = \sum_{i=0}^{m-1} 2^i s_i。$$

4.2.8 点到字节串转换

输入：椭圆曲线上的点 $P = (x_P, y_P)$ ，且 $P \neq O$ 。

输出：字节串 S 。若选用未压缩表示形式或混合表示形式，则输出字节串长度为 $2l + 1$ ；若选用压缩表示形式，则输出字节串长度为 $l + 1$ 。（ $l = \lceil (\log_2 q) / 8 \rceil$ 。）

- a) 按4.2.5中的细节把域元素 x_P 转换成长度为 l 的字节串 X_1 ；
- b) 若选用压缩表示形式，则：
 - b.1) 计算比特 \tilde{y}_P ；（参见附录A.5。）
 - b.2) 若 $\tilde{y}_P = 0$ ，则令 $PC = 02$ ；若 $\tilde{y}_P = 1$ ，则令 $PC = 03$ ；
 - b.3) 字节串 $S = PC \parallel X_1$ ；
- c) 若选用未压缩表示形式，则：
 - c.1) 按4.2.5的细节把域元素 y_P 转换成长度为 l 的字节串 Y_1 ；
 - c.2) 令 $PC = 04$ ；
 - c.3) 字节串 $S = PC \parallel X_1 \parallel Y_1$ ；
- d) 若选用混合表示形式，则：
 - d.1) 按4.2.5的细节把域元素 y_P 转换成长度为 l 的字节串 Y_1 ；
 - d.2) 计算比特 \tilde{y}_P ；（参见附录A.5。）
 - d.3) 若 $\tilde{y}_P = 0$ ，则令 $PC = 06$ ；若 $\tilde{y}_P = 1$ ，则令 $PC = 07$ ；
 - d.4) 字节串 $S = PC \parallel X_1 \parallel Y_1$ 。

4.2.9 字节串到点的转换

输入：定义 F_q 上椭圆曲线的域元素 a 、 b ，字节串 S 。若选用未压缩表示形式或混合表示形式，则字节串 S 长度为 $2l + 1$ ；若选用压缩表示形式，则字节串 S 长度为 $l + 1$ 。（ $l = \lceil (\log_2 q) / 8 \rceil$ 。）

输出：椭圆曲线上的点 $P = (x_P, y_P)$ ，且 $P \neq O$ 。

- a) 若选用压缩表示形式，则 $S = PC \parallel X_1$ ；若选用未压缩表示形式或混合表示形式，则 $S = PC \parallel X_1 \parallel Y_1$ ，其中 PC 是单一字节， X_1 和 Y_1 都是长度为 l 的字节串；
- b) 按4.2.6的细节把字节串 X_1 转换成域元素 x_P ；
- c) 若选用压缩表示形式，则：
 - c.1) 检验 $PC = 02$ 或者是 $PC = 03$ ，若不是这种情形，则报错；
 - c.2) 若 $PC = 02$ ，则令 $\tilde{y}_P = 0$ ；若 $PC = 03$ ，则令 $\tilde{y}_P = 1$ ；
 - c.3) 将 (x_P, \tilde{y}_P) 转换为椭圆曲线上的一个点 (x_P, y_P) ；（参见附录A.5。）
- d) 若选用未压缩表示形式，则：
 - d.1) 检验 $PC = 04$ ，若不是这种情形，则报错；
 - d.2) 按4.2.6的细节把字节串 Y_1 转换成域元素 y_P ；
- e) 若选用混合表示形式，则：
 - e.1) 检验 $PC = 06$ 或者 $PC = 07$ ，若不是这种情形，则报错；
 - e.2) 执行步骤e.2.1)或者e.2.2):
 - e.2.1) 按4.2.6的细节把字节串 Y_1 转换成域元素 y_P ；
 - e.2.2) 若 $PC = 06$ ，则令 $\tilde{y}_P = 0$ ，否则令 $\tilde{y}_P = 1$ ；将 (x_P, \tilde{y}_P) 转换为椭圆曲线上的一个点 (x_P, y_P) ；（参见附录A.5。）
- f) 若 q 为奇素数，则验证 $y_P^2 \equiv x_P^3 + ax_P + b \pmod{q}$ ，若不是这种情形，则报错；
若 $q = 2^m$ ，则在 F_{2^m} 中验证 $y_P^2 + x_P y_P \equiv x_P^3 + ax_P^2 + b$ ，若不是这种情形，则报错；
- g) $P = (x_P, y_P)$ 。

5 椭圆曲线系统参数及其验证

5.1 一般要求

椭圆曲线系统参数是可以公开的，系统的安全性不依赖于对这些参数的保密。本文本不规定椭圆曲线系统参数的生成方法，但规定了系统参数的验证方法。椭圆曲线阶的计算和基点的选取方法可参见附录B.3，曲线参数的生成方法可参见附录D。

椭圆曲线系统参数按照基域的不同可以分为两种情形：

- 当基域是 F_p (p 为大于3的素数)时， F_p 上的椭圆曲线系统参数；
- 当基域是 F_{2^m} 时， F_{2^m} 上的椭圆曲线系统参数。

5.2 F_p 上椭圆曲线系统参数及其验证

5.2.1 F_p 上椭圆曲线系统参数

F_p 上椭圆曲线系统参数包括：

- a) 域的规模 $q = p$ ， p 是大于3的素数；
- b) (选项)一个长度至少为192的比特串 $SEED$ (若按照附录D描述的方法产生椭圆曲线)；
- c) F_p 中的两个元素 a 和 b ，它们定义椭圆曲线 E 的方程： $y^2 = x^3 + ax + b$ ；
- d) 基点 $G = (x_G, y_G) \in E(F_p)$ ， $G \neq O$ ；
- e) 基点 G 的阶 n (要求： $n > 2^{191}$ 且 $n > 4p^{1/2}$)；
- f) (选项)余因子 $h = \#E(F_p)/n$ 。

5.2.2 F_p 上椭圆曲线系统参数的验证

下面的条件应由椭圆曲线系统参数的生成者加以验证。椭圆曲线系统参数的用户可选择验证这些条件。

输入： F_p 上椭圆曲线系统参数的集合。

输出：若椭圆曲线系统参数是有效的，则输出“有效”；否则输出“无效”。

- a) 验证 $q = p$ 是奇素数；(参见附录B.1.10。)
- b) 验证 a 、 b 、 x_G 和 y_G 是区间 $[0, p - 1]$ 中的整数；
- c) 若按照附录D描述的方法拟随机产生椭圆曲线，验证 $SEED$ 是长度至少为192的比特串，且 a 、 b 由 $SEED$ 派生得到；
- d) 验证 $(4a^3 + 27b^2) \bmod p \neq 0$ ；
- e) 验证 $y_G^2 \equiv x_G^3 + ax_G + b \pmod{p}$ ；
- f) 验证 n 是素数， $n > 2^{191}$ 且 $n > 4p^{1/2}$ ；(参见附录B.1.10。)
- g) 验证 $[n]G = O$ ；(参见附录A.3。)
- h) (选项)计算 $h' = \lfloor (p^{1/2} + 1)^2 / n \rfloor$ ，并验证 $h = h'$ ；
- i) 验证抗MOV攻击条件和抗异常曲线攻击条件成立；(参见附录A.4.2.1和A.4.2.2。)
- j) 若以上任何一个验证失败，则输出“无效”；否则，输出“有效”。

5.3 F_{2^m} 上椭圆曲线系统参数及其验证

5.3.1 F_{2^m} 上椭圆曲线系统参数

F_{2^m} 上的椭圆曲线系统参数包括：

- a) 域的规模 $q = 2^m$ ，对 F_{2^m} 中元素表示法(三项式基TPB、五项式基PPB或高斯正规基GNB)的标识，一个 F_2 上的 m 次约化多项式(若所用的基是TPB或PPB)；

- b) (选项) 一个长度至少为192的比特串 $SEED$ (若按照附录D描述的方法拟随机产生椭圆曲线);
- c) F_{2^m} 中的两个元素 a 和 b , 它们定义椭圆曲线 E 的方程: $y^2 + xy = x^3 + ax^2 + b$;
- d) 基点 $G = (x_G, y_G) \in E(F_{2^m})$, $G \neq O$;
- e) 基点 G 的阶 n (要求: $n > 2^{191}$ 且 $n > 2^{2+m/2}$);
- f) (选项)余因子 $h = \#E(F_{2^m})/n$ 。

5.3.2 F_{2^m} 上椭圆曲线系统参数的验证

下面的条件应由椭圆曲线系统参数的生成者加以验证。椭圆曲线系统参数的用户可选择验证这些条件。

输入: F_{2^m} 上椭圆曲线系统参数的集合。

输出: 若椭圆曲线系统参数是有效的, 则输出“有效”; 否则输出“无效”。

- a) 对某个 m , 验证 $q = 2^m$; 若所用的是TPB, 则验证约化多项式是 F_2 上的不可约三项式(参见表A.3); 若所用的是PPB, 则验证不存在 m 次不可约三项式, 且约化多项式是 F_2 上的不可约五项式(参见表A.4); 若所用的是GNB, 则验证 m 不能被8整除;
- b) 验证 a, b, x_G 和 y_G 是长度为 m 的比特串;
- c) 若按照附录D描述的方法拟随机产生椭圆曲线, 验证 $SEED$ 是长度至少为192的比特串, 且 a, b 由 $SEED$ 派生得到;
- d) 验证 $b \neq 0$;
- e) 在 F_{2^m} 中验证 $y_G^2 + x_G y_G = x_G^3 + a x_G^2 + b$;
- f) 验证 n 是一个素数, $n > 2^{191}$ 且 $n > 2^{2+m/2}$; (参见附录B.1.10。)
- g) 验证 $[n]G = O$; (参见附录A.3.2。)
- h) (选项)计算 $h' = \lfloor (2^{m/2} + 1)^2 / n \rfloor$, 验证 $h = h'$;
- i) 验证抗MOV攻击条件成立; (参见附录A.4.2.1。)
- j) 若以上任何一个验证失败, 则输出“无效”; 否则输出“有效”。

6 密钥对的生成与公钥的验证

6.1 密钥对的生成

输入: 一个有效的 $F_q(q = p$ 且 p 为大于3的素数, 或 $q = 2^m$)上椭圆曲线系统参数的集合。

输出: 与椭圆曲线系统参数相关的一个密钥对 (d, P) 。

- a) 用随机数发生器产生整数 $d \in [1, n - 2]$;
- b) G 为基点, 计算点 $P = (x_P, y_P) = [d]G$; (参见附录A.3.2。)
- c) 密钥对是 (d, P) , 其中 d 为私钥, P 为公钥。

6.2 公钥的验证

6.2.1 F_p 上椭圆曲线公钥的验证

输入: 一个有效的 $F_p(p > 3$ 且 p 为素数)上椭圆曲线系统参数集合及一个相关的公钥 P 。

输出: 对于给定的椭圆曲线系统参数, 若公钥 P 是有效的, 则输出“有效”; 否则输出“无效”。

- a) 验证 P 不是无穷远点 O ;
- b) 验证公钥 P 的坐标 x_P 和 y_P 是域 F_p 中的元素; (即验证 x_P 和 y_P 是区间 $[0, p - 1]$ 中的整数。)
- c) 验证 $y_P^2 \equiv x_P^3 + a x_P + b \pmod{p}$;
- d) 验证 $[n]P = O$;
- e) 若通过了所有验证, 则输出“有效”; 否则输出“无效”。

6.2.2 F_{2^m} 上椭圆曲线公钥的验证

输入： 一个有效的 F_{2^m} 上椭圆曲线系统参数集合及一个相关的公钥 P 。

输出： 对于给定的椭圆曲线系统参数，若公钥 P 是有效的，则输出“有效”；否则输出“无效”。

- a) 验证 P 不是无穷远点 O ;
- b) 验证公钥 P 的坐标 x_P 和 y_P 是域 F_{2^m} 中的元素；(即验证 x_P 和 y_P 是长度为 m 的比特串。)
- c) 在 F_{2^m} 中验证 $y_P^2 + x_P y_P = x_P^3 + ax_P^2 + b$;
- d) 验证 $[n]P = O$;
- e) 若通过了所有验证，则输出“有效”；否则输出“无效”。

注：公钥的验证是可选项。

附录 A (资料性附录)

关于椭圆曲线的背景知识

A.1 素域 F_p

A.1.1 素域 F_p 的定义

设 p 是一个素数, F_p 由 $\{0, 1, 2, \dots, p-1\}$ 中 p 个元素构成, 称 F_p 为素域。加法单位元是整数 0, 乘法单位元是整数 1, F_p 的元素满足如下运算法则:

— 加法: 设 $a, b \in F_p$, 则 $a + b = r$, 其中 $r = (a + b) \bmod p$, $r \in [0, p-1]$ 。

— 乘法: 设 $a, b \in F_p$, 则 $a \cdot b = s$, 其中 $s = (a \cdot b) \bmod p$, $s \in [0, p-1]$ 。

记 F_p^* 是由 F_p 中所有非零元构成的乘法群, 由于 F_p^* 是循环群, 所以在 F_p 中至少存在一个元素 g , 使得 F_p 中任一非零元都可以由 g 的一个方幂表示, 称 g 为 F_p^* 的生成元(或本原元), 即 $F_p^* = \{g^i | 0 \leq i \leq p-2\}$ 。设 $a = g^i \in F_p^*$, 其中 $0 \leq i \leq p-2$, 则 a 的乘法逆元为: $a^{-1} = g^{p-1-i}$ 。

示例1: 素域 F_2 , $F_2 = \{0, 1\}$

F_2 的加法表如表 A.1, 乘法表如表 A.2:

表 A.1

+	0	1
0	0	1
1	1	0

表 A.2

·	0	1
0	0	0
1	0	1

示例2: 素域 F_{19} , $F_{19} = \{0, 1, 2, \dots, 18\}$

F_{19} 中加法的示例: $10, 14 \in F_{19}$, $10+14=24$, $24 \bmod 19=5$, 则 $10+14=5$ 。

F_{19} 中乘法的示例: $7, 8 \in F_{19}$, $7 \times 8=56$, $56 \bmod 19=18$, 则 $7 \cdot 8=18$ 。

13 是 F_{19}^* 的一个生成元, 则 F_{19}^* 中元素可由 13 的方幂表示出来:

$13^0 = 1, 13^1 = 13, 13^2 = 17, 13^3 = 12, 13^4 = 4, 13^5 = 14, 13^6 = 11, 13^7 = 10, 13^8 = 16, 13^9 = 18,$

$13^{10} = 6, 13^{11} = 2, 13^{12} = 7, 13^{13} = 15, 13^{14} = 5, 13^{15} = 8, 13^{16} = 9, 13^{17} = 3, 13^{18} = 1$ 。

A.1.2 F_p 上椭圆曲线的定义

A.1.2.1 概述

F_p 上椭圆曲线常用的表示形式有两种: 仿射坐标表示和射影坐标表示。

A.1.2.2 仿射坐标表示

当 p 是大于 3 的素数时, F_p 上椭圆曲线方程在仿射坐标系下可以简化为 $y^2 = x^3 + ax + b$, 其中 $a, b \in F_p$, 且使得 $(4a^3 + 27b^2) \bmod p \neq 0$ 。椭圆曲线上的点集记为 $E(F_p) = \{(x, y) | x, y \in F_p \text{ 且满足曲线方程 } y^2 = x^3 + ax + b\} \cup \{O\}$, 其中 O 是椭圆曲线的无穷远点。

$E(F_p)$ 上的点按照下面的加法运算规则, 构成一个阿贝尔群:

a) $O + O = O$;

b) $\forall P = (x, y) \in E(F_p) \setminus \{O\}, P + O = O + P = P$;

c) $\forall P = (x, y) \in E(F_p) \setminus \{O\}$, P 的逆元素 $-P = (x, -y)$, $P + (-P) = O$;

d) 点 $P_1 = (x_1, y_1) \in E(F_p) \setminus \{O\}$, $P_2 = (x_2, y_2) \in E(F_p) \setminus \{O\}$, $P_3 = (x_3, y_3) = P_1 + P_2 \neq O$, 则

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = \lambda(x_1 - x_3) - y_1, \end{cases}$$

其中

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{若 } x_1 \neq x_2, \\ \frac{3x_1^2 + a}{2y_1}, & \text{若 } x_1 = x_2 \text{ 且 } P_2 \neq -P_1. \end{cases}$$

示例3: 有限域 F_{19} 上一条椭圆曲线

F_{19} 上方程: $y^2 = x^3 + x + 1$, 其中 $a = 1$, $b = 1$ 。则 F_{19} 上曲线的点为:

$(0,1), (0,18), (2,7), (2,12), (5,6), (5,13), (7,3), (7,16), (9,6), (9,13), (10,2), (10,17), (13,8), (13,11), (14,2), (14,17), (15,3), (15,16), (16,3), (16,16)$,

则 $E(F_{19})$ 有21个点(包括无穷远点 O)。

a) 取 $P_1 = (10,2)$, $P_2 = (9,6)$, 计算 $P_3 = P_1 + P_2$:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{6 - 2}{9 - 10} = \frac{4}{-1} = -4 \equiv 15 \pmod{19},$$

$$x_3 = 15^2 - 10 - 9 = 225 - 10 - 9 \equiv 16 - 10 - 9 = -3 \equiv 16 \pmod{19},$$

$$y_3 = 15 \times (10 - 16) - 2 = 15 \times (-6) - 2 \equiv 3 \pmod{19},$$

所以 $P_3 = (16,3)$ 。

b) 取 $P_1 = (10,2)$, 计算 $[2]P_1$:

$$\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{3 \times 10^2 + 1}{2 \times 2} = \frac{3 \times 5 + 1}{4} = \frac{16}{4} = 4 \pmod{19},$$

$$x_3 = 4^2 - 10 - 10 = -4 \equiv 15 \pmod{19},$$

$$y^3 = 4 \times (10 - 15) - 2 = -22 \equiv 16 \pmod{19},$$

所以 $[2]P_1 = (15,16)$ 。

A.1.2.3 射影坐标表示

A.1.2.3.1 标准射影坐标系

当 p 是大于3的素数时, F_p 上椭圆曲线方程在标准射影坐标系下可以简化为 $y^2z = x^3 + axz^2 + bz^3$, 其中 $a, b \in F_p$, 且 $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ 。椭圆曲线上的点集记为 $E(F_p) = \{(x, y, z) | x, y, z \in F_p \text{ 且满足曲线方程 } y^2z = x^3 + axz^2 + bz^3\}$ 。对于 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) , 若存在某个 $u \in F_p$ 且 $u \neq 0$, 使得: $x_1 = ux_2$, $y_1 = uy_2$, $z_1 = uz_2$, 则称这两个三元组等价, 表示同一个点。

若 $z \neq 0$, 记 $X = x/z$, $Y = y/z$, 则可从标准射影坐标表示转化为仿射坐标表示: $Y^2 = X^3 + aX + b$;

若 $z = 0$, $(0,1,0)$ 对应的仿射坐标系下的点即无穷远点 O 。

标准射影坐标系下, $E(F_p)$ 上点的加法运算定义如下:

a) $O + O = O$;

b) $\forall P = (x, y, z) \in E(F_p) \setminus \{O\}$, $P + O = O + P = P$;

c) $\forall P = (x, y, z) \in E(F_p) \setminus \{O\}$, P 的逆元素 $-P = (ux, -uy, uz)$, $u \in F_p$ 且 $u \neq 0$, $P + (-P) = O$;

d) 设点 $P_1 = (x_1, y_1, z_1) \in E(F_p) \setminus \{O\}$, $P_2 = (x_2, y_2, z_2) \in E(F_p) \setminus \{O\}$, $P_3 = P_1 + P_2 = (x_3, y_3, z_3) \neq O$, 若 $P_1 \neq P_2$, 则:

$$\lambda_1 = x_1z_2, \lambda_2 = x_2z_1, \lambda_3 = \lambda_1 - \lambda_2, \lambda_4 = y_1z_2, \lambda_5 = y_2z_1, \lambda_6 = \lambda_4 - \lambda_5,$$

$$\lambda_7 = \lambda_1 + \lambda_2, \lambda_8 = z_1z_2, \lambda_9 = \lambda_3^2, \lambda_{10} = \lambda_3\lambda_9, \lambda_{11} = \lambda_8\lambda_6^2 - \lambda_7\lambda_9, x_3 = \lambda_3\lambda_{11},$$

$$y_3 = \lambda_6(\lambda_9\lambda_1 - \lambda_{11}) - \lambda_4\lambda_{10}, z_3 = \lambda_{10}\lambda_8;$$

若 $P_1 = P_2$, 则:

$$\lambda_1 = 3x_1^2 + az_1^2, \lambda_2 = 2y_1z_1, \lambda_3 = y_1^2, \lambda_4 = \lambda_3x_1z_1, \lambda_5 = \lambda_2^2, \lambda_6 = \lambda_1^2 - 8\lambda_4, \\ x_3 = \lambda_2\lambda_6, y_3 = \lambda_1(4\lambda_4 - \lambda_6) - 2\lambda_5\lambda_3, z_3 = \lambda_2\lambda_5.$$

A.1.2.3.2 Jacobian加重射影坐标系

F_p 上椭圆曲线方程在Jacobian加重射影坐标系下可以简化为 $y^2 = x^3 + axz^4 + bz^6$ 。其中 $a, b \in F_p$, 且 $4a^3 + 27b^2 \neq 0 \pmod{p}$ 。椭圆曲线上的点集记为 $E(F_p) = \{(x, y, z) | x, y, z \in F_p \text{ 且满足曲线方程 } y^2 = x^3 + axz^4 + bz^6\}$ 。对于 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) , 若存在某个 $u \in F_p$ 且 $u \neq 0$, 使得: $x_1 = u^2x_2, y_1 = u^3y_2, z_1 = uz_2$, 则称这两个三元组等价, 表示同一个点。

若 $z \neq 0$, 记 $X = x/z^2, Y = y/z^3$, 则可从Jacobian加重射影坐标表示转化为仿射坐标表示: $Y^2 = X^3 + aX + b$;

若 $z = 0$, $(1, 1, 0)$ 对应的仿射坐标系下的点即无穷远点 O 。

Jacobian加重射影坐标系下, $E(F_p)$ 上点的加法运算定义如下:

a) $O + O = O$;

b) $\forall P = (x, y, z) \in E(F_p) \setminus \{O\}, P + O = O + P = P$;

c) $\forall P = (x, y, z) \in E(F_p) \setminus \{O\}, P$ 的逆元素 $-P = (u^2x, -u^3y, uz), u \in F_p$ 且 $u \neq 0, P + (-P) = O$;

d) 设点 $P_1 = (x_1, y_1, z_1) \in E(F_p) \setminus \{O\}, P_2 = (x_2, y_2, z_2) \in E(F_p) \setminus \{O\}, P_3 = P_1 + P_2 = (x_3, y_3, z_3) \neq O$, 若 $P_1 \neq P_2$, 则:

$$\lambda_1 = x_1z_2^2, \lambda_2 = x_2z_1^2, \lambda_3 = \lambda_1 - \lambda_2, \lambda_4 = y_1z_2^3, \lambda_5 = y_2z_1^3, \lambda_6 = \lambda_4 - \lambda_5, \lambda_7 = \lambda_1 + \lambda_2, \\ \lambda_8 = \lambda_4 + \lambda_5, x_3 = \lambda_6^2 - \lambda_7\lambda_3^2, \lambda_9 = \lambda_7\lambda_3^2 - 2x_3, y_3 = (\lambda_9\lambda_6 - \lambda_8\lambda_3^3)/2, z_3 = z_1z_2\lambda_3;$$

若 $P_1 = P_2$, 则:

$$\lambda_1 = 3x_1^2 + az_1^4, \lambda_2 = 4x_1y_1^2, \lambda_3 = 8y_1^4, x_3 = \lambda_1^2 - 2\lambda_2, y_3 = \lambda_1(\lambda_2 - x_3) - \lambda_3, z_3 = 2y_1z_1.$$

A.1.3 F_p 上椭圆曲线的阶

F_p (p 为大于3的素数)上一条椭圆曲线的阶是指点集 $E(F_p)$ 中元素的个数, 记为 $\#E(F_p)$ 。由Hasse定理知: $p + 1 - 2p^{1/2} \leq \#E(F_p) \leq p + 1 + 2p^{1/2}$ 。

在素域 F_p 上, 若一条曲线的阶 $\#E(F_p) = p + 1$, 则称此曲线为超奇异的, 否则为非超奇异的。

A.2 二元扩域 F_{2^m}

A.2.1 二元扩域 F_{2^m} 的定义

由 2^m 个元素构成的有限域 F_{2^m} 是 F_2 的 m 次扩张, 称为 m 次二元扩域。 F_{2^m} 可以看成 F_2 上维数为 m 的向量空间, 也就是说, 在 F_{2^m} 中存在 m 个元素 $\alpha_0, \alpha_1, \dots, \alpha_{m-1}$, 使得 $\forall \alpha \in F_{2^m}, \alpha$ 可以唯一表示为: $\alpha = a_0\alpha_0 + a_1\alpha_1 + \dots + a_{m-1}\alpha_{m-1}$, 其中 $a_i \in F_2$, 称 $\{\alpha_0, \alpha_1, \dots, \alpha_{m-1}\}$ 为 F_{2^m} 在 F_2 上的一组基。给定这样一组基, 就可以由向量 $(a_0, a_1, \dots, a_{m-1})$ 来表示域元素 α 。 F_{2^m} 在 F_2 上的基有多种选择, 域元素的加法在不同的基下的运算规则是一致的, 都可以通过向量按分量异或运算得到; 域元素的乘法在不同的基下有不同的运算规则(如用多项式基表示和用正规基表示时其运算规则就不一致)。

A.2.1.1 多项式基

设 F_2 上 m 次不可约多项式 $f(x) = x^m + f_{m-1}x^{m-1} + \dots + f_2x^2 + f_1x + f_0$ (其中 $f_i \in F_2, i = 0, 1, \dots, m-1$)是二元扩域 F_{2^m} 的约化多项式。 F_{2^m} 由 F_2 上所有次数低于 m 的多项式构成, 即:

$$F_{2^m} = \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \dots + a_1x + a_0 | a_i \in F_2, i = 0, 1, \dots, m-1\}.$$

多项式集合 $\{x^{m-1}, x^{m-2}, \dots, x, 1\}$ 是 F_{2^m} 作为向量空间在 F_2 上的一组基, 称为多项式基。

域元素 $a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0$ 相对多项式基可以由长度为 m 的比特串 $(a_{m-1}a_{m-2}\cdots a_1a_0)$ 来表示, 所以

$$F_{2^m} = \{(a_{m-1}a_{m-2}\cdots a_1a_0) | a_i \in F_2, i = 0, 1, \dots, m-1\}$$

乘法单位元1由(00...01)表示, 零元由(00...00)表示。域元素的加法和乘法定义如下:

— 加法运算

$\forall (a_{m-1}a_{m-2}\cdots a_1a_0), (b_{m-1}b_{m-2}\cdots b_1b_0) \in F_{2^m}$, 则 $(a_{m-1}a_{m-2}\cdots a_1a_0) + (b_{m-1}b_{m-2}\cdots b_1b_0) = (c_{m-1}c_{m-2}\cdots c_1c_0)$, 其中 $c_i = a_i \oplus b_i, i = 0, 1, \dots, m-1$, 亦即, 加法运算按分位异或运算执行。

— 乘法运算

$\forall (a_{m-1}a_{m-2}\cdots a_1a_0), (b_{m-1}b_{m-2}\cdots b_1b_0) \in F_{2^m}$, 则 $(a_{m-1}a_{m-2}\cdots a_1a_0) \cdot (b_{m-1}b_{m-2}\cdots b_1b_0) = (r_{m-1}r_{m-2}\cdots r_1r_0)$, 其中多项式 $(r_{m-1}x^{m-1} + r_{m-2}x^{m-2} + \cdots + r_1x + r_0)$ 是 $(a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0) \cdot (b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \cdots + b_1x + b_0)$ 在 F_2 上 $\text{mod} f(x)$ 的余式。

注意, F_{2^m} 恰包含 2^m 个元素。记 $F_{2^m}^*$ 是由 F_{2^m} 中所有非零元构成的乘法群, $F_{2^m}^*$ 是循环群, 在 F_{2^m} 中至少存在一个元素 g , 使得 $F_{2^m}^*$ 中任一非零元都可以由 g 的一个方幂表示, 称 g 为 $F_{2^m}^*$ 的生成元(或本原元), 即: $F_{2^m}^* = \{g^i | 0 \leq i \leq 2^m - 2\}$ 。设 $a = g^i \in F_{2^m}^*$, 其中 $0 \leq i \leq 2^m - 2$, 则 a 的乘法逆元为: $a^{-1} = g^{2^m-1-i}$ 。

示例4: 二元扩域 F_{2^5} 的多项式基表示

取 F_2 上的一个不可约多项式 $f(x) = x^5 + x^2 + 1$, 则 F_{2^5} 中的元素是:

(00000), (00001), (00010), (00011), (00100), (00101), (00110),
(00111), (01000), (01001), (01010), (01011), (01100), (01101),
(01110), (01111), (10000), (10001), (10010), (10011), (10100),
(10101), (10110), (10111), (11000), (11001), (11010), (11011),
(11100), (11101), (11110), (11111)。

加法: (11011)+(10011)=(01000)

乘法: (11011)·(10011)=(00100)

$$\begin{aligned} (x^4 + x^3 + x + 1) \cdot (x^4 + x + 1) &= x^8 + x^7 + x^4 + x^3 + x^2 + 1 \\ &= (x^5 + x^2 + 1) \cdot (x^3 + x^2 + 1) + x^2 \\ &\equiv x^2 \pmod{f(x)} \end{aligned}$$

即 x^2 是 $(x^4 + x^3 + x + 1) \cdot (x^4 + x + 1)$ 除以 $f(x)$ 的余式。

乘法单位元是(00001), $\alpha = x$ 是 $F_{2^5}^*$ 的一个生成元, 则 α 的方幂为:

$\alpha^0 = (00001), \alpha^1 = (00010), \alpha^2 = (00100), \alpha^3 = (01000), \alpha^4 = (10000), \alpha^5 = (00101),$
 $\alpha^6 = (01010), \alpha^7 = (10100), \alpha^8 = (01101), \alpha^9 = (11010), \alpha^{10} = (10001), \alpha^{11} = (00111),$
 $\alpha^{12} = (01110), \alpha^{13} = (11100), \alpha^{14} = (11101), \alpha^{15} = (11111), \alpha^{16} = (11011), \alpha^{17} = (10011),$
 $\alpha^{18} = (00011), \alpha^{19} = (00110), \alpha^{20} = (01100), \alpha^{21} = (11000), \alpha^{22} = (10101), \alpha^{23} = (01111),$
 $\alpha^{24} = (11110), \alpha^{25} = (11001), \alpha^{26} = (10111), \alpha^{27} = (01011), \alpha^{28} = (10110), \alpha^{29} = (01001),$
 $\alpha^{30} = (10010), \alpha^{31} = (00001)。$

A.2.1.2 三项式和五项式基

A.2.1.2.1 概述

三项式基(TPB)和五项式基(PPB)是特殊的多项式基。

A.2.1.2.2 三项式基

F_2 上的三项式是形如 $x^m + x^k + 1$ 的多项式, 其中 $1 \leq k \leq m-1$ 。

F_{2^m} 的一个三项式基表示是由 F_2 上一个 m 次不可约三项式决定的, 只有某些特定的 m 值存在这样的三项式。上述示例4即为 F_{2^5} 的三项式基表示。

对 $192 \leq m \leq 512$, 表A.3给出了存在 m 次不可约三项式的每一个 m 值; 并对每个这样的 m , 给出了最小的 k , 使得三项式 $x^m + x^k + 1$ 在 F_2 上是不可约的。

表A.3

m, k	m, k	m, k	m, k	m, k	m, k
193, 15	194, 87	196, 3	198, 9	199, 34	201, 14
202, 55	204, 27	207, 43	209, 6	210, 7	212, 105
214, 73	215, 23	217, 45	218, 11	220, 7	223, 33
225, 32	228, 113	231, 26	233, 74	234, 31	236, 5
238, 73	239, 36	241, 70	242, 95	244, 111	247, 82
249, 35	250, 103	252, 15	253, 46	255, 52	257, 12
258, 71	260, 15	263, 93	265, 42	266, 47	268, 25
270, 53	271, 58	273, 23	274, 67	276, 63	278, 5
279, 5	281, 93	282, 35	284, 53	286, 69	287, 71
289, 21	292, 37	294, 33	295, 48	297, 5	300, 5
302, 41	303, 1	305, 102	308, 15	310, 93	313, 79
314, 15	316, 63	318, 45	319, 36	321, 31	322, 67
324, 51	327, 34	329, 50	330, 99	332, 89	333, 2
337, 55	340, 45	342, 125	343, 75	345, 22	346, 63
348, 103	350, 53	351, 34	353, 69	354, 99	358, 57
359, 68	362, 63	364, 9	366, 29	367, 21	369, 91
370, 139	372, 111	375, 16	377, 41	378, 43	380, 47
382, 81	383, 90	385, 6	386, 83	388, 159	390, 9
391, 28	393, 7	394, 135	396, 25	399, 26	401, 152
402, 171	404, 65	406, 141	407, 71	409, 87	412, 147
414, 13	415, 102	417, 107	418, 199	420, 7	422, 149
423, 25	425, 12	426, 63	428, 105	431, 120	433, 33
436, 165	438, 65	439, 49	441, 7	444, 81	446, 105
447, 73	449, 134	450, 47	455, 38	457, 16	458, 203
460, 19	462, 73	463, 93	465, 31	468, 27	470, 9
471, 1	473, 200	474, 191	476, 9	478, 121	479, 104
481, 138	484, 105	486, 81	487, 94	489, 83	490, 219
492, 7	494, 17	495, 76	497, 78	498, 155	500, 27
503, 3	505, 156	506, 23	508, 9	510, 69	511, 10

A.2.1.2.3 五项式基

F_2 上的五项式是形如 $x^m + x^{k_3} + x^{k_2} + x^{k_1} + 1$ 的多项式，其中 $1 \leq k_1 < k_2 < k_3 \leq m-1$ 。 F_{2^m} 的五项式基表示是由 F_2 上一个 m 次不可约五项式决定的。对 $4 \leq m \leq 512$ ，均存在这样的五项式。

对 $192 \leq m \leq 512$ 且不存在不可约三项式的 m ，表A.4列出了其不可约五项式的 m 值；并对每一个这样的 m ，列出三元组 (k_1, k_2, k_3) ，满足：

- a) $x^m + x^{k_3} + x^{k_2} + x^{k_1} + 1$ 在 F_2 上不可约；
- b) k_1 尽可能地小；
- c) 对这个选定的 k_1 ， k_2 尽可能地小；
- d) 对选定的 k_1 和 k_2 ， k_3 尽可能地小。

表A.4

m (k_1, k_2, k_3)	m (k_1, k_2, k_3)	m (k_1, k_2, k_3)	m (k_1, k_2, k_3)
192 (1, 2, 7)	195 (1, 2, 37)	197 (1, 2, 21)	200 (1, 2, 81)
203 (1, 2, 45)	205 (1, 2, 21)	206 (1, 2, 63)	208 (1, 2, 83)
211 (1, 2, 165)	213 (1, 2, 62)	216 (1, 2, 107)	219 (1, 2, 65)
221 (1, 2, 18)	222 (1, 2, 73)	224 (1, 2, 159)	226 (1, 2, 30)
227 (1, 2, 21)	229 (1, 2, 21)	230 (1, 2, 13)	232 (1, 2, 23)
235 (1, 2, 45)	237 (1, 2, 104)	240 (1, 3, 49)	243 (1, 2, 17)
245 (1, 2, 37)	246 (1, 2, 11)	248 (1, 2, 243)	251 (1, 2, 45)
254 (1, 2, 7)	256 (1, 2, 155)	259 (1, 2, 254)	261 (1, 2, 74)
262 (1, 2, 207)	264 (1, 2, 169)	267 (1, 2, 29)	269 (1, 2, 117)
272 (1, 3, 56)	275 (1, 2, 28)	277 (1, 2, 33)	280 (1, 2, 113)
283 (1, 2, 200)	285 (1, 2, 77)	288 (1, 2, 191)	290 (1, 2, 70)
291 (1, 2, 76)	293 (1, 3, 154)	296 (1, 2, 123)	298 (1, 2, 78)
299 (1, 2, 21)	301 (1, 2, 26)	304 (1, 2, 11)	306 (1, 2, 106)
307 (1, 2, 93)	309 (1, 2, 26)	311 (1, 3, 155)	312 (1, 2, 83)
315 (1, 2, 142)	317 (1, 3, 68)	320 (1, 2, 7)	323 (1, 2, 21)
325 (1, 2, 53)	326 (1, 2, 67)	328 (1, 2, 51)	331 (1, 2, 134)
334 (1, 2, 5)	335 (1, 2, 250)	336 (1, 2, 77)	338 (1, 2, 112)
339 (1, 2, 26)	341 (1, 2, 57)	344 (1, 2, 7)	347 (1, 2, 96)
349 (1, 2, 186)	352 (1, 2, 263)	355 (1, 2, 138)	356 (1, 2, 69)
357 (1, 2, 28)	360 (1, 2, 49)	361 (1, 2, 44)	363 (1, 2, 38)
365 (1, 2, 109)	368 (1, 2, 85)	371 (1, 2, 156)	373 (1, 3, 172)
374 (1, 2, 109)	376 (1, 2, 77)	379 (1, 2, 222)	381 (1, 2, 5)
384 (1, 2, 299)	387 (1, 2, 146)	389 (1, 2, 159)	392 (1, 2, 145)
395 (1, 2, 333)	397 (1, 2, 125)	398 (1, 3, 23)	400 (1, 2, 245)
403 (1, 2, 80)	405 (1, 2, 38)	408 (1, 2, 323)	410 (1, 2, 16)
411 (1, 2, 50)	413 (1, 2, 33)	416 (1, 3, 76)	419 (1, 2, 129)
421 (1, 2, 81)	424 (1, 2, 177)	427 (1, 2, 245)	429 (1, 2, 14)
430 (1, 2, 263)	432 (1, 2, 103)	434 (1, 2, 64)	435 (1, 2, 166)
437 (1, 2, 6)	440 (1, 2, 37)	442 (1, 2, 32)	443 (1, 2, 57)
445 (1, 2, 225)	448 (1, 3, 83)	451 (1, 2, 33)	452 (1, 2, 10)
453 (1, 2, 88)	454 (1, 2, 195)	456 (1, 2, 275)	459 (1, 2, 332)
461 (1, 2, 247)	464 (1, 2, 310)	466 (1, 2, 78)	467 (1, 2, 210)
469 (1, 2, 149)	472 (1, 2, 33)	475 (1, 2, 68)	477 (1, 2, 121)
480 (1, 2, 149)	482 (1, 2, 13)	483 (1, 2, 352)	485 (1, 2, 70)
488 (1, 2, 123)	491 (1, 2, 270)	493 (1, 2, 171)	496 (1, 3, 52)
499 (1, 2, 174)	501 (1, 2, 332)	502 (1, 2, 99)	504 (1, 3, 148)
507 (1, 2, 26)	509 (1, 2, 94)	512 (1, 2, 51)	

A.2.1.2.4 选择多项式基的规则

F_{2^m} 的不同多项式基表示取决于约化多项式的选择:

a) 若存在 F_2 上的 m 次不可约三项式, 则约化多项式 $f(x)$ 选用不可约三项式 $x^m + x^k + 1$, 为了使实现的效果更好, k 的取值应尽可能小; (这样的多项式在附录A.2.1.2.2中的表A.3给出。)

b) 若不存在 F_2 上的 m 次不可约三项式, 则约化多项式 $f(x)$ 选用不可约五项式 $x^m + x^{k_3} + x^{k_2} + x^{k_1} + 1$, 为了使实现的效果更好, (i) k_1 应尽可能小; (ii) 对这个选定的 k_1 , k_2 应尽可能小; (iii) 对选定的 k_1 和 k_2 , k_3 应尽可能小。(这样的多项式在附录A.2.1.2.3中的表A.4给出。)

A.2.1.3 正规基

形如 $\{\beta, \beta^2, \beta^{2^2}, \dots, \beta^{2^{m-1}}\}$ 的基是 F_{2^m} 在 F_2 上的一组正规基, 其中 $\beta \in F_{2^m}$ 。这样的基总是存在的。 $\forall \alpha \in F_{2^m}$, 则 $\alpha = a_0\beta^{2^0} + a_1\beta^{2^1} + \dots + a_{m-1}\beta^{2^{m-1}}$, 其中 $a_i \in F_2$, ($i=0, 1, \dots, m-1$), 并记为 $\alpha = (a_0a_1a_2 \dots a_{m-2}a_{m-1})$, 域元素 α 由长度为 m 的比特串表示。所以 $F_{2^m} = \{(a_0a_1a_2 \dots a_{m-2}a_{m-1}) | a_i \in F_2, 0 \leq i \leq m-1\}$, 乘法单位元1由 m 个1的比特串(11...1)表示, 零元由 m 个0的比特串(00...0)表示。

注: 通过约定, 正规基表示的比特排序同多项式基表示的比特排序是不一样的。(参见附录A.2.1.1。)

在正规基表示下, F_{2^m} 中求平方运算是循环右移位运算:

$$\forall \alpha \in F_{2^m}, \alpha = a_0\beta^{2^0} + a_1\beta^{2^1} + \dots + a_{m-1}\beta^{2^{m-1}} = (a_0a_1a_2 \dots a_{m-2}a_{m-1}),$$

$$\alpha^2 = \left(\sum_{i=0}^{m-1} a_i\beta^{2^i}\right)^2 = \sum_{i=0}^{m-1} a_i^2\beta^{2^{i+1}} = \sum_{i=0}^{m-1} a_{i-1}\beta^{2^i} = (a_{m-1}a_0 \dots a_{m-2}).$$

在这种情况下, 求平方运算只是长度为 m 的比特串的循环移位, 便于在硬件上实现。

A.2.1.4 高斯正规基

由A.2.1.3可知, F_{2^m} 在 F_2 上的正规基是形式为 $N = \{\beta, \beta^2, \beta^{2^2}, \dots, \beta^{2^{m-1}}\}$ 的一组基, 其中 $\beta \in F_{2^m}$ 。正规基表示在求取元素的平方时有计算优势, 但对于一般意义下的不同元素的乘法运算不太方便。因此, 通常专用一种称为高斯正规基的基, 对这样的基, 乘法既简单又有效。

当 m 不能被8整除时 F_{2^m} 存在高斯正规基。高斯正规基的类型 T 是指在此基下度量乘法运算复杂度的一个正整数。一般情况下, 类型 T 愈小, 乘法效率愈高。对于给定的 m 和 T , 域 F_{2^m} 至多有一个类型 T 的高斯正规基。在所有正规基中, 类型1和类型2的高斯正规基有最有效的乘法运算, 因而也称它们为最优正规基。类型1的高斯正规基称为I型最优正规基, 类型2的高斯正规基称为II型最优正规基。

有限域 F_{2^m} 中的元素 a 在高斯正规基下可以由长度为 m 的比特串 $(a_{m-1}a_{m-2} \dots a_1a_0)$ 来表示。

- a) 乘法单位元1由 m 个1的比特串表示;
- b) 零元0由 m 个0的比特串表示;
- c) 两个域元素的加法由比特串对位异或运算完成;
- d) 域元素的乘法在附录A.2.1.4.3中描述。

A.2.1.4.1 选择正规基的规则

选择 F_{2^m} 存在的最小类型的高斯正规基。

表A.5列出[192, 512]中素数 m 的 F_{2^m} 上高斯正规基的类型。

表A.5

m 类型	m 类型	m 类型	m 类型	m 类型	m 类型
193 4	197 18	199 4	211 10	233 12	227 24
229 12	233 2	239 2	241 6	251 2	257 6
263 6	269 8	271 6	277 4	281 2	283 6
293 2	307 4	311 6	313 6	317 26	331 6
337 10	347 6	349 10	353 14	359 2	367 6

373 4	379 12	383 12	389 24	397 6	401 8
409 4	419 2	421 10	431 2	433 4	439 10
443 2	449 8	457 30	461 6	463 12	467 6
479 8	487 4	491 2	499 4	503 6	509 2

A.2.4.1.2 高斯正规基的检验

给定类型 T ，利用下述算法可以检验 F_{2^m} (m 大于1且不能被8整除)中类型 T 的高斯正规基的存在性。

输入：大于1且不被8整除的整数 m ，正整数 T 。

输出：若 F_{2^m} 存在一个类型 T 的高斯正规基，输出“正确”；否则输出“错误”。

- a) 计算 $p=T \cdot m+1$;
- b) 若 p 不是素数，则输出“错误”并停止;
- c) 计算2模 p 的阶 k ; (参见附录B.1.8。)
- d) 计算 $u = T \cdot m/k$;
- e) 计算 $d = \gcd(u, m)$;
- f) 若 $d = 1$,则输出“正确”；否则输出“错误”。

A.2.1.4.3 高斯正规基下的乘法算法

对于任意给定的高斯正规基，其乘法运算包含三部分：乘法预运算；给定两个元素后，其乘积的第一项 c_0 的公式；利用两个元素乘积的第一项 c_0 的公式，计算两个元素的乘积。下面对这三部分进行详细描述：

——乘法预运算：

输入：大于1的整数 m ，正整数 T ，其中在 F_{2^m} 上存在类型 T 的高斯正规基 B 。

输出：相对于 B 的序列 $f(1), f(2), \dots, f(p-1)$ 。

- a) 计算 $p=T \cdot m+1$;
- b) 产生模 p 阶为 T 的整数 u ; (参见附录B.1.9。)
- c) 计算序列 $f(1), f(2), \dots, f(p-1)$:
 - c.1) 置 $w = 1$;
 - c.2) 从 $j = 0$ 到 $T-1$ 执行:
 - c.2.1) 置 $n = w$;
 - c.2.2) 从 $i=0$ 到 $m-1$ 执行:
 - c.2.2.1) 置 $f(n) = i$;
 - c.2.2.2) 置 $n = 2n \bmod p$;
 - c.2.2.3) 置 $w = u \cdot w \bmod p$;

d) 输出序列 $f(1), f(2), \dots, f(p-1)$ 。

——给定在高斯正规基 B 表示下的两个域元素 a 和 b ，其乘积的第一项 c_0 的公式：

记 $c_0=F(a, b)$ 。

输入：大于1的整数 m ，正整数 T (其中在 F_{2^m} 上存在类型 T 的高斯正规基 B)及在高斯正规基 B 表示下的两个域元素 a 、 b 。

输出：在高斯正规基 B 表示下的两个域元素 a 、 b 乘积的第一项 c_0 的公式。

- a) 利用乘法预运算得到输出序列 $f(1), f(2), \dots, f(p-1)$;
- b) T 为偶数，则 $J = 0$ ，否则

$$J = \sum_{k=1}^m (a_{k-1} b_{m/2+k-1} + a_{m/2+k-1} b_{k-1})$$

c) 输出公式

$$c_0 = J + \sum_{k=1}^{p-2} a_{f(k+1)} b_{f(p-k)}$$

——利用域元素 a 和 b 乘积的第一项 c_0 的公式，计算域元素 a 和 b 的乘积：

对 $u = (u_0 u_1 \cdots u_{m-1})$ ， $v = (v_0 v_1 \cdots v_{m-1})$ ，设 $F(u, v)$ 是以 $c_0 = F(a, b)$ 导出的表达式。

输入：大于1的整数 m ，正整数 T (其中在 F_{2^m} 上存在类型 T 的高斯正规基 B)及在高斯正规基 B 表示下的两个域元素 a 、 b 。

输出：积 $(c_0 c_1 \cdots c_{m-1}) = (a_0 a_1 \cdots a_{m-1}) \times (b_0 b_1 \cdots b_{m-1})$ 。

a) 置 $(u_0 u_1 \cdots u_{m-1}) = (a_0 a_1 \cdots a_{m-1})$ ；

b) 置 $(v_0 v_1 \cdots v_{m-1}) = (b_0 b_1 \cdots b_{m-1})$ ；

c) 对 k 从0到 $m-1$ 执行：

c.1) 计算 $c_k = F(u, v)$ ；

c.2) 置 $u = \text{LeftRotate}(u)$ ，并置 $v = \text{LeftRotate}(v)$ ，其中 $\text{LeftRotate}()$ 表示循环左移1位运算，

即 $\text{LeftRotate}(u) = \text{LeftRotate}(u_0 u_1 \cdots u_{m-2} u_{m-1}) = (u_1 u_2 \cdots u_{m-1} u_0)$ ；

d) 输出 $c = (c_0 c_1 \cdots c_{m-1})$ 。

在示例4中，用多项式乘法和带余除法描述了 F_{2^5} ，下面的示例5用高斯正规基表示 F_{2^5} 。

示例5：二元扩域 F_{2^5} 的高斯正规基表示

F_{2^5} 域元素为比特五位组：

(00000), (00001), (00010), (00011), (00100), (00101), (00110), (00111),
 (01000), (01001), (01010), (01011), (01100), (01101), (01110), (01111),
 (10000), (10001), (10010), (10011), (10100), (10101), (10110), (10111),
 (11000), (11001), (11010), (11011), (11100), (11101), (11110), (11111)。

域加法： $(a_0 a_1 a_2 a_3 a_4) + (b_0 b_1 b_2 b_3 b_4) = (c_0 c_1 c_2 c_3 c_4)$ ，其中 $c_i = a_i \oplus b_i$ ， $0 \leq i < 5$ ，也就是域加法通过向量表示按分量异或运算来实现。

域乘法：

因为 $2 \times 5 + 1 = 11$ ，11是一素数，则 $T=2$ ，2模11的阶为10，且 $\gcd(5, (11-1)/10)=1$ ，所以 F_{2^5} 有第2类高斯正规基。10模11的阶为2，取 $u=10$ ，则可计算出 f 的值如下：

$$f(1)=0, f(2)=1, f(4)=2, f(8)=3, f(5)=4,$$

$$f(10)=0, f(9)=1, f(7)=2, f(3)=3, f(6)=4。$$

$a=(10000)$ ， $b=(11001)$ ，由 $T=2$ 为偶数，则有 $J=0$ 。

$$\text{则有 } F(a, b) = \sum_{k=1}^{11-2} a_{f(k+1)} b_{f(11-k)}$$

$$= a_0 b_1 + a_1 (b_0 + b_3) + a_2 (b_3 + b_4) + a_3 (b_1 + b_2) + a_4 (b_2 + b_4)。$$

$$c_0 = a_0 b_1 + a_1 (b_0 + b_3) + a_2 (b_3 + b_4) + a_3 (b_1 + b_2) + a_4 (b_2 + b_4),$$

$$c_1 = a_1 b_2 + a_2 (b_1 + b_4) + a_3 (b_4 + b_0) + a_4 (b_2 + b_3) + a_0 (b_3 + b_0),$$

$$c_2 = a_2 b_3 + a_3 (b_2 + b_0) + a_4 (b_0 + b_1) + a_0 (b_3 + b_4) + a_1 (b_4 + b_1),$$

$$c_3 = a_3 b_4 + a_4 (b_3 + b_1) + a_0 (b_1 + b_2) + a_1 (b_4 + b_0) + a_2 (b_0 + b_2),$$

$$c_4 = a_4 b_0 + a_0 (b_4 + b_2) + a_1 (b_2 + b_3) + a_2 (b_0 + b_1) + a_3 (b_1 + b_3)。$$

可以得出：

$$c_0 = F((10000), (11001)) = 1,$$

$$c_1 = F((00001), (10011)) = 1,$$

$$c_2 = F((00010), (00111)) = 1,$$

$$c_3 = F((00100), (01110)) = 1,$$

$$c_4 = F((01000), (11100)) = 1,$$

记 $c = (c_0 c_1 c_2 c_3 c_4)$ ，所以 $c = a \cdot b = (1111)$ 。

A.2.2 F_{2^m} 上椭圆曲线的定义

A.2.2.1 概述

F_{2^m} 上椭圆曲线常用的表示形式有两种：仿射坐标表示和射影坐标表示。

A.2.2.2 仿射坐标表示

F_{2m} 上非超奇异椭圆曲线方程在仿射坐标系下可以简化为 $y^2 + xy = x^3 + ax^2 + b$, 其中 $a, b \in F_{2m}$, 且 $b \neq 0$ 。椭圆曲线上的点集记为 $E(F_{2m}) = \{(x, y) | x, y \in F_{2m} \text{ 且满足曲线方程 } y^2 + xy = x^3 + ax^2 + b\} \cup \{O\}$, 其中 O 是椭圆曲线的无穷远点, 又称为零点。

$E(F_{2m})$ 按照下面的加法运算规则, 构成一个阿贝尔群:

a) $O + O = O$;

b) $\forall P = (x, y) \in E(F_{2m}) \setminus \{O\}, P + O = O + P = P$;

c) $\forall P = (x, y) \in E(F_{2m}) \setminus \{O\}, P$ 的逆元素 $-P = (x, x + y), P + (-P) = O$;

d) 两个非互逆的不同点相加的规则:

设 $P_1 = (x_1, y_1) \in E(F_{2m}) \setminus \{O\}, P_2 = (x_2, y_2) \in E(F_{2m}) \setminus \{O\}$, 且 $x_1 \neq x_2$,

设 $P_3 = (x_3, y_3) = P_1 + P_2$, 则:

$$\begin{cases} x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \\ y_3 = \lambda(x_1 + x_3) + x_3 + y_1, \end{cases}$$

其中 $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$;

e) 倍点规则:

设 $P_1 = (x_1, y_1) \in E(F_{2m}) \setminus \{O\}$ 且 $x_1 \neq 0, P_3 = (x_3, y_3) = P_1 + P_1$, 则:

$$\begin{cases} x_3 = \lambda^2 + \lambda + a, \\ y_3 = x_1^2 + (\lambda + 1)x_3, \end{cases}$$

其中 $\lambda = x_1 + \frac{y_1}{x_1}$ 。

下面给出 F_{2^5} 上椭圆曲线的两个示例。示例6用三项式基表示 F_{2^5} , 示例7用一种最优正规基表示 F_{2^5} 。

示例6: 用三项式基表示 F_{2^5} 的椭圆曲线

取不可约三项式 $f(x) = x^5 + x^2 + 1$, 取其中一个根 $\alpha = x$, 则 $F_{2^5}^*$ 可由 α 生成:

$$\begin{aligned} \alpha^0 &= (00001), \alpha^1 = (00010), \alpha^2 = (00100), \alpha^3 = (01000), \alpha^4 = (10000), \alpha^5 = (00101), \alpha^6 = (01010), \\ \alpha^7 &= (10100), \alpha^8 = (01101), \alpha^9 = (11010), \alpha^{10} = (10001), \alpha^{11} = (00111), \alpha^{12} = (01110), \alpha^{13} = (11100), \\ \alpha^{14} &= (11101), \alpha^{15} = (11111), \alpha^{16} = (11011), \alpha^{17} = (10011), \alpha^{18} = (00011), \alpha^{19} = (00110), \alpha^{20} = (01100), \\ \alpha^{21} &= (11000), \alpha^{22} = (10101), \alpha^{23} = (01111), \alpha^{24} = (11110), \alpha^{25} = (11001), \alpha^{26} = (10111), \alpha^{27} = (01011), \\ \alpha^{28} &= (10110), \alpha^{29} = (01001), \alpha^{30} = (10010), \alpha^{31} = \alpha^0 = (00001). \end{aligned}$$

取一条非超奇异曲线, 方程为: $y^2 + xy = x^3 + ax^2 + b$, 其中 $a=1, b=1$ 。此方程可以表示如下:

$$(00001)y^2 + (00001)xy = (00001)x^3 + (00001)x^2 + (00001)$$

这是因为乘法单位元为(00001)。

F_{2^5} 上此曲线的点为:

$$\begin{aligned} (0, 1), & (\alpha^3, \alpha^{15}), (\alpha^3, \alpha^{26}), (\alpha^6, \alpha^{21}), (\alpha^6, \alpha^{30}), (\alpha^7, \alpha^8), (\alpha^7, \alpha^{25}), \\ & (\alpha^{12}, \alpha^{11}), (\alpha^{12}, \alpha^{29}), (\alpha^{14}, \alpha^{16}), (\alpha^{14}, \alpha^{19}), (\alpha^{17}, \alpha^{13}), (\alpha^{17}, \alpha^{23}), (\alpha^{19}, \alpha^4), \\ & (\alpha^{19}, \alpha^{28}), (\alpha^{24}, \alpha^{22}), (\alpha^{24}, \alpha^{27}), (\alpha^{25}, \alpha^2), (\alpha^{25}, \alpha^{14}), (\alpha^{28}, \alpha), (\alpha^{28}, \alpha^7), \end{aligned}$$

则 $E(F_{2^5})$ 有22个点(包括无穷远点 O)。

a) 取 $P_1 = (x_1, y_1) = (\alpha^6, \alpha^{21}), P_2 = (x_2, y_2) = (\alpha^3, \alpha^{15})$, 计算 $P_3 = (x_3, y_3) = P_1 + P_2$:

$$\begin{aligned} \lambda &= \frac{y_2 + y_1}{x_2 + x_1} = \frac{\alpha^{15} + \alpha^{21}}{\alpha^3 + \alpha^6} = \frac{\alpha^{11}}{\alpha} = \alpha^{10}, \\ x_3 &= \lambda^2 + \lambda + x_1 + x_2 + a = \alpha^{20} + \alpha^{10} + \alpha^6 + \alpha^3 + 1 = \alpha^{24}, \\ y_3 &= \lambda(x_1 + x_3) + x_3 + y_1 = \alpha^{10}(\alpha^6 + \alpha^{24}) + \alpha^{24} + \alpha^{21} = \alpha^{22}, \\ P_3 &= (\alpha^{24}, \alpha^{22}). \end{aligned}$$

b) $P_1 = (\alpha^6, \alpha^{21})$, 计算 $[2]P_1 = (x_3, y_3)$:

$$\begin{aligned} \lambda &= x_1 + \frac{y_1}{x_1} = \alpha^6 + \frac{\alpha^{21}}{\alpha^6} = \alpha^{22}, \\ x_3 &= \lambda^2 + \lambda + a = \alpha^{44} + \alpha^{22} + 1 = \alpha^3, \\ y_3 &= x_1^2 + (\lambda + 1)x_3 = \alpha^{12} + (\alpha^{22} + 1)\alpha^3 = \alpha^{15}, \\ [2]P_1 &= (\alpha^3, \alpha^{15}). \end{aligned}$$

示例7: 用II型最优正规基表示 F_{2^5} 的椭圆曲线

取生成元 $\alpha = (11000)$, (11111) 是其乘法单位元, 则 $F_{2^5}^*$ 可由 α 生成:

$$\begin{aligned} \alpha^1 &= (11000), \alpha^2 = (01100), \alpha^3 = (11100), \alpha^5 = (10111), \alpha^6 = (01110), \\ \alpha^7 &= (00001), \alpha^9 = (11110), \alpha^{10} = (11011), \alpha^{11} = (10010), \alpha^{13} = (10100), \\ \alpha^{14} &= (10000), \alpha^{15} = (11010), \alpha^{17} = (11001), \alpha^{18} = (01111), \alpha^{19} = (00010), \\ \alpha^{21} &= (00101), \alpha^{22} = (01001), \alpha^{23} = (10101), \alpha^{25} = (00100), \alpha^{26} = (01010), \\ \alpha^{27} &= (01011), \alpha^{28} = (01000), \alpha^{29} = (10110), \alpha^{30} = (01101), \alpha^{31} = \alpha^0 = (11111). \end{aligned}$$

取一条非超奇异曲线, 方程为: $y^2 + xy = x^3 + 1$, 其中 $a=0$, $b=1$. 此方程可以表示如下:

$$(11111)y^2 + (11111)xy = (11111)x^3 + (11111)$$

这是因为乘法单位元为 (11111) 。

F_{2^5} 上此曲线的点为:

$$\begin{aligned} &(0, \alpha^0), (\alpha^0, \alpha^0), (\alpha^0, 0), (\alpha^3, \alpha^{17}), (\alpha^3, \alpha^{21}), (\alpha^5, \alpha^8), (\alpha^5, \alpha^{13}), (\alpha^6, \alpha^3), \\ &(\alpha^6, \alpha^{11}), (\alpha^7, \alpha^{21}), (\alpha^7, \alpha^{25}), (\alpha^9, \alpha^2), (\alpha^9, \alpha^{11}), (\alpha^{10}, \alpha^{16}), (\alpha^{10}, \alpha^{26}), (\alpha^{11}, \alpha^8), \\ &(\alpha^{11}, \alpha^{16}), (\alpha^{12}, \alpha^6), (\alpha^{12}, \alpha^{22}), (\alpha^{13}, \alpha^1), (\alpha^{13}, \alpha^2), (\alpha^{14}, \alpha^{11}), (\alpha^{14}, \alpha^{19}), (\alpha^{17}, \alpha^{24}), \\ &(\alpha^{17}, \alpha^{26}), (\alpha^{18}, \alpha^4), (\alpha^{18}, \alpha^{22}), (\alpha^{19}, \alpha^{26}), (\alpha^{19}, \alpha^{28}), (\alpha^{20}, \alpha^1), (\alpha^{20}, \alpha^{21}), (\alpha^{21}, \alpha^4), \\ &(\alpha^{21}, \alpha^8), (\alpha^{22}, \alpha^1), (\alpha^{22}, \alpha^{16}), (\alpha^{24}, \alpha^{12}), (\alpha^{24}, \alpha^{13}), (\alpha^{25}, \alpha^{13}), (\alpha^{25}, \alpha^{14}), (\alpha^{26}, \alpha^2), \\ &(\alpha^{26}, \alpha^4), (\alpha^{28}, \alpha^7), (\alpha^{28}, \alpha^{22}), \end{aligned}$$

则 $E(F_{2^5})$ 有44个点(包括无穷远点 O), 且 $E(F_{2^5})$ 是循环群。

取 $P = (\alpha^3, \alpha^{17})$, 按加法运算规则, 有:

$$\begin{aligned} [1]P &= (\alpha^3, \alpha^{17}), [2]P = (\alpha^{26}, \alpha^2), [3]P = (\alpha^{14}, \alpha^{19}), [4]P = (\alpha^9, \alpha^2), \\ [5]P &= (\alpha^{12}, \alpha^6), [6]P = (\alpha^{13}, \alpha^1), [7]P = (\alpha^6, \alpha^{11}), [8]P = (\alpha^{10}, \alpha^{26}), \\ [9]P &= (\alpha^{24}, \alpha^{12}), [10]P = (\alpha^{11}, \alpha^8), [11]P = (\alpha^0, \alpha^0), [12]P = (\alpha^{20}, \alpha^1), \\ [13]P &= (\alpha^7, \alpha^{21}), [14]P = (\alpha^{21}, \alpha^8), [15]P = (\alpha^{25}, \alpha^{14}), [16]P = (\alpha^{18}, \alpha^4), \\ [17]P &= (\alpha^{19}, \alpha^{26}), [18]P = (\alpha^{22}, \alpha^{16}), [19]P = (\alpha^{17}, \alpha^{26}), [20]P = (\alpha^5, \alpha^8), \\ [21]P &= (\alpha^{28}, \alpha^{22}), [22]P = (0, \alpha^0), [23]P = (\alpha^{28}, \alpha^7), [24]P = (\alpha^5, \alpha^{13}), \\ [25]P &= (\alpha^{17}, \alpha^{24}), [26]P = (\alpha^{22}, \alpha^1), [27]P = (\alpha^{19}, \alpha^{28}), [28]P = (\alpha^{18}, \alpha^{22}), \\ [29]P &= (\alpha^{25}, \alpha^{13}), [30]P = (\alpha^{21}, \alpha^4), [31]P = (\alpha^7, \alpha^{25}), [32]P = (\alpha^{20}, \alpha^{21}), \\ [33]P &= (\alpha^0, 0), [34]P = (\alpha^{11}, \alpha^{16}), [35]P = (\alpha^{24}, \alpha^{13}), [36]P = (\alpha^{10}, \alpha^{16}), \\ [37]P &= (\alpha^6, \alpha^3), [38]P = (\alpha^{13}, \alpha^2), [39]P = (\alpha^{12}, \alpha^{22}), [40]P = (\alpha^9, \alpha^{11}), \\ [41]P &= (\alpha^{14}, \alpha^{11}), [42]P = (\alpha^{26}, \alpha^4), [43]P = (\alpha^3, \alpha^{21}), [44]P = O. \end{aligned}$$

A.2.2.3 射影坐标表示

A.2.2.3.1 标准射影坐标系

F_{2^m} 上非超奇异椭圆曲线方程在标准射影坐标系下可以简化为 $y^2z + xyz = x^3 + ax^2z + bz^3$, 其中 $a, b \in F_{2^m}$, 且 $b \neq 0$ 。 $E(F_{2^m}) = \{(x, y, z) | x, y, z \in F_{2^m} \text{ 且满足曲线方程 } y^2z + xyz = x^3 + ax^2z + bz^3\}$ 。对于 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) , 若存在某个 $u \in F_{2^m}$ 且 $u \neq 0$, 使得: $x_1 = ux_2, y_1 = uy_2, z_1 = uz_2$, 则称这两个三元组等价, 表示同一个点。

若 $z \neq 0$, 记 $X = x/z, Y = y/z$, 则可从标准射影坐标表示转化为仿射坐标表示: $Y^2 + XY = X^3 + aX^2 + b$;

若 $z = 0$, 则 $(0,1,0)$ 对应的仿射坐标系下的点即无穷远点 O 。

标准射影坐标系下, $E(F_{2^m})$ 上点的加法运算定义如下:

椭圆曲线 $E(F_{2^m})$ 上的点按照下面的加法运算规则, 构成一个交换群:

a) $O + O = O$;

b) $\forall P = (x, y, z) \in E(F_{2^m}) \setminus \{O\}, P + O = O + P = P$;

c) $\forall P = (x, y, z) \in E(F_{2^m}) \setminus \{O\}, P$ 的逆元素 $-P = (ux, u(x+y), uz)$, $u \in F_{2^m}$ 且 $u \neq 0, P + (-P) = O$;

d) 设点 $P_1 = (x_1, y_1, z_1) \in E(F_{2^m}) \setminus \{O\}, P_2 = (x_2, y_2, z_2) \in E(F_{2^m}) \setminus \{O\}, P_3 = P_1 + P_2 = (x_3, y_3, z_3) \neq O$,

若 $P_1 \neq P_2$, 则:

$$\begin{aligned}\lambda_1 &= x_1 z_2, \lambda_2 = x_2 z_1, \lambda_3 = \lambda_1 + \lambda_2, \lambda_4 = y_1 z_2, \lambda_5 = y_2 z_1, \lambda_6 = \lambda_4 + \lambda_5, \lambda_7 = z_1 z_2, \\ \lambda_8 &= \lambda_3^2, \lambda_9 = \lambda_8 \lambda_7, \lambda_{10} = \lambda_3 \lambda_8, \lambda_{11} = \lambda_6 \lambda_7 (\lambda_6 + \lambda_3) + \lambda_{10} + a \lambda_9, x_3 = \lambda_3 \lambda_{11}, \\ y_3 &= \lambda_6 (\lambda_1 \lambda_8 + \lambda_{11}) + x_3 + \lambda_{10} \lambda_4, z_3 = \lambda_3 \lambda_9;\end{aligned}$$

若 $P_1 = P_2$, 则:

$$\begin{aligned}\lambda_1 &= x_1 z_1, \lambda_2 = x_1^2, \lambda_3 = \lambda_2 + y_1 z_1, \lambda_4 = \lambda_1^2, \lambda_5 = \lambda_3 (\lambda_1 + \lambda_3) + a \lambda_4, x_3 = \lambda_1 \lambda_5, \\ y_3 &= \lambda_2^2 \lambda_1 + \lambda_3 \lambda_5 + x_3, z_3 = \lambda_1 \lambda_4.\end{aligned}$$

A.2.2.3.2 Jacobian加重射影坐标系

F_{2^m} 上非超奇异椭圆曲线方程在Jacobian加重射影坐标系下可以简化为 $y^2 + xyz = x^3 + ax^2z^2 + bz^6$, 其中 $a, b \in F_{2^m}$, 且 $b \neq 0$. $E(F_{2^m}) = \{(x, y, z) | x, y, z \in F_{2^m} \text{ 且满足曲线方程 } y^2 + xyz = x^3 + ax^2z^2 + bz^6\}$. 对于 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) , 若存在某个 $u \in F_{2^m}$ 且 $u \neq 0$, 使得: $x_1 = u^2 x_2, y_1 = u^3 y_2, z_1 = u z_2$, 则称这两个三元组等价, 表示同一个点。

若 $z \neq 0$, 记 $X = x/z^2, Y = y/z^3$, 则可从Jacobian加重射影坐标表示转化为仿射坐标表示: $Y^2 + XY = X^3 + aX^2 + b$;

若 $z = 0$, 则 $(1, 1, 0)$ 对应的仿射坐标系下的点即无穷远点 O 。

Jacobian加重射影坐标系下, $E(F_{2^m})$ 上点的加法运算定义如下:

椭圆曲线 $E(F_{2^m})$ 上的点按照下面的加法运算规则, 构成一个交换群:

a) $O + O = O$;

b) $\forall P = (x, y, z) \in E(F_{2^m}) \setminus \{O\}, P + O = O + P = P$;

c) $\forall P = (x, y, z) \in E(F_{2^m}) \setminus \{O\}$, P 的逆元素 $-P = (u^2 x, u^2 y, u z)$, $u \in F_{2^m}$ 且 $u \neq 0$, $P + (-P) = O$;

d) 设点 $P_1 = (x_1, y_1, z_1) \in E(F_{2^m}) \setminus \{O\}$, $P_2 = (x_2, y_2, z_2) \in E(F_{2^m}) \setminus \{O\}$, $P_3 = P_1 + P_2 = (x_3, y_3, z_3) \neq O$,

若 $P_1 \neq P_2$, 则:

$$\begin{aligned}\lambda_1 &= x_1 z_2^2, \lambda_2 = x_2 z_1^2, \lambda_3 = \lambda_1 + \lambda_2, \lambda_4 = y_1 z_2^3, \lambda_5 = y_2 z_1^3, \lambda_6 = \lambda_4 + \lambda_5, \lambda_7 = z_1 \lambda_3, \\ \lambda_8 &= \lambda_6 x_2 + \lambda_7 y_2, z_3 = \lambda_7 z_2, \lambda_9 = \lambda_6 + z_3, x_3 = a z_3^3 + \lambda_6 \lambda_9 + \lambda_3^3, y_3 = \lambda_9 x_3 + \lambda_8 \lambda_7^2;\end{aligned}$$

若 $P_1 = P_2$, 则:

$$z_3 = x_1 z_1^2, x_3 = (x_1 + b z_1^2)^4, \lambda = z_3 + x_1^2 + y_1 z_1, y_3 = x_1^4 z_3 + \lambda x_3.$$

A.2.3 F_{2^m} 上椭圆曲线的阶

F_{2^m} 上的一条椭圆曲线 E 的阶是指点集 $E(F_{2^m})$ 中元素的个数, 记为 $\#E(F_{2^m})$ 。

由Hasse定理知: $2^m + 1 - 2^{1+m/2} \leq \#E(F_{2^m}) \leq 2^m + 1 + 2^{1+m/2}$ 。

A.3 椭圆曲线多倍点运算

A.3.1 概述

设 P 是椭圆曲线 E 上阶为 N 的点, k 为正整数, P 的 k 倍点为 Q , 即

$$Q = [k]P = \underbrace{P + P + \cdots + P}_{k \text{ 个}}$$

A.3.2 椭圆曲线多倍点运算的实现

椭圆曲线多倍点运算的实现有多种方法, 这里给出三种方法, 以下都假设 $1 \leq k < N$ 。

算法一: 二进制展开法

输入: 点 P , l 比特的整数 $k = \sum_{j=0}^{l-1} k_j 2^j$, $k_j \in \{0, 1\}$ 。

输出: $Q=[k]P$ 。

- a) 置 $Q = O$;
- b) j 从 $l-1$ 下降到 0 执行:
 - b.1) $Q = [2]Q$;
 - b.2) 若 $k_j = 1$, 则 $Q = Q + P$;
- c) 输出 Q 。

算法二: 加减法

输入: 点 P , l 比特的整数 $k = \sum_{j=0}^{l-1} k_j 2^j$, $k_j \in \{0, 1\}$ 。

输出: $Q=[k]P$ 。

- a) 设 $3k$ 的二进制表示是 $h_r h_{r-1} \cdots h_1 h_0$, 其中最高位 h_r 为 1;
- b) 设 k 的二进制表示是 $k_r k_{r-1} \cdots k_1 k_0$, 显然 $r = l$ 或 $l+1$;
- c) 置 $Q = P$;
- d) 对 i 从 $r-1$ 下降到 1 执行:
 - d.1) $Q = [2]Q$;
 - d.2) 若 $h_i = 1$, 且 $k_i = 0$, 则 $Q = Q + P$;
 - d.3) 若 $h_i = 0$, 且 $k_i = 1$, 则 $Q = Q - P$;
- e) 输出 Q 。

注: 减去点 (x, y) , 只要加上 $(x, -y)$ (对域 F_p), 或者 $(x, x+y)$ (对域 F_{2^m})。有多种不同的变种可以加速这一运算。

算法三: 滑动窗法

输入: 点 P , l 比特的整数 $k = \sum_{j=0}^{l-1} k_j 2^j$, $k_j \in \{0, 1\}$ 。

输出: $Q=[k]P$ 。

设窗口长度 $r > 1$ 。

预计算

- a) $P_1 = P, P_2 = [2]P$;
- b) i 从 1 到 $2^{r-1} - 1$ 计算 $P_{2i+1} = P_{2i-1} + P_2$;
- c) 置 $j = l-1, Q = O$ 。

主循环

- d) 当 $j \geq 0$ 执行:
 - d.1) 若 $k_j = 0$, 则 $Q = [2]Q, j = j-1$;
 - d.2) 否则
 - d.2.1) 令 t 是使 $j-t+1 \leq r$ 且 $k_t = 1$ 的最小整数;
 - d.2.2) $h_j = \sum_{i=0}^{j-t} k_{t+i} 2^i$;
 - d.2.3) $Q = [2^{j-t+1}]Q + P_{h_j}$;
 - d.2.4) 置 $j = t-1$;
- e) 输出 Q 。

A.3.3 椭圆曲线多倍点运算复杂度估计

不同坐标系下椭圆曲线点加运算和倍点运算的复杂度如表 A.6 和表 A.7。

表 A.6 素域上椭圆曲线加法运算复杂度

运算	坐标系		
	仿射坐标	标准射影坐标	Jacobian加重射影坐标
一般加法	11+2M+1S	13M+2S	12M+4S
倍点	11+2M+2S	8M+5S	4M+6S

表A.7 二元扩域上椭圆曲线加法运算复杂度

运算	坐标系		
	仿射坐标	标准射影坐标	Jacobian加重射影坐标
一般加法($a \neq 0$)	$1I+2M+1S$	$15M+1S$	$15M+5S$
倍点	$1I+2M+2S$	$8M+3S$	$5M+5S$

注：表中I、M和S分别表示有限域中的求逆运算、乘法运算和平方运算。

计算多倍点 $Q = [k]P$ ，设 k 的比特数为 l ， k 的Hamming重量为 W ，则算法一需要 $l-1$ 次椭圆曲线2倍点和 $W-1$ 次点加运算；算法二需要 l 次椭圆曲线2倍点和 $l/3$ 次点加运算；算法三分两部分：预计算时需要一次2倍点运算和 $2^{r-1} - 1$ 次点加运算，主循环部分需要 $l-1$ 次2倍点运算和 $l/(r+1) - 1$ 次点加运算，共需要 l 次2倍点运算和 $2^{r-1} + l/(r+1) - 2$ 次点加运算。一般有 $W \approx l/2$ ，则多倍点运算的复杂度如下：(基域为二元扩域时，假设 $a \neq 0$ ，当 $a = 0$ 时，少一次乘法运算。)

算法一：

基域为素域：

仿射坐标下的复杂度： $1.5I+3M+2.5S$

标准射影坐标下的复杂度： $14.5M+6S$

Jacobian加重射影坐标下的复杂度： $10M+8S$

基域为二元扩域：

仿射坐标下的复杂度： $1.5I+3M+2.5S$

标准射影坐标下的复杂度： $15.5M+3.5S$

Jacobian加重射影坐标下的复杂度： $12.5M+7.5S$

算法二：

基域为素域：

仿射坐标下的复杂度： $1.33I+2.67M+2.33S$

标准射影坐标下的复杂度： $12.33M+5.67S$

Jacobian加重射影坐标下的复杂度： $8M+7.33S$

基域为二元扩域：

仿射坐标下的复杂度： $1.33I+2.67M+2.33S$

标准射影坐标下的复杂度： $13M+3.33S$

Jacobian加重射影坐标下的复杂度： $10M+6.67S$

算法三：

基域为素域：

仿射坐标下的复杂度： $(l+l/(r+1)+2^{r-1}-2)(2M+I+S)+I$

标准射影坐标下的复杂度： $(l/(r+1)+2^{r-1}-2)(13M+2S)+l(8M+5S)$

Jacobian加重射影坐标下的复杂度： $(l/(r+1)+2^{r-1}-2)(12M+4S)+l(4M+6S)$

基域为二元扩域：

仿射坐标下的复杂度： $(l+l/(r+1)+2^{r-1}-2)(2M+I+S)+I$

标准射影坐标下的复杂度： $(l/(r+1)+2^{r-1}-2)(15M+1S)+l(8M+3S)$

Jacobian加重射影坐标下的复杂度： $(l/(r+1)+2^{r-1}-2)(15M+5S)+l(5M+5S)$

A.4 求解椭圆曲线离散对数问题的方法

A.4.1 椭圆曲线离散对数求解方法

已知椭圆曲线 $E(F_q)$ ，阶为 n 的点 $P \in E(F_q)$ 及 $Q \in \langle P \rangle$ ，椭圆曲线离散对数问题是指确定整数 $k \in [0, n-1]$ ，使得 $Q = [k]P$ 成立。

ECDLP现有攻击方法有:

- Pohlig-Hellman方法: 设 l 是 n 的最大素因子, 则算法复杂度为 $O(l^{1/2})$;
- BSGS方法: 时间复杂度与空间复杂度均为 $(\pi n/2)^{1/2}$;
- Pollard方法: 算法复杂度为 $(\pi n/2)^{1/2}$;
- 并行Pollard方法: 设 r 为并行处理器个数, 算法复杂度降至 $(\pi n/2)^{1/2}/r$;
- MOV-方法: 把超奇异椭圆曲线及具有相似性质的曲线的ECDLP降到 F_q 的小扩域上的离散对数问题(亚指数级计算复杂度算法);
- 异常曲线离散对数求解方法: 对异常曲线($\#E(F_p) = p$ 的曲线)的有效攻击方法(多项式级计算复杂度算法);
- GHS-方法: 利用Weil下降技术求解扩张次数为合数的二元扩域上椭圆曲线离散对数问题, 将ECDLP转化为超椭圆曲线离散对数问题, 而求解高亏格的超椭圆曲线离散对数存在亚指数级计算复杂度算法。

对于一般曲线的离散对数问题, 目前的求解方法都为指数级计算复杂度, 未发现有效的亚指数级计算复杂度的一般攻击方法; 而对于某些特殊曲线的离散对数问题, 存在多项式级计算复杂度或者亚指数级计算复杂度算法。

选择曲线时, 应避免使用易受上述方法攻击的密码学意义上的弱椭圆曲线。

A.4.2 安全椭圆曲线满足的条件

A.4.2.1 抗MOV攻击条件

A.Menezes、T.Okamoto、S.Vanstone、G.Frey和H.Rück的约化攻击将有限域 F_q 上的椭圆曲线离散对数问题约化为 F_{q^B} ($B > 1$)上的离散对数问题。这个攻击方法只有在 B 较小时是实用的, 大多数椭圆曲线不符合这种情况。抗MOV攻击条件确保一条椭圆曲线不易受此约化方法攻击。多数 F_q 上的椭圆曲线确实满足抗MOV攻击条件。

在验证抗MOV攻击条件之前, 必须选择一个MOV阈, 它是使得求取 F_{q^B} 上的离散对数问题至少与求取 F_q 上的椭圆曲线离散对数问题同样难的一个正整数 B 。对于 $q > 2^{191}$ 的标准, 要求 $B \geq 27$ 。选择 $B \geq 27$ 也排除了对超奇异椭圆曲线的选取。

下述算法用于验证椭圆曲线系统参数是否满足抗MOV攻击条件。

输入: MOV阈 B , 素数幂 q 和素数 n 。(n 是 $\#E(F_q) = p$ 的素因子, 其中 $E(F_q)$ 是 F_q 上的椭圆曲线。)

输出: 若 F_q 上包含 n 阶基点的椭圆曲线满足抗MOV攻击条件, 则输出“正确”; 否则输出“错误”。

- 置 $t = 1$;
- 对 i 从1到 B 执行:
 - 置 $t = (t \cdot q) \bmod n$;
 - 若 $t = 1$, 则输出“错误”并结束;
- 输出“正确”。

A.4.2.2 抗异常曲线攻击条件

设 $E(F_p)$ 为定义在素域 F_p 上的椭圆曲线, 若 $\#E(F_p) = p$, 则称椭圆曲线 $E(F_p)$ 为异常曲线。N.Smart、T.Satoh和K.Araki证明可在多项式时间内求解异常曲线的离散对数。抗异常曲线攻击条件为 $\#E(F_p) \neq p$, 满足此条件确保椭圆曲线不受异常曲线攻击。 F_p 上的绝大多数椭圆曲线确实满足抗异常曲线攻击条件。

下述算法用于验证椭圆曲线系统参数是否满足抗异常曲线攻击条件。

输入: F_p 上的椭圆曲线 $E(F_p)$, 阶 $N = \#E(F_p)$ 。

输出：若 $E(F_p)$ 满足抗异常曲线攻击条件，则输出消息“正确”；否则输出消息“错误”。

a) 若 $N = p$ ，则输出“错误”；否则输出“正确”。

A.4.2.3 其它条件

为避免Pohlig-Hellman方法和Pollard方法的攻击，基点的阶 n 必须是一个足够大的素数；为避免GHS方法的攻击， F_{2^m} 中的 m 应该选择素数。

A.5 椭圆曲线上点的压缩

A.5.1 概述

对于椭圆曲线 $E(F_q)$ 上的任意非无穷远点 $P=(x_P, y_P)$ ，该点能由仅存储 x -坐标 $x_P \in F_q$ 以及由 x_P 和 y_P 导出的一个特定比特简洁地表示，称为点的压缩表示。

A.5.2 F_p 上椭圆曲线点的压缩与解压缩方法

设 $P=(x_P, y_P)$ 是定义在 F_p 上椭圆曲线 $E: y^2 = x^3 + ax + b$ 上的一个点， \tilde{y}_P 为 y_P 的最右边的一个比特，则点 P 可由 x_P 和比特 \tilde{y}_P 表示。

由 x_P 和 \tilde{y}_P 恢复 y_P 的方法如下：

- a) 计算域元素 $\alpha = (x_P^3 + ax_P + b) \bmod p$;
- b) 计算 $\alpha \bmod p$ 的平方根 β (参见附录B.1.4)，若输出是“不存在平方根”，则报错；
- c) 若 β 的最右边比特等于 \tilde{y}_P ，则置 $y_P = \beta$ ；否则置 $y_P = p - \beta$ 。

A.5.3 F_{2^m} 上椭圆曲线点的压缩与解压缩方法

设 $P=(x_P, y_P)$ 是定义在 F_{2^m} 上的椭圆曲线 $E: y^2 + xy = x^3 + ax^2 + b$ 上的一个点。若 $x_P = 0$ ，则令 \tilde{y}_P 为0；若 $x_P \neq 0$ ，则令 \tilde{y}_P 为域元素 $y_P \cdot x_P^{-1}$ 的最右边一个比特。

由 x_P 和 \tilde{y}_P 恢复 y_P 的方法如下：

- a) 若 $x_P = 0$ ，则 $y_P = b^{2^{m-1}}$ (y_P 是 b 在 F_{2^m} 中的平方根)；
- b) 若 $x_P \neq 0$ ，则执行：
 - b.1) 在 F_{2^m} 中计算域元素 $\beta = x_P + a + bx_P^{-2}$ ；
 - b.2) 寻找一个域元素 z ，使得 $z^2 + z = \beta$ (参见附录B.1.6)，若输出是“解不存在”，则报错；
 - b.3) 设 \tilde{z} 为 z 的最后一个比特；
 - b.4) 若 $y_P \neq \tilde{z}$ ，则置 $z = z + 1$ ，其中1是乘法单位元；
 - b.5) 计算 $y_P = x_P \cdot z$ 。

附 录 B
(资料性附录)
数论算法

B.1 有限域和模运算

B.1.1 有限域中的指数运算

设 a 是正整数， g 是域 F_q 上的元素，指数运算是计算 g^a 的运算过程。通过以下概述的二进制方法可以有效地执行指数运算。

输入：正整数 a ，域 F_q ，域元素 g 。

输出： g^a 。

- a) 置 $e = a \bmod (q-1)$ ，若 $e = 0$ ，则输出1；
- b) 设 e 的二进制表示是 $e = e_r e_{r-1} \cdots e_1 e_0$ ，其最高位 e_r 为1；
- c) 置 $x = g$ ；
- d) 对 i 从 $r-1$ 下降到0执行：
 - d.1) 置 $x = x^2$ ；
 - d.2) 若 $e_i = 1$ ，则置 $x = g \cdot x$ ；
- e) 输出 x 。

其它加速算法参见(Brickell et al.1993)、(Knuth 1981)。

B.1.2 有限域中的逆运算

设 g 是域 F_q 上的非零元素，则逆元素 g^{-1} 是使得 $g \cdot c = 1$ 成立的域元素 c 。由于 $c = g^{q-2}$ ，因此求逆可通过指数运算实现。注意到，若 q 是素数， g 是满足 $1 \leq g \leq q-1$ 的整数，则 g^{-1} 是整数 c ， $1 \leq c \leq q-1$ ，且 $g \cdot c \equiv 1 \pmod{q}$ 。

输入：域 F_q ， F_q 中的非零元素 g 。

输出：逆元素 g^{-1} 。

- a) 计算 $c = g^{q-2}$ (参见附录B.1.1)；
- b) 输出 c 。

更为有效的方法是扩展的欧几里德算法，参见(Knuth 1981)。

B.1.3 Lucas序列的生成

令 X 和 Y 是非零整数， X 和 Y 的Lucas序列 U_k, V_k 的定义如下：

$$U_0 = 0, U_1 = 1, \quad \text{当 } k \geq 2 \text{ 时, } U_k = X \cdot U_{k-1} - Y \cdot U_{k-2};$$

$$V_0 = 2, V_1 = X, \quad \text{当 } k \geq 2 \text{ 时, } V_k = X \cdot V_{k-1} - Y \cdot V_{k-2}。$$

上述递归式适于计算 k 值较小的 U_k 和 V_k 。对大整数 k ，下面的算法可有效地计算 $U_k \bmod p$ 和 $V_k \bmod p$ 。

输入：奇素数 p ，整数 X 和 Y ，正整数 k 。

输出： $U_k \bmod p$ 和 $V_k \bmod p$ 。

- a) 置 $\Delta = X^2 - 4Y$ ；
- b) 设 k 的二进制表示是 $k = k_r k_{r-1} \cdots k_1 k_0$ ，其中最高位 k_r 为1；
- c) 置 $U = 1, V = X$ ；
- d) 对 i 从 $r-1$ 下降到0执行：
 - d.1) 置 $(U, V) = ((U \cdot V) \bmod p, ((V^2 + \Delta \cdot U^2)/2) \bmod p)$ ；

- d.2) 若 $k_i = 1$, 则置 $(U, V) = (((X \cdot U + V)/2) \bmod p, ((X \cdot V + \Delta \cdot U)/2) \bmod p)$;
e) 输出 U 和 V 。

B.1.4 模素数平方根的求解

设 p 是奇素数, g 是满足 $0 \leq g < p$ 的整数, g 的平方根(mod p)是整数 y , $0 \leq y < p$, 且 $y^2 \equiv g \pmod{p}$ 。
若 $g = 0$, 则只有一个平方根, 即 $y = 0$; 若 $g \neq 0$, 则 g 有0个或2个平方根(mod p), 若 y 是其中一个平方根, 则另一个平方根就是 $p - y$ 。

下面的算法可以确定 g 是否有平方根(mod p), 若有, 就计算其中一个根。

输入: 奇素数 p , 整数 g , $0 < g < p$ 。

输出: 若存在 g 的平方根, 则输出一个平方根(mod p), 否则输出“不存在平方根”。

算法1: 对 $p \equiv 3 \pmod{4}$, 即存在正整数 u , 使得 $p = 4u + 3$ 。

- a) 计算 $y = g^{u+1} \bmod p$ (参见附录B.1.1);
- b) 计算 $z = y^2 \bmod p$;
- c) 若 $z = g$, 则输出 y ; 否则输出“不存在平方根”。

算法2: 对 $p \equiv 5 \pmod{8}$, 即存在正整数 u , 使得 $p = 8u + 5$ 。

- a) 计算 $z = g^{2u+1} \bmod p$ (参见附录B.1.1);
- b) 若 $z \equiv 1 \pmod{p}$, 计算 $y = g^{u+1} \bmod p$, 输出 y , 终止算法;
- c) 若 $z \equiv -1 \pmod{p}$, 计算 $y = (2g \cdot (4g)^u) \bmod p$, 输出 y , 终止算法;
- d) 输出“不存在平方根”。

算法3: 对 $p \equiv 1 \pmod{8}$, 即存在正整数 u , 使得 $p = 8u + 1$ 。

- a) 置 $Y = g$;
- b) 生成随机数 X , $0 < X < p$;
- c) 计算Lucas序列元素(见附录B.1.3): $U = U_{4u+1} \bmod p$, $V = V_{4u+1} \bmod p$;
- d) 若 $V^2 \equiv 4Y \pmod{p}$, 则输出 $y = (V/2) \bmod p$, 并终止;
- e) 若 $U \bmod p \neq 1$ 且 $U \bmod p \neq p - 1$, 则输出“不存在平方根”, 并终止;
- f) 返回步骤b)。

B.1.5 迹函数和半迹函数

设 α 是 F_{2^m} 中的元素, α 的迹是: $Tr(\alpha) = \alpha + \alpha^2 + \alpha^{2^2} + \cdots + \alpha^{2^{m-1}}$ 。

F_{2^m} 中有一半元素的迹是0, 一半元素的迹是1。迹的计算方法如下:

若 F_{2^m} 中的元素用正规基表示:

设 $\alpha = (\alpha_0 \alpha_1 \cdots \alpha_{m-1})$, 则 $Tr(\alpha) = \alpha_0 \oplus \alpha_1 \oplus \cdots \oplus \alpha_{m-1}$ 。

若 F_{2^m} 中元素用多项式基表示:

- a) 置 $T = \alpha$;
- b) 对 i 从1到 $m - 1$ 执行:

b.1) $T = T^2 + \alpha$;

c) 输出 $Tr(\alpha) = T$ 。

若 m 是奇数, 则 α 的半迹是: $\alpha + \alpha^{2^2} + \alpha^{2^4} \cdots + \alpha^{2^{m-1}}$ 。

若 F_{2^m} 中元素用多项式基表示, 则半迹可通过下面的方法计算:

- a) 置 $T = \alpha$;
- b) 对 i 从1到 $(m - 1)/2$ 执行:
 - b.1) $T = T^2$;
 - b.2) $T = T^2 + \alpha$;
- c) 输出半迹 T 。

B.1.6 F_{2^m} 上二次方程的求解

设 β 是 F_{2^m} 中元素, 则方程 $z^2 + z = \beta$ 在 F_{2^m} 上有 $2-2\text{Tr}(\beta)$ 个解, 因此方程有0个或2个解。若 $\beta = 0$, 则解是0和1; 若 $\beta \neq 0$, z 是方程的解, 则 $z+1$ 也是方程的解。

给定 β , 利用下面的算法可确定解 z 是否存在, 若存在, 则算出一个解。

输入: F_{2^m} 及表示其元素的一组基, 以及元素 $\beta \neq 0$ 。

输出: 若存在解, 则输出元素 z , 使 $z^2 + z = \beta$; 否则输出“无解”。

算法1: 对正规基表示

- a) 设 $(\beta_0\beta_1 \cdots \beta_{m-1})$ 是 β 的表示;
- b) 置 $z_0 = 0$;
- c) 对 i 从1到 $m-1$ 执行:
 - c.1) $z_i = z_{i-1} \oplus \beta_i$;
- d) 置 $z = (z_0z_1 \cdots z_{m-1})$;
- e) 计算 $\gamma = z^2 + z$;
- f) 若 $\gamma = \beta$, 则输出 z ; 否则输出“无解”。

算法2: 对多项式基(m 是奇数)表示

- a) 计算 $z = \beta$ 的半迹(参见附录B.1.5);
- b) 计算 $\gamma = z^2 + z$;
- c) 若 $\gamma = \beta$, 则输出 z ; 否则输出“无解”。

算法3: 对任意基

- a) 选择 $\tau \in F_{2^m}$, 使得 $\tau + \tau^2 + \cdots + \tau^{2^{m-1}} = 1$;
- b) 置 $z = 0$, $w = \beta$;
- c) 对 i 从1到 $m-1$ 执行:
 - c.1) $z = z^2 + w^2 \cdot \tau$;
 - c.2) $w = w^2 + \beta$;
- d) 若 $w \neq 0$, 则输出“无解”, 并终止;
- e) 输出 z 。

B.1.7 整数模素数阶的检查

设 p 是一个素数, 整数 g 满足 $1 < g < p$, $g \bmod p$ 的阶是指最小正整数 k , 使得 $g^k \equiv 1 \pmod{p}$ 。以下算法测试 $g \bmod p$ 的阶是否为 k 。

输入: 素数 p , 整除 $p-1$ 的正整数 k , 整数 g 满足 $1 < g < p$ 。

输出: 若 k 是 $g \bmod p$ 的阶, 则输出为“正确”, 否则输出“错误”。

- a) 确定 k 的素因子;
- b) 若 $g^k \bmod p \neq 1$, 则输出“错误”, 终止;
- c) 对 k 的每一个素因子 l , 执行:
 - c.1) 若 $g^{k/l} \bmod p = 1$, 则输出“错误”, 终止;
- d) 输出“正确”。

B.1.8 整数模素数阶的计算

设 p 是素数, 整数 g 满足 $1 < g < p$ 。下面的算法确定 $g \bmod p$ 的阶, 此算法只在 p 较小时有效。

输入: 素数 p 和满足 $1 < g < p$ 的整数 g 。

输出: $g \bmod p$ 的阶 k 。

- a) 置 $b = g$, $j = 1$;

- b) $b = (g \cdot b) \bmod p, j = j + 1$;
- c) 若 $b > 1$, 则返回步骤b);
- d) 输出 $k = j$ 。

B.1.9 模素数的阶为给定值的整数的构造

设 p 是素数且 T 整除 $p-1$, 下面的算法可求出 F_p 中阶为 T 的元素。此算法只在 p 值较小时有效。

输入: 素数 p 和整除 $p-1$ 的整数 T 。

输出: 模 p 的阶为 T 的整数 u 。

- a) 随机生成整数 $g, 1 < g < p$;
- b) 计算 $g \bmod p$ 的阶 k (参见附录B.1.8);
- c) 若 T 不整除 k , 则返回步骤a);
- d) 输出 $u = g^{k/T} \bmod p$ 。

B.1.10 概率素性检测

u 是一个大的正整数, 下面的概率算法 (Miller-Rabin检测) 将确定 u 是素数还是合数。

输入: 一个大的奇数 u 和一个大的正整数 T 。

输出: “概率素数” 或 “合数”。

- a) 计算 v 和奇数 w , 使得 $u - 1 = 2^v \cdot w$;
- b) 对 j 从 1 到 T 执行:
 - b.1) 在区间 $[2, u-1]$ 中选取随机数 a ;
 - b.2) 置 $b = a^w \bmod u$;
 - b.3) 若 $b = 1$ 或 $u - 1$, 转到步骤b.6);
 - b.4) 对 i 从 1 到 $v-1$ 执行:
 - b.4.1) 置 $b = b^2 \bmod u$;
 - b.4.2) 若 $b = u - 1$, 转到步骤b.6);
 - b.4.3) 若 $b = 1$, 输出 “合数” 并终止;
 - b.4.4) 下一个 i ;
 - b.5) 输出 “合数”, 并终止;
 - b.6) 下一个 j ;
- c) 输出 “概率素数”。

若算法输出 “合数”, 则 u 是一个合数。若算法输出 “概率素数”, 则 u 是合数的概率小于 2^{-2T} 。这样, 通过选取足够大的 T , 误差可以忽略。

B.1.11 近似素性检测

给定一个试除的界 l_{max} , 若正整数 h 的每个素因子都不超过 l_{max} , 则称 h 为 l_{max} -光滑的。给定一个正整数 r_{min} , 若存在某个素数 $v \geq r_{min}$, 使得正整数 $u = h \cdot v$, 且整数 h 是 l_{max} -光滑的, 则称 u 为近似素数。下面的算法检查 u 的近似素性。

输入: 正整数 u, l_{max} 和 r_{min} 。

输出: 若 u 是近似素数则输出 h 和 v , 否则输出 “不是近似素数”。

- a) 置 $v = u, h = 1$;
- b) 对 l 从 2 到 l_{max} 执行:
 - b.1) 若 l 是合数, 则转到步骤b.3);
 - b.2) 当 l 整除 v 时, 循环执行:

- b.2.1) 置 $v = v/l$ 和 $h = h \cdot l$;
- b.2.2) 若 $v < r_{min}$, 则输出“不是近似素数”并终止;
- b.3) 下一个 l ;
- c) 若 v 是概率素数, 则输出 h 和 v 且终止;
- d) 输出“不是近似素数”。

B.2 有限域上的多项式

B.2.1 最大公因式

若 $f(t) \neq 0$ 和 $g(t) \neq 0$ 是系数在域 F_q 中的两个多项式, 则唯一地存在系数也在域 F_q 中的次数最高的首一多项式 $d(t)$, 它同时整除 $f(t)$ 和 $g(t)$ 。多项式 $d(t)$ 称为 $f(t)$ 和 $g(t)$ 的最大公因子, 记为 $\gcd(f(t), g(t))$ 。下面的算法(欧几里德算法)计算两个多项式的最大公因子。

输入: 有限域 F_q , F_q 上的两个非零多项式 $f(t) \neq 0, g(t) \neq 0$ 。

输出: $d(t) = \gcd(f(t), g(t))$ 。

- a) 置 $a(t) = f(t)$, $b(t) = g(t)$;
- b) 当 $b(t) \neq 0$ 时, 循环执行:
 - b.1) 置 $c(t) = a(t) \bmod b(t)$;
 - b.2) 置 $a(t) = b(t)$;
 - b.3) 置 $b(t) = c(t)$;
- c) 设 α 是 $a(t)$ 的首项系数并输出 $\alpha^{-1}a(t)$ 。

B.2.2 F_2 上不可约多项式在 F_{2^m} 中根的求解

设 $f(t)$ 是 F_2 上 m 次不可约多项式, 则 $f(t)$ 在域 F_{2^m} 中有 m 个不同的根。利用以下的算法可有效地求解一个根。

输入: F_2 上 m 次不可约多项式 $f(t)$, 域 F_{2^m} 。

输出: $f(t)$ 在 F_{2^m} 中的一个根。

- a) 置 $g(t) = f(t)$;
- b) 当 $\deg(g) > 1$ 时, 循环执行:
 - b.1) 随机选择 $u \in F_{2^m}$;
 - b.2) 置 $c(t) = ut$;
 - b.3) 对 i 从 1 到 $m-1$ 执行:
 - b.3.1) $c(t) = (c(t)^2 + ut) \bmod g(t)$;
 - b.4) 置 $h(t) = \gcd(c(t), g(t))$;
 - b.5) 若 $h(t)$ 是常数, 或者 $\deg(g) = \deg(h)$, 则返回步骤 b.1);
 - b.6) 若 $2\deg(h) > \deg(g)$, 则置 $g(t) = g(t)/h(t)$; 否则, 置 $g(t) = h(t)$;
- c) 输出 $g(0)$ 。

注: 上述多项式运算皆在 F_{2^m} 中进行。

B.2.3 基的转换

给定域 F_{2^m} , 以及在 F_2 上两组基(多项式基或正规基) B_1 和 B_2 , 下面的算法可在基 B_1 和 B_2 之间进行转换。

- a) 令 $f(t)$ 是 B_2 的域多项式, 即:
 - a.1) 若 B_2 是多项式基, 设 $f(t)$ 是 F_2 上 m 次约化多项式;
 - a.2) 若 B_2 是 I 型最优正规基, 设 $f(t) = t^m + t^{m-1} + \dots + t + 1$;

a.3) 若 B_2 是II型最优正规基, 设 $f(t) = \sum_{\substack{0 \leq j \leq m \\ m-j < m+j}} t^j$,

其中 a, b 的二进制表示为: $a = \sum u_i 2^i, b = \sum w_i 2^i$, 则 $a \prec b$ 表示对所有的 i , 都有 $u_i \leq w_i$;

a.4) 若 B_2 是类型 $T \geq 3$ 的高斯正规基, 则:

a.4.1) 置 $p = T \cdot m + 1$;

a.4.2) 生成模 p 的阶为 T 的整数 u (参见附录B.1.9);

a.4.3) 对 k 从1到 m 执行:

$$e_k = \sum_{j=0}^{T-1} \exp\left(\frac{2^k u^j \pi i}{p}\right), \text{ 其中 } i \text{ 为虚数单位};$$

a.4.4) 计算多项式

$$g(t) = \prod_{k=1}^m (t - e_k);$$

(多项式 $g(t)$ 的系数为整数。)

a.4.5) 输出 $f(t) = g(t) \bmod 2$ 。

复数 e_k 的计算必须足够精确才能与 $g(t)$ 的每一个系数保持一致。由于每一个系数都是整数, 这就意味着计算系数过程中的偏差必须小于 $1/2$ 。

b) 设 γ 是相对于 B_1 的 $f(t)$ 的根。(γ 可通过附录B.2.2中的方法计算。)

c) 令 Γ 是如下矩阵:

$$\Gamma = \begin{pmatrix} \gamma_{0,0} & \gamma_{0,1} & \cdots & \gamma_{0,m-1} \\ \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{m-1,0} & \gamma_{m-1,1} & \cdots & \gamma_{m-1,m-1} \end{pmatrix},$$

其中的项 $\gamma_{i,j}$ 定义如下:

c.1) 若 B_2 是多项式基, 则相对于 B_1 有:

$$1 = (\gamma_{0,0} \gamma_{0,1} \cdots \gamma_{0,m-1});$$

$$\gamma = (\gamma_{1,0} \gamma_{1,1} \cdots \gamma_{1,m-1}),$$

$$\gamma^2 = (\gamma_{2,0} \gamma_{2,1} \cdots \gamma_{2,m-1}),$$

.....

$$\gamma^{m-1} = (\gamma_{m-1,0} \gamma_{m-1,1} \cdots \gamma_{m-1,m-1}).$$

(通过不断地乘以 γ 得到项 $\gamma_{i,j}$ 。)

c.2) 若 B_2 是高斯正规基(类型 $T \geq 1$), 则相对于 B_1 有:

$$\gamma = (\gamma_{0,0} \gamma_{0,1} \cdots \gamma_{0,m-1}),$$

$$\gamma^2 = (\gamma_{1,0} \gamma_{1,1} \cdots \gamma_{1,m-1}),$$

$$\gamma^4 = (\gamma_{2,0} \gamma_{2,1} \cdots \gamma_{2,m-1}),$$

.....

$$\gamma^{2^{m-1}} = (\gamma_{m-1,0} \gamma_{m-1,1} \cdots \gamma_{m-1,m-1}).$$

(通过不断求 γ 的平方得到项 $\gamma_{i,j}$ 。)

d) 若一个元素相对于 B_2 的表示是 $(\beta_0 \beta_1 \cdots \beta_{m-1})$, 则它相对于 B_1 的表示为:

$$(\alpha_0 \alpha_1 \cdots \alpha_{m-1}) = (\beta_0 \beta_1 \cdots \beta_{m-1}) \Gamma;$$

若一个元素相对于 B_1 的表示是 $(\alpha_0 \alpha_1 \cdots \alpha_{m-1})$, 则它相对于 B_2 的表示为:

$$(\beta_0 \beta_1 \cdots \beta_{m-1}) = (\alpha_0 \alpha_1 \cdots \alpha_{m-1}) \Gamma^{-1},$$

其中 Γ^{-1} 表示 Γ 的模2逆。

示例8:

假定 B_1 是 F_{2^5} 的多项式基, 其域多项式为 $t^5 + t^2 + 1$, B_2 是 F_{2^5} 的II型最优正规基, 则域多项式为 $f(t) = t^5 + t^4 + t^2 + t + 1$, $f(t)$ 相对于 B_1 的一个根为 $\gamma = (01100)$, 则:

$$\gamma = (01100),$$

$$\gamma^2 = (11010),$$

$$\begin{aligned} \gamma^4 &= (00011), \\ \gamma^8 &= (00101), \\ \gamma^{16} &= (10001). \end{aligned}$$

$$\text{因此 } \Gamma = \begin{Bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{Bmatrix}, \quad \Gamma^{-1} = \begin{Bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{Bmatrix}.$$

若 λ 相对于 B_2 的表示为 $\lambda=(11001)$, 则 λ 相对于 B_1 的表示是: $(00011)=(11001)\Gamma$;

若 λ 相对于 B_1 的表示为 $\lambda=(10110)$, 则 λ 相对于 B_2 的表示是: $(11000)=(10110)\Gamma^{-1}$ 。

B.2.4 F_2 上多项式不可约性的检测

设 $f(x)$ 是 F_2 上的多项式, 利用下面的算法可以有效地检测 $f(x)$ 的不可约性。

输入: F_2 上的多项式 $f(x)$ 。

输出: 若 $f(x)$ 在 F_2 上不可约, 则输出“正确”; 否则, 输出“错误”。

- a) 置 $d = \deg(f(x))$;
- b) 置 $u(x) = x$;
- c) 对 i 从1到 $\lfloor d/2 \rfloor$ 执行:
 - c.1) 置 $u(x) = u(x)^2 \bmod f(x)$;
 - c.2) 置 $g(x) = \gcd(u(x) + x, f(x))$;
 - c.3) 若 $g(x) \neq 1$, 则输出“错误”并终止;
- d) 输出“正确”。

B.3 椭圆曲线算法

B.3.1 椭圆曲线阶的计算

对于有限域上随机的椭圆曲线, 其阶的计算是一个相当复杂的问题。目前有效的计算方法有SEA算法和Sato算法。关于计算椭圆曲线阶的详细描述参见(Lehmann et al. 1994)、(Müller 1995)、(Sato 2000)、(Sato 2002)、(Sato et al. 2003)、(Schoof 1985)和(Schoof 1995)。

B.3.2 椭圆曲线上点的寻找

给定有限域上的椭圆曲线, 利用下面的算法可有效地找出曲线上任意一个非无穷远点。

B.3.2.1 F_p 上的椭圆曲线

输入: 素数 p , F_p 上一条椭圆曲线 E 的参数 a, b 。

输出: E 上一个非无穷远点。

- a) 选取随机整数 x , $0 \leq x < p$;
- b) 置 $\alpha = (x^3 + ax + b) \bmod p$;
- c) 若 $\alpha = 0$, 则输出 $(x, 0)$ 并终止;
- d) 求 $\alpha \bmod p$ 的平方根(参见附录B.1.4);
- e) 若步骤d)的输出是“不存在平方根”, 则返回步骤a);
否则, 步骤d)的输出是整数 y , $0 < y < p$, 且 $y^2 \equiv \alpha \pmod{p}$;
- f) 输出 (x, y) 。

B.3.2.2 F_{2^m} 上的椭圆曲线

输入：二元扩域 F_{2^m} ， F_{2^m} 上的椭圆曲线E的参数 a, b 。

输出： E 上一个非无穷远点。

a) 在 F_{2^m} 中选取随机元素 x ;

b) 若 $x = 0$ ，则输出 $(0, b^{2^{m-1}})$ 并终止;

c) 置 $\alpha = x^3 + ax^2 + b$;

d) 若 $\alpha = 0$ ，则输出 $(x, 0)$ 并终止;

e) 置 $\beta = x^{-2}\alpha$;

f) 求 z ，使得 $z^2 + z = \beta$ （参见附录B.1.6）;

g) 若步骤f)的输出是“无解”，则返回步骤a); 否则，步骤f)的输出是解 z ;

h) 置 $y = x \cdot z$;

i) 输出 (x, y) 。

附 录 C
(资料性附录)
曲线示例

C.1 一般要求

在此附录中所有值均以16进制表示，左边为高位，右边为低位。

C.2 F_p 上椭圆曲线

椭圆曲线方程为： $y^2 = x^3 + ax + b$

示例1: F_p -192曲线

素数 p :

BDB6F4FE 3E8B1D9E 0DA8C0D4 6F4C318C EFE4AFE3 B6B8551F

系数 a :

BB8E5E8F BC115E13 9FE6A814 FE48AAA6 F0ADA1AA 5DF91985

系数 b :

1854BEBD C31B21B7 AEF80AB 0ECD10D5 B1B3308E 6DBF11C1

基点 $G = (x,y)$ ，其阶记为 n 。

坐标 x :

4AD5F704 8DE709AD 51236DE6 5E4D4B48 2C836DC6 E4106640

坐标 y :

02BB3A02 D4AAADAC AE24817A 4CA3A1B0 14B52704 32DB27D2

阶 n :

BDB6F4FE 3E8B1D9E 0DA8C0D4 0FC96219 5DFAE76F 56564677

示例2: F_p -256曲线

素数 p :

8542D69E 4C044F18 E8B92435 BF6FF7DE 45728391 5C45517D 722EDB8B 08F1DFC3

系数 a :

787968B4 FA32C3FD 2417842E 73BBFEFF 2F3C848B 6831D7E0 EC65228B 3937E498

系数 b :

63E4C6D3 B23B0C84 9CF84241 484BFE48 F61D59A5 B16BA06E 6E12D1DA 27C5249A

基点 $G = (x,y)$ ，其阶记为 n 。

坐标 x :

421DEBD6 1B62EAB6 746434EB C3CC315E 32220B3B ADD50BDC 4C4E6C14 7FEDD43D

坐标 y :

0680512B CBB42C07 D47349D2 153B70C4 E5D7FDFC BFA36EA1 A85841B9 E46E09A2

阶 n :

8542D69E 4C044F18 E8B92435 BF6FF7DD 29772063 0485628D 5AE74EE7 C32E79B7

C.3 F_{2^m} 上椭圆曲线

椭圆曲线方程为： $y^2 + xy = x^3 + ax^2 + b$

示例3: F_{2^m} -193曲线

基域生成多项式： $x^{193} + x^{15} + 1$

系数 a : 0

系数 b :

00 2FE22037 B624DBEB C4C618E1 3FD998B1 A18E1EE0 D05C46FB

基点 $G = (x, y)$, 其阶记为 n 。

坐标 x :

00 D78D47E8 5C936440 71BC1C21 2CF994E4 D21293AA D8060A84

坐标 y :

00 615B9E98 A31B7B2F DDEEECB7 6B5D8755 86293725 F9D2FC0C

阶 n :

80000000 00000000 00000000 43E9885C 46BF45D8 C5EBF3A1

示例4: F_{2^m} -257曲线

基域生成多项式: $x^{257} + x^{12} + 1$

系数 a : 0

系数 b :

00 E78BCD09 746C2023 78A7E72B 12BCE002 66B9627E CB0B5A25 367AD1AD 4CC6242B

基点 $G = (x, y)$, 其阶记为 n 。

坐标 x :

00 CDB9CA7F 1E6B0441 F658343F 4B10297C 0EF9B649 1082400A 62E7A748 5735FADD

坐标 y :

01 3DE74DA6 5951C4D7 6DC89220 D5F7777A 611B1C38 BAE260B1 75951DC8 060C2B3E

阶 n :

7FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF BC972CF7 E6B6F900 945B3C6A 0CF6161D

附 录 D
(资料性附录)

椭圆曲线方程参数的拟随机生成及验证

D.1 椭圆曲线方程参数的拟随机生成

D.1.1 F_p 上椭圆曲线方程参数的拟随机生成

方式1:

输入: 素域的规模 p 。

输出: 比特串 $SEED$ 及 F_p 中的元素 a, b 。

- a) 任意选择长度至少为192的比特串 $SEED$;
- b) 计算 $H = H_{256}(SEED)$, 并记 $H = (h_{255}, h_{254}, \dots, h_0)$;
- c) 置 $R = \sum_{i=0}^{255} h_i 2^i$;
- d) 置 $r = R \bmod p$;
- e) 任意选择 F_p 中的元素 a 和 b , 使 $r \cdot b^2 \equiv a^3 \pmod{p}$;
- f) 若 $(4a^3 + 27b^2) \bmod p = 0$, 则转步骤a);
- g) 所选择的 F_p 上的椭圆曲线为 $E: y^2 = x^3 + ax + b$;
- h) 输出 $(SEED, a, b)$ 。

方式2:

输入: 素域的规模 p 。

输出: 比特串 $SEED$ 及 F_p 中的元素 a, b 。

- a) 任意选择长度至少为192的比特串 $SEED$;
- b) 计算 $H = H_{256}(SEED)$, 并记 $H = (h_{255}, h_{254}, \dots, h_0)$;
- c) 置 $R = \sum_{i=0}^{255} h_i 2^i$;
- d) 置 $r = R \bmod p$;
- e) 置 $b = r$;
- f) 取 F_p 中的元素 a 为某固定值;
- g) 若 $(4a^3 + 27b^2) \bmod p = 0$, 则转步骤a);
- h) 所选择的 F_p 上的椭圆曲线为 $E: y^2 = x^3 + ax + b$;
- i) 输出 $(SEED, a, b)$ 。

D.1.2 F_{2^m} 上椭圆曲线方程参数的拟随机生成

输入: 域的规模 $q = 2^m$, F_{2^m} 的约化多项式 $f(x) = x^m + f_{m-1}x^{m-1} + \dots + f_2x^2 + f_1x + f_0$ (其中 $f_i \in F_2$, $i = 0, 1, \dots, m-1$)。

输出: 比特串 $SEED$ 及 F_{2^m} 中的元素 a, b 。

- a) 任意选择至少192比特长的比特串 $SEED$;
- b) 计算 $H = H_{256}(SEED)$, 并记 $H = (h_{255}, h_{254}, \dots, h_0)$;
- c) 若 $i \geq 256$, 令 $h_i = 1$, 置比特串 $HH = (h_{m-1}, h_{m-2}, \dots, h_0)$, b 为与 HH 对应的 F_{2^m} 中的元素;
- d) 若 $b = 0$, 则转步骤a);
- e) 取 a 为 F_{2^m} 中的任意元素;
- f) 所选择的 F_{2^m} 上的椭圆曲线为 $E: y^2 + xy = x^3 + ax^2 + b$;
- g) 输出 $(SEED, a, b)$ 。

D.2 椭圆曲线方程参数的验证

D.2.1 F_p 上椭圆曲线方程参数的验证

方式1:

输入: 比特串 $SEED$ 及 F_p 中的元素 a, b 。

输出: 输入参数“有效”或“无效”。

- 计算 $H' = H_{256}(SEED)$, 并记 $H' = (h_{255}, h_{254}, \dots, h_0)$;
- 置 $R' = \sum_{i=0}^{255} h_i 2^i$;
- 置 $r' = R' \bmod p$;
- 若 $r' \cdot b^2 \equiv a^3 \pmod{p}$, 则输出“有效”; 否则输出“无效”。

方式2:

输入: 比特串 $SEED$ 及 F_p 中的元素 b 。

输出: 输入参数“有效”或“无效”。

- 计算 $H' = H_{256}(SEED)$, 并记 $H' = (h_{255}, h_{254}, \dots, h_0)$;
- 置 $R' = \sum_{i=0}^{255} h_i 2^i$;
- 置 $r' = R' \bmod p$;
- 若 $r' = b$, 则输出“有效”; 否则输出“无效”。

D.2.2 F_{2^m} 上椭圆曲线方程参数的验证

输入: 比特串 $SEED$ 及 F_{2^m} 中的元素 b 。

输出: 输入参数“有效”或“无效”。

- 计算 $H' = H_{256}(SEED)$, 并记 $H' = (h_{255}, h_{254}, \dots, h_0)$;
- 若 $i \geq 256$, 令 $h_i = 1$, 置比特串 $HH' = (h_{m-1}, h_{m-2}, \dots, h_0)$, b' 为与 HH' 对应的 F_{2^m} 中的元素;
- 若 $b' = b$, 则输出“有效”; 否则输出“无效”。

注: 本附录中的函数 $H_{256}()$ 是输出长度为256比特的密码杂凑函数。

参考文献

- [1] GB/T 15843.1-1999 信息技术安全技术实体鉴别第1部分：概述
- [2] GB/T 5271.8-2001 信息技术词汇第8部分：安全
- [3] Agnew G, Beth T, Mullin R, et al. 1993. Arithmetic operations in $GF(2^m)$. *Journal of Cryptology*, (6):3~13
- [4] Agnew G, Mullin R, Onyszchuk I, et al. 1991. An implementation for a fast public-key cryptosystem. *Journal of Cryptology*, (3):63~79
- [5] ANSI X9.62-1999 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standards Institute
- [6] ANSI X9.63-2001 Public Key Cryptography for the Financial Services Industry: Key Agreement and key Transport Using Elliptic Curve Cryptography .American Nation Standard Institute
- [7] Brickell E, Gordon D, McCurley K, et al. 1993. Fast Exponentiation with precomputation. *Advances in Cryptology - EUROCRYPT' 92*. LNCS 658. Berlin: Springer-Verlag .200~207
- [8] Blake I, Seroussi G, Smart N. 1999. *Elliptic Curves in Cryptography*. Cambridge: Cambridge University Press
- [9] ISO/IEC 15946-1: 2002 Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 1: General
- [10] ISO/IEC 15946-2: 2002 Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 2: Digital signatures
- [11] ISO/IEC 15946-3: 2002 Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 3: Key establishment
- [12] ISO/IEC 15946-4: 2003 Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 4: Digital signatures giving message recovery
- [13] ITU-T Recommendation X.680 Information Technology — Abstract Syntax Notation One (ASN.1): Specification of Basic Notation (eqv ISO/IEC 8824-1)
- [14] ITU-T Recommendation X.681 Information Technology — Abstract Syntax Notation One (ASN.1): Information Object Specification (eqv ISO/IEC 8824-2)
- [15] ITU-T Recommendation X.682 Information Technology — Abstract Syntax Notation One (ASN.1): Constraint Specification (eqv ISO/IEC 8824-3)
- [16] ITU-T Recommendation X.683 Information Technology — Abstract Syntax Notation One (ASN.1): Parametrization of ASN.1 Specifications (eqv ISO/IEC 8824-4)
- [17] ITU-T Recommendation X.690 Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) (eqv ISO/IEC 8825-1)
- [18] ITU-T Recommendation X.691 Information Technology — ASN.1 Encoding Rules: Specification of Packed Encoding Rules (PER) (eqv ISO/IEC 8825-2)

- [19] Knuth D. 1981. The Art of Computer Programming .v.2. 2nd ed, Reading(MA): Addison-Wesley
- [20] Koblitz N.1987. Elliptic curve cryptosystems. Mathematics of Computation, (48)203~209
- [21] Lehmann F , Maurer M, Müller V, et al. 1994. Counting the number of points on elliptic curves over finite field of characteristic greater than three.In:Adleman L,Huang M D,ed. Algorithmic Number Theory. LNCS 877.Berlin: Springer-Verlag.60~70
- [22] Lidl R,Niederreiter H.1987. Finite Fields. Cambridge:Cambridge University Press
- [23] McEliece R.1987. Finite Fields for Computer Scientists and Engineers. Boston:Kluwer Academic Publishers
- [24] Menezes A.1993. Elliptic Curve Public Key Cryptosystems.Boston: Kluwer Academic Publishers
- [25] Menezes A,Okamoto T, Vanstone S.1993. Reducing elliptic curve logarithms to logarithms in a finite field.IEEE Transactions on Information Theory, 39:1639~1646
- [26] Müller V.1995. Counting the number of points on elliptic curves over finite fields of characteristic greater than three:[Doctorate Dissertation].Saarlandes: University of Saarlandes
- [27] Pollard J.1978. Monte Carlo methods for index computation mod p. Mathematics of Computation, 32:918-924
- [28] Satoh T,Araki K.1998. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves.Comment.Math.Univ.St.Paul.,47(1):81~92
- [29] Satoh T.2000. The canonical lift of an ordinary elliptic curve over a finite fields and its point counting. J.Ramanujan Math.Soc.,15:247~270
- [30] Satoh T. 2002. On p-adic point counting algorithms for elliptic curves over finite fields. In:Fieker C,Kohel D R,eds. Algorithmic Number Theory,LNCS 2369, Berlin: Springer-Verlag,43~66
- [31] Satoh T,Skjerna B ,Taguchi Y.2003. Fast computation of canonical lifts of elliptic curves and its application to point counting. Finite Fields Appl.,9:89~101
- [32] Schoof R.1985. Elliptic curves over finite fields and the computation of square roots mod p.Mathematics of Computation, 44(170):483~494
- [33] Schoof R.1995. Counting Points on Elliptic Curves over Finite Fields. Jl. de Theorie des Nombres de Bordeaux, 7:219~254
- [34] Silverman J.1986. The Arithmetic of Elliptic Curves. Berlin: Springer-Verlag , GTM 106
- [35] Smart N.1999. The discrete logarithm problem on elliptic curves of trace one. Journal of Cryptology,12(3):193~196
- [36] ГОСТ Р 34.10-2001 ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ — КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ — Процессы формирования и проверки электронной цифровой подписи. ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

SM2椭圆曲线公钥密码算法 第2部分：数字签名算法

Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves

Part 2: Digital Signature Algorithm

国家密码管理局

2010年12月

目 次

前 言	II
引 言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 符号	2
5 数字签名算法	3
5.1 总则	3
5.2 椭圆曲线系统参数	3
5.3 用户密钥对	3
5.4 辅助函数	4
5.4.1 概述	4
5.4.2 密码杂凑函数	4
5.4.3 随机数发生器	4
5.5 用户其它信息	4
6 数字签名的生成算法及流程	4
6.1 数字签名的生成算法	4
6.2 数字签名生成算法流程	5
7 数字签名的验证算法及流程	5
7.1 数字签名的验证算法	5
7.2 数字签名验证算法流程	6
附录A (资料性附录) 数字签名与验证示例	7
A.1 一般要求	7
A.2 F_p 上的椭圆曲线数字签名	7
A.3 F_{2^m} 上的椭圆曲线数字签名	8

前 言

《SM2椭圆曲线公钥密码算法》分为四个部分：

- 第1部分：总则
 - 第2部分：数字签名算法
 - 第3部分：密钥交换协议
 - 第4部分：公钥加密算法
- 本部分为第2部分。
本部分的附录A为资料性附录。

引 言

N.Koblitz和V.Miller在1985年各自独立地提出将椭圆曲线应用于公钥密码系统。椭圆曲线公钥密码所基于的曲线性质如下：

- 有限域上椭圆曲线在点加运算下构成有限交换群，且其阶与基域规模相近；
- 类似于有限域乘法群中的乘幂运算，椭圆曲线多倍点运算构成一个单向函数。

在多倍点运算中，已知多倍点与基点，求解倍数的问题称为椭圆曲线离散对数问题。对于一般椭圆曲线的离散对数问题，目前只存在指数级计算复杂度的求解方法。与大数分解问题及有限域上离散对数问题相比，椭圆曲线离散对数问题的求解难度要大得多。因此，在相同安全程度要求下，椭圆曲线密码较其它公钥密码所需的密钥规模要小得多。

本部分描述了基于椭圆曲线的数字签名算法。

SM2椭圆曲线公钥密码算法

第2部分：数字签名算法

1 范围

本部分规定了SM2椭圆曲线公钥密码算法的数字签名算法，包括数字签名生成算法和验证算法，并给出了数字签名与验证示例及其相应的流程。

本部分适用于商用密码应用中的数字签名和验证，可满足多种密码应用中的身份认证和数据完整性、真实性的安全需求。同时，本部分还可为安全产品生产商提供产品和技术的标准定位以及标准化的参考，提高安全产品的可信性与互操作性。

2 规范性引用文件

下列文件中的条款通过本部分引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

SM2椭圆曲线公钥密码算法 第1部分：总则

3 术语和定义

下列术语和定义适用于本部分。

3.1

数字签名 **digital signature**

附加在数据单元上的一些数据，或是对数据单元所作密码变换的结果，当正常应用时提供如下服务：

- a) 数据来源的确认；
- b) 数据完整性的验证；
- c) 签名者不可抵赖的保证。

[ANSI X9.63-2001]

3.2

消息 **message**

任意有限长度的比特串。

[ISO/IEC 15946-4 3.7]

3.3

签名消息 **signed message**

由消息以及该消息的签名部分所组成的一组数据项。

[ISO/IEC 15946-4 3.14]

3.4

密钥 **key**

确定密码函数运算的一个参数，它用于：

- a) 加密或解密变换；
- b) 同步产生共享秘密；
- c) 数字签名的生成或验证。

[ANSI X9.63-2001]

3.5

私钥 private key

在非对称密码体制中，实体自己保持的、只有实体自身知道的一种密钥。

[ISO/IEC 15946-3 3.24]

3.6

公钥 public key

在非对称密码体制中，实体的一种可以公开的密钥。

[ISO/IEC 15946-3 3.25]

3.7

签名密钥 signature key

在数字签名生成过程中由签名者专用的秘密数据项，即签名者的私钥。

[ISO/IEC 15946-2 3.1.7]

3.8

随机数序列 random number sequence

数列中的每一项在已知其它项的情况下都无法被推断的一种数列。

[ГОСТ Р 34.10-2001 3.1.7]

3.9

杂凑函数 hash function

将一个比特串映射为一个固定长度比特串的函数。该函数满足如下性质：

- a) 对于任意给定的输出，要找到其对应的输入，在计算上是不可行的；
- b) 对于任意给定的输入，要找到输出相同的另一个输入，在计算上是不可行的。

注：计算可行性依赖于具体的安全需求和环境。

[ISO/IEC 15946-2 3.1.3]

3.10

杂凑值 hash value

杂凑函数作用于一条消息时输出的比特串。

[ISO/IEC 15946-2 3.1.2]

3.11

签名生成过程 signature process

输入消息、签名密钥和椭圆曲线系统参数，并输出数字签名的过程。

[ISO/IEC 15946-2 3.1.8]

3.12

不可抵赖 non-repudiation

证明某一动作或事件已经发生的能力，以使事后不能否认这一动作或事件。

[ISO/IEC 13888-1]

3.13

可辨别标识 distinguishing identifier

可以无歧义辨别某一实体身份的信息。

[ISO/IEC 15946-3 3.9]

4 符号

下列符号适用于本部分。

A,B: 使用公钥密码系统的两个用户。

a, b : F_q 中的元素，它们定义 F_q 上的一条椭圆曲线 E 。

d_A : 用户A的私钥。

$E(F_q)$: F_q 上椭圆曲线 E 的所有有理点(包括无穷远点 O)组成的集合。

e : 密码杂凑函数作用于消息 M 的输出值。

e' : 密码杂凑函数作用于消息 M' 的输出值。

F_q : 包含 q 个元素的有限域。

G : 椭圆曲线的一个基点, 其阶为素数。

$H_v(\cdot)$: 消息摘要长度为 v 比特的密码杂凑函数。

ID_A : 用户A的可辨别标识。

M : 待签名的消息。

M' : 待验证消息。

$\text{mod } n$: 模 n 运算。例如, $23 \text{ mod } 7 = 2$ 。

n : 基点 G 的阶(n 是 $\#E(F_q)$ 的素因子)。

O : 椭圆曲线上的一个特殊点, 称为无穷远点或零点, 是椭圆曲线加法群的单位元。

P_A : 用户A的公钥。

q : 有限域 F_q 中元素的数目。

$x \parallel y$: x 与 y 的拼接, 其中 x 、 y 可以是比特串或字节串。

Z_A : 关于用户A的可辨别标识、部分椭圆曲线系统参数和用户A公钥的杂凑值。

(r,s) : 发送的签名。

(r',s') : 收到的签名。

$[k]P$: 椭圆曲线上点 P 的 k 倍点, 即, $[k]P = \underbrace{P + P + \dots + P}_{k \text{ 个}}$, k 是正整数。

$[x,y]$: 大于或等于 x 且小于或等于 y 的整数的集合。

$\lceil x \rceil$: 顶函数, 大于或等于 x 的最小整数。例如, $\lceil 7 \rceil = 7$, $\lceil 8.3 \rceil = 9$ 。

$\lfloor x \rfloor$: 底函数, 小于或等于 x 的最大整数。例如, $\lfloor 7 \rfloor = 7$, $\lfloor 8.3 \rfloor = 8$ 。

$\#E(F_q)$: $E(F_q)$ 上点的数目, 称为椭圆曲线 $E(F_q)$ 的阶。

5 数字签名算法

5.1 总则

数字签名算法由一个签名者对数据产生数字签名, 并由一个验证者验证签名的可靠性。每个签名者有一个公钥和一个私钥, 其中私钥用于产生签名, 验证者用签名者的公钥验证签名。在签名的生成过程之前, 要用密码杂凑函数对 \overline{M} (包含 Z_A 和待签消息 M)进行压缩; 在验证过程之前, 要用密码杂凑函数对 \overline{M}' (包含 Z_A 和验证消息 M')进行压缩。

5.2 椭圆曲线系统参数

椭圆曲线系统参数包括有限域 F_q 的规模 q (当 $q=2^m$ 时, 还包括元素表示法的标识和约化多项式); 定义椭圆曲线 $E(F_q)$ 的方程的两个元素 $a, b \in F_q$; $E(F_q)$ 上的基点 $G=(x_G, y_G)$ ($G \neq O$), 其中 x_G 和 y_G 是 F_q 中的两个元素; G 的阶 n 及其它可选项(如 n 的余因子 h 等)。

椭圆曲线系统参数及其验证应符合本文本第1部分第5章的规定。

5.3 用户密钥对

用户A的密钥对包括其私钥 d_A 和公钥 $P_A=[d_A]G=(x_A, y_A)$ 。

用户密钥对的生成算法与公钥验证算法应符合本文本第1部分第6章的规定。

5.4 辅助函数

5.4.1 概述

在本部分规定的椭圆曲线数字签名算法中，涉及到两类辅助函数：密码杂凑函数与随机数发生器。

5.4.2 密码杂凑函数

本部分规定使用国家密码管理局批准的密码杂凑算法，如SM3密码杂凑算法。

5.4.3 随机数发生器

本部分规定使用国家密码管理局批准的随机数发生器。

5.5 用户其它信息

作为签名者的用户A具有长度为 $entlen_A$ 比特的可辨别标识 ID_A ，记 $ENTL_A$ 是由整数 $entlen_A$ 转换而成的两个字节，在本部分规定的椭圆曲线数字签名算法中，签名者和验证者都需要用密码杂凑函数求得用户A的杂凑值 Z_A 。按本文本第1部分4.2.5和4.2.4给出的细节，将椭圆曲线方程参数 a 、 b 、 G 的坐标 x_G 、 y_G 和 P_A 的坐标 x_A 、 y_A 的数据类型转换为比特串， $Z_A = H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ 。

6 数字签名的生成算法及流程

6.1 数字签名的生成算法

设待签名的消息为 M ，为了获取消息 M 的数字签名 (r,s) ，作为签名者的用户A应实现以下运算步骤：

A1: 置 $\overline{M} = Z_A \parallel M$;

A2: 计算 $e = H_v(\overline{M})$ ，按本文本第1部分4.2.3和4.2.2给出的细节将 e 的数据类型转换为整数；

A3: 用随机数发生器产生随机数 $k \in [1, n-1]$;

A4: 计算椭圆曲线点 $(x_1, y_1) = [k]G$ ，按本文本第1部分4.2.7给出的细节将 x_1 的数据类型转换为整数；

A5: 计算 $r = (e + x_1) \bmod n$ ，若 $r=0$ 或 $r+k=n$ 则返回A3；

A6: 计算 $s = ((1 + d_A)^{-1} \cdot (k - r \cdot d_A)) \bmod n$ ，若 $s=0$ 则返回A3；

A7: 按本文本第1部分4.2.1给出的细节将 r 、 s 的数据类型转换为字节串，消息 M 的签名为 (r,s) 。

注：数字签名生成过程的示例参见附录A。

6.2 数字签名生成算法流程

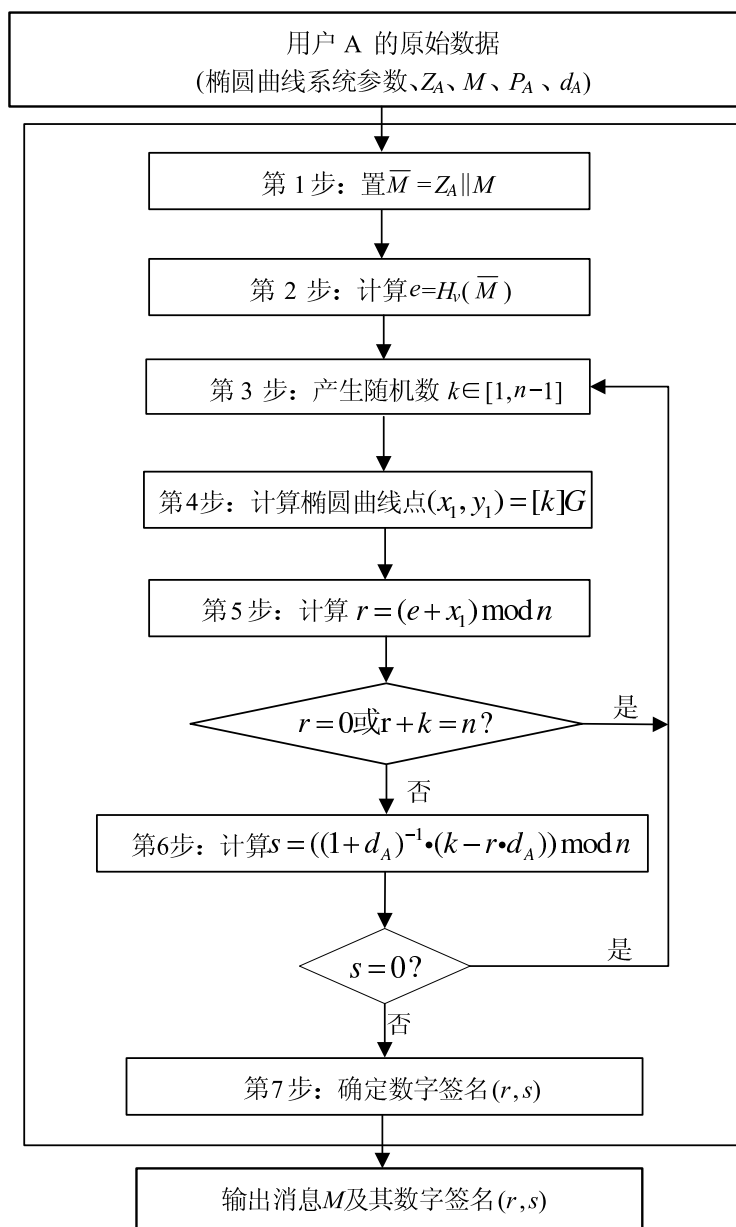


图1 数字签名生成算法流程

7 数字签名的验证算法及流程

7.1 数字签名的验证算法

为了检验收到的消息 M' 及其数字签名 (r', s') ，作为验证者的用户 **B** 应实现以下运算步骤：

B1: 检验 $r' \in [1, n-1]$ 是否成立，若不成立则验证不通过；

B2: 检验 $s' \in [1, n-1]$ 是否成立，若不成立则验证不通过；

B3: 置 $\bar{M}' = Z_A \parallel M'$ ；

B4: 计算 $e' = H_v(\bar{M}')$ ，按本文第1部分4.2.3和4.2.2给出的细节将 e' 的数据类型转换为整数；

B5: 按本文第1部分4.2.2给出的细节将 r' 、 s' 的数据类型转换为整数，计算 $t = (r' + s') \bmod n$ ，若 $t = 0$ ，则验证不通过；

B6: 计算椭圆曲线点 $(x'_1, y'_1) = [s']G + [t]P_A$ ；

B7: 按本文本第1部分4.2.7给出的细节将 x'_1 的数据类型转换为整数, 计算 $R = (e' + x'_1) \bmod n$, 检验 $R=r'$ 是否成立, 若成立则验证通过; 否则验证不通过。

注: 如果 Z_A 不是用户A所对应的杂凑值, 验证自然不通过。数字签名验证过程的示例参见附录A。

7.2 数字签名验证算法流程

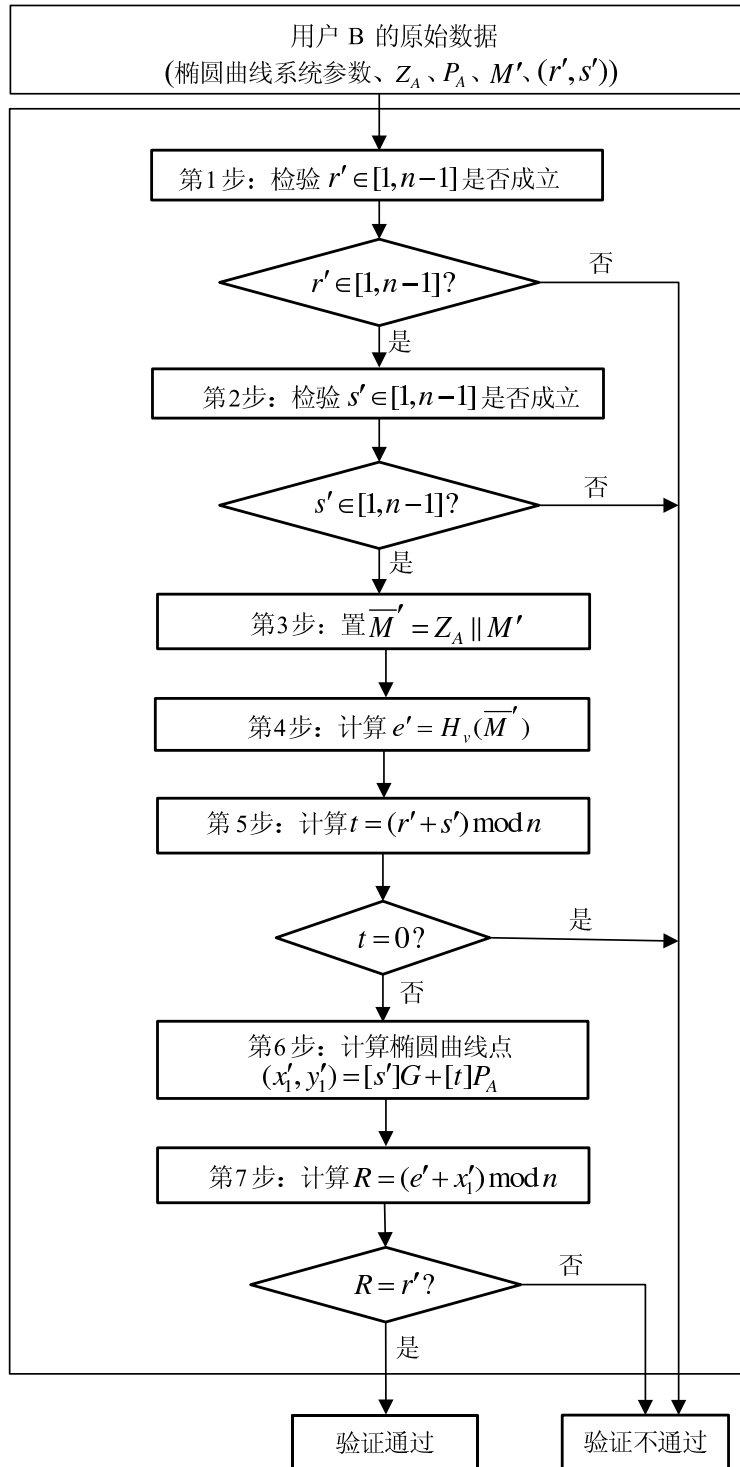


图2 数字签名验证算法流程

附录 A
(资料性附录)
数字签名与验证示例

A.1 一般要求

本附录选用《SM3密码杂凑算法》给出的密码杂凑函数，其输入是长度小于 2^{64} 的消息比特串，输出是长度为256比特的杂凑值，记为 $H_{256}()$ 。

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，消息采用ASCII编码。

设用户A的身份是：ALICE123@YAHOO.COM。用ASCII编码记 ID_A ：

414C 49434531 32334059 41484F4F 2E434F4D。 $ENTL_A=0090$ 。

A.2 F_p 上的椭圆曲线数字签名

椭圆曲线方程为： $y^2 = x^3 + ax + b$

示例1: F_p -256

素数 p :

8542D69E 4C044F18 E8B92435 BF6FF7DE 45728391 5C45517D 722EDB8B 08F1DFC3

系数 a :

787968B4 FA32C3FD 2417842E 73BBFEFF 2F3C848B 6831D7E0 EC65228B 3937E498

系数 b :

63E4C6D3 B23B0C84 9CF84241 484BFE48 F61D59A5 B16BA06E 6E12D1DA 27C5249A

基点 $G=(x_G, y_G)$ ，其阶记为 n 。

坐标 x_G :

421DEBD6 1B62EAB6 746434EB C3CC315E 32220B3B ADD50BDC 4C4E6C14 7FEDD43D

坐标 y_G :

0680512B CBB42C07 D47349D2 153B70C4 E5D7FDFC BFA36EA1 A85841B9 E46E09A2

阶 n :

8542D69E 4C044F18 E8B92435 BF6FF7DD 29772063 0485628D 5AE74EE7 C32E79B7

待签名的消息 M : message digest

私钥 d_A :

128B2FA8 BD433C6C 068C8D80 3DFF7979 2A519A55 171B1B65 0C23661D 15897263

公钥 $P_A=(x_A, y_A)$:

坐标 x_A :

0AE4C779 8AA0F119 471BEE11 825BE462 02BB79E2 A5844495 E97C04FF 4DF2548A

坐标 y_A :

7C0240F8 8F1CD4E1 6352A73C 17B7F16F 07353E53 A176D684 A9FE0C6B B798E857

杂凑值 $Z_A=H_{256}(ENTL_A || ID_A || a || b || x_G || y_G || x_A || y_A)$ 。

Z_A :

F4A38489 E32B45B6 F876E3AC 2168CA39 2362DC8F 23459C1D 1146FC3D BFB7BC9A

签名各步骤中的有关值:

$\overline{M}=Z_A || M$:

F4A38489 E32B45B6 F876E3AC 2168CA39 2362DC8F 23459C1D 1146FC3D BFB7BC9A
6D657373 61676520 64696765 7374

密码杂凑函数值 $e=H_{256}(\overline{M})$:

B524F552 CD82B8B0 28476E00 5C377FB1 9A87E6FC 682D48BB 5D42E3D9 B9EFFE76

产生随机数 k :

6CB28D99 385C175C 94F94E93 4817663F C176D925 DD72B727 260DBAAE 1FB2F96F

计算椭圆曲线点 $(x_1, y_1)=[k]G$:

坐标 x_1 :

110FCDA5 7615705D 5E7B9324 AC4B856D 23E6D918 8B2AE477 59514657 CE25D112

坐标 y_1 :

1C65D68A 4A08601D F24B431E 0CAB4EBE 084772B3 817E8581 1A8510B2 DF7ECA1A

计算 $r=(e+x_1) \bmod n$:

40F1EC59 F793D9F4 9E09DCEF 49130D41 94F79FB1 EED2CAA5 5BACDB49 C4E755D1

$(1+d_A)^{-1}$:

79BFCF30 52C80DA7 B939E0C6 914A18CB B2D96D85 55256E83 122743A7 D4F5F956

计算 $s = ((1+d_A)^{-1} \cdot (k-r \cdot d_A)) \bmod n$:

6FC6DAC3 2C5D5CF1 0C77DFB2 0F7C2EB6 67A45787 2FB09EC5 6327A67E C7DEEBE7

消息 M 的签名为 (r, s) :

值 r :

40F1EC59 F793D9F4 9E09DCEF 49130D41 94F79FB1 EED2CAA5 5BACDB49 C4E755D1

值 s :

6FC6DAC3 2C5D5CF1 0C77DFB2 0F7C2EB6 67A45787 2FB09EC5 6327A67E C7DEEBE7

验证各步骤中的有关值:

密码杂凑函数值 $e' = H_{256}(\overline{M}')$:

B524F552 CD82B8B0 28476E00 5C377FB1 9A87E6FC 682D48BB 5D42E3D9 B9EFFE76

计算 $t=(r'+s') \bmod n$:

2B75F07E D7ECE7CC C1C8986B 991F441A D324D6D6 19FE06DD 63ED32E0 C997C801

计算椭圆曲线点 $(x'_0, y'_0)=[s']G$

坐标 x'_0 :

7DEACE5F D121BC38 5A3C6317 249F413D 28C17291 A60DFD83 B835A453 92D22B0A

坐标 y'_0 :

2E49D5E5 279E5FA9 1E71FD8F 693A64A3 C4A94611 15A4FC9D 79F34EDC 8BDDEBD0

计算椭圆曲线点 $(x'_{00}, y'_{00})=[t]P_A$:

坐标 x'_{00} :

1657FA75 BF2ADCDC 3C1F6CF0 5AB7B45E 04D3ACBE 8E4085CF A669CB25 64F17A9F

坐标 y'_{00} :

19F0115F 21E16D2F 5C3A485F 8575A128 BBCDDF80 296A62F6 AC2EB842 DD058E50

计算椭圆曲线点 $(x'_1, y'_1)=[s']G + [t]P_A$:

坐标 x'_1 :

110FCDA5 7615705D 5E7B9324 AC4B856D 23E6D918 8B2AE477 59514657 CE25D112

坐标 y'_1 :

1C65D68A 4A08601D F24B431E 0CAB4EBE 084772B3 817E8581 1A8510B2 DF7ECA1A

计算 $R = (e' + x'_1) \bmod n$:

40F1EC59 F793D9F4 9E09DCEF 49130D41 94F79FB1 EED2CAA5 5BACDB49 C4E755D1

A.3 F_{2^m} 上的椭圆曲线数字签名

椭圆曲线方程为: $y^2 + xy = x^3 + ax^2 + b$

示例2: F_{2^m} -257

基域生成多项式: $x^{257} + x^{12} + 1$

系数 a : 0

系数 b :

00 E78BCD09 746C2023 78A7E72B 12BCE002 66B9627E CB0B5A25 367AD1AD 4CC6242B

基点 $G=(x_G, y_G)$, 其阶记为 n 。

坐标 x_G :

00 CDB9CA7F 1E6B0441 F658343F 4B10297C 0EF9B649 1082400A 62E7A748 5735FADD

坐标 y_G :

01 3DE74DA6 5951C4D7 6DC89220 D5F7777A 611B1C38 BAE260B1 75951DC8 060C2B3E

阶 n :

7FFFFFFFF FFFFFFFFF FFFFFFFFF FFFFFFFFF BC972CF7 E6B6F900 945B3C6A 0CF6161D

待签名的消息 M : message digest

私钥 d_A :

771EF3DB FF5F1CDC 32B9C572 93047619 1998B2BF 7CB981D7 F5B39202 645F0931

公钥 $P_A=(x_A, y_A)$:

坐标 x_A :

01 65961645 281A8626 607B917F 657D7E93 82F1EA5C D931F40F 6627F357 542653B2

坐标 y_A :

01 68652213 0D590FB8 DE635D8F CA715CC6 BF3D05BE F3F75DA5 D5434544 48166612

杂凑值 $Z_A=H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ 。

Z_A :

26352AF8 2EC19F20 7BBC6F94 74E11E90 CE0F7DDA CE03B27F 801817E8 97A81FD5

签名各步骤中的有关值:

$\overline{M}=Z_A \parallel M$:

26352AF8 2EC19F20 7BBC6F94 74E11E90 CE0F7DDA CE03B27F 801817E8 97A81FD5
6D657373 61676520 64696765 7374

密码杂凑函数值 $e=H_{256}(\overline{M})$:

AD673CBD A3114171 29A9EAA5 F9AB1AA1 633AD477 18A84DFD 46C17C6F A0AA3B12

产生随机数 k :

36CD79FC 8E24B735 7A8A7B4A 46D454C3 97703D64 98158C60 5399B341 ADA186D6

计算椭圆曲线点 $(x_1, y_1)=[k]G$:

坐标 x_1 :

00 3FD87D69 47A15F94 25B32EDD 39381ADF D5E71CD4 BB357E3C 6A6E0397 EEA7CD66

坐标 y_1 :

00 80771114 6D73951E 9EB373A6 58214054 B7B56D1D 50B4CD6E B32ED387 A65AA6A2

计算 $r=(e+x_1) \bmod n$:

6D3FBA26 EAB2A105 4F5D1983 32E33581 7C8AC453 ED26D339 1CD4439D 825BF25B

$(1+d_A)^{-1}$:

73AF2954 F951A9DF F5B4C8F7 119DAA1C 230C9BAD E60568D0 5BC3F432 1E1F4260

计算 $s=((1+d_A)^{-1} \cdot (k-r \cdot d_A)) \bmod n$:

3124C568 8D95F0A1 0252A9BE D033BEC8 4439DA38 4621B6D6 FAD77F94 B74A9556

消息 M 的签名为 (r,s) :

值 r :

6D3FBA26 EAB2A105 4F5D1983 32E33581 7C8AC453 ED26D339 1CD4439D 825BF25B

值 s :

3124C568 8D95F0A1 0252A9BE D033BEC8 4439DA38 4621B6D6 FAD77F94 B74A9556

验证各步骤中的有关值:

密码杂凑函数值 $e' = H_{256}(\overline{M}')$:

AD673CBD A3114171 29A9EAA5 F9AB1AA1 633AD477 18A84DFD 46C17C6F A0AA3B12

计算 $t=(r'+s') \bmod n$:

1E647F8F 784891A6 51AFC342 0316F44A 042D7194 4C91910F 835086C8 2CB07194

计算椭圆曲线点 $(x'_0, y'_0)=[s']G$:

坐标 x'_0 :

00 252CF6B6 3A044FCE 553EAA77 3E1E9264 44E0DAA1 0E4B8873 89D11552 EA6418F7

坐标 y'_0 :

00 776F3C5D B3A0D312 9EAE44E0 21C28667 92E4264B E1BEEBCA 3B8159DC A382653A

计算椭圆曲线点 $(x'_{00}, y'_{00})=[t]P_A$:

坐标 x'_{00} :

00 07DA3F04 0EFB9C28 1BE107EC C389F56F E76A680B B5FDEE1D D554DC11 EB477C88

坐标 y'_{00} :

01 7BA2845D C65945C3 D48926C7 0C953A1A F29CE2E1 9A7EEE6B E0269FB4 803CA68B

计算椭圆曲线点 $(x'_1, y'_1)=[s']G + [t]P_A$:

坐标 x'_1 :

00 3FD87D69 47A15F94 25B32EDD 39381ADF D5E71CD4 BB357E3C 6A6E0397 EEA7CD66

坐标 y'_1 :

00 80771114 6D73951E 9EB373A6 58214054 B7B56D1D 50B4CD6E B32ED387 A65AA6A2

计算 $R = (e' + x'_1) \bmod n$:

6D3FBA26 EAB2A105 4F5D1983 32E33581 7C8AC453 ED26D339 1CD4439D 825BF25B

SM2椭圆曲线公钥密码算法 第3部分：密钥交换协议

Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves

Part 3: Key Exchange Protocol

国家密码管理局

2010年12月

目 次

前 言	II
引 言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 符号	2
5 算法参数与辅助函数	3
5.1 总则	3
5.2 椭圆曲线系统参数	3
5.3 用户密钥对	3
5.4 辅助函数	3
5.4.1 概述	3
5.4.2 密码杂凑函数	4
5.4.3 密钥派生函数	4
5.4.4 随机数发生器	4
5.5 用户其它信息	4
6 密钥交换协议及流程	4
6.1 密钥交换协议	4
6.2 密钥交换协议流程	6
附录A (资料性附录) 密钥交换及验证示例	7
A.1 一般要求	7
A.2 F_p 上椭圆曲线密钥交换协议	7
A.3 F_{2^m} 上椭圆曲线密钥交换协议	11

前 言

《SM2椭圆曲线公钥密码算法》分为四个部分：

- 第1部分：总则
 - 第2部分：数字签名算法
 - 第3部分：密钥交换协议
 - 第4部分：公钥加密算法
- 本部分为第3部分。
本部分的附录A为资料性附录。

引 言

N.Koblitz和V.Miller在1985年各自独立地提出将椭圆曲线应用于公钥密码系统。椭圆曲线公钥密码所基于的曲线性质如下：

- 有限域上椭圆曲线在点加运算下构成有限交换群，且其阶与基域规模相近；
- 类似于有限域乘法群中的乘幂运算，椭圆曲线多倍点运算构成一个单向函数。

在多倍点运算中，已知多倍点与基点，求解倍数的问题称为椭圆曲线离散对数问题。对于一般椭圆曲线的离散对数问题，目前只存在指数级计算复杂度的求解方法。与大数分解问题及有限域上离散对数问题相比，椭圆曲线离散对数问题的求解难度要大得多。因此，在相同安全程度要求下，椭圆曲线密码较其它公钥密码所需的密钥规模要小得多。

本部分描述了基于椭圆曲线的密钥交换协议。

SM2椭圆曲线公钥密码算法

第3部分：密钥交换协议

1 范围

本部分规定了SM2椭圆曲线公钥密码算法的密钥交换协议，并给出了密钥交换与验证示例及其相应的流程。

本部分适用于商用密码应用中的密钥交换，可满足通信双方经过两次或可选三次信息传递过程，计算获取一个由双方共同决定的共享秘密密钥（会话密钥）。同时，本部分还可为安全产品生产商提供产品和技术的标准定位以及标准化的参考，提高安全产品的可信性与互操作性。

2 规范性引用文件

下列文件中的条款通过本部分引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

SM2椭圆曲线公钥密码算法 第1部分：总则

3 术语和定义

下列术语和定义适用于本部分。

3.1

密钥 **key**

确定密码函数运算的一个参数，它用于：

- a) 加密或解密变换；
- b) 同步产生共享秘密；
- c) 数字签名的生成或验证。

[ANSI X9.63-2001]

3.2

密钥交换 **key exchange**

在通信实体之间安全地建立一个共享密钥的协商过程。

3.3

密钥协商 **key agreement**

多个用户之间建立一个共享秘密密钥的过程，并且其中的任何一个用户都不能预先确定该密钥的值。

[ISO/IEC 15946-3 3.16]

3.4

从A到B的密钥确认 **key confirmation from A to B**

使用户B确信用户A拥有特定秘密密钥的保证。

[ISO/IEC 15946-3 3.17]

3.5

密钥派生函数 **key derivation function**

通过作用于共享秘密和双方都知道的其它参数，产生一个或多个共享秘密密钥的函数。

[ANSI X9.63-2001 2.1]

3.6

杂凑函数 **hash function**

将一个比特串映射为一个固定长度比特串的函数。该函数满足如下性质：

- a) 对于任意给定的输出，要找到其对应的输入，在计算上是不可行的；
- b) 对于任意给定的输入，要找到输出相同的另一个输入，在计算上是不可行的。

注：计算可行性依赖于具体的安全需求和环境。

[ISO/IEC 15946-2 3.1.3]

3.7

杂凑值 **hash value**

杂凑函数作用于一条消息时输出的比特串。

[ISO/IEC 15946-2 3.1.2]

3.8

对称密码算法 **symmetric cryptographic algorithm**

一种执行加密的算法或执行相应解密的算法，其中加密和解密使用的密钥容易从计算上相互求得。

3.9

发起方 **initiator**

在一个协议的操作过程中发送首轮交换信息的用户。

[ANSI X9.63-2001 2.1]

3.10

响应方 **responder**

在一个协议的操作过程中不是发送首轮交换信息的用户。

[ANSI X9.63-2001 2.1]

3.11

可辨别标识 **distinguishing identifier**

可以无歧义辨别某一实体身份的信息。

[ISO/IEC 15946-3 3.9]

4 符号

下列符号适用于本部分。

A, B : 使用公钥密码系统的两个用户。

a, b : F_q 中的元素，它们定义 F_q 上的一条椭圆曲线 E 。

d_A : 用户A的私钥。

d_B : 用户B的私钥。

$E(F_q)$: F_q 上椭圆曲线 E 的所有有理点(包括无穷远点 O)组成的集合。

F_q : 包含 q 个元素的有限域。

G : 椭圆曲线的一个基点，其阶为素数。

$Hash()$: 密码杂凑函数。

$H_v()$: 消息摘要长度为 v 比特的密码杂凑函数。

h : 余因子， $h = \#E(F_q)/n$ ，其中 n 是基点 G 的阶。

ID_A, ID_B : 用户A和用户B的可辨别标识。

K, K_A, K_B : 密钥交换协议商定的共享秘密密钥。

$KDF()$: 密钥派生函数。

$\text{mod } n$: 模 n 运算。例如， $23 \text{ mod } 7 = 2$ 。

n : 基点 G 的阶(n 是 $\# E(F_q)$ 的素因子)。

O : 椭圆曲线上的一个特殊点, 称为无穷远点或零点, 是椭圆曲线加法群的单位元。

P_A : 用户A的公钥。

P_B : 用户B的公钥。

q : 有限域 F_q 中元素的数目。

r_A : 密钥交换中用户A产生的临时密钥值。

r_B : 密钥交换中用户B产生的临时密钥值。

$x||y$: x 与 y 的拼接, 其中 x 、 y 可以是比特串或字节串。

Z_A : 关于用户A的可辨别标识、部分椭圆曲线系统参数和用户A公钥的杂凑值。

Z_B : 关于用户B的可辨别标识、部分椭圆曲线系统参数和用户B公钥的杂凑值。

$\#E(F_q)$: $E(F_q)$ 上的点的数目, 称为椭圆曲线 $E(F_q)$ 的阶。

$[k]P$: 椭圆曲线上点 P 的 k 倍点, 即, $[k]P = \underbrace{P + P + \dots + P}_{k \uparrow}$, k 是正整数。

x,y : 大于或等于 x 且小于或等于 y 的整数的集合。

$\lceil x \rceil$: 顶函数, 大于或等于 x 的最小整数。例如, $\lceil 7 \rceil = 7$, $\lceil 8.3 \rceil = 9$ 。

$\lfloor x \rfloor$: 底函数, 小于或等于 x 的最大整数。例如, $\lfloor 7 \rfloor = 7$, $\lfloor 8.3 \rfloor = 8$ 。

$\&$: 两个整数的按比特与运算。

5 算法参数与辅助函数

5.1 总则

密钥交换协议是两个用户A和B通过交互的信息传递, 用各自的私钥和对方的公钥来商定一个只有他们知道的秘密密钥。这个共享的秘密密钥通常用在某个对称密码算法中。该密钥交换协议能够用于密钥管理和协商。

5.2 椭圆曲线系统参数

椭圆曲线系统参数包括有限域 F_q 的规模 q (当 $q = 2^m$ 时, 还包括元素表示法的标识和约化多项式); 定义椭圆曲线 $E(F_q)$ 的方程的两个元素 a 、 $b \in F_q$; $E(F_q)$ 上的基点 $G = (x_G, y_G)$ ($G \neq O$), 其中 x_G 和 y_G 是 F_q 中的两个元素; G 的阶 n 及其它可选项(如 n 的余因子 h 等)。

椭圆曲线系统参数及其验证应符合第1部分第5章的规定。

5.3 用户密钥对

用户A的密钥对包括其私钥 d_A 和公钥 $P_A = [d_A]G = (x_A, y_A)$, 用户B的密钥对包括其私钥 d_B 和公钥 $P_B = [d_B]G = (x_B, y_B)$ 。

用户密钥对的生成算法与公钥验证算法应符合第1部分第6章的规定。

5.4 辅助函数

5.4.1 概述

在本部分规定的椭圆曲线密钥交换协议中, 涉及到三类辅助函数: 密码杂凑函数, 密钥派生函数与随机数发生器。这三类辅助函数的强弱直接影响密钥交换协议的安全性。

5.4.2 密码杂凑函数

本部分规定使用国家密码管理局批准的密码杂凑算法，如SM3密码杂凑算法。

5.4.3 密钥派生函数

密钥派生函数的作用是从一个共享的秘密比特串中派生出密钥数据。在密钥协商过程中，密钥派生函数作用在密钥交换所获共享的秘密比特串上，从中产生所需的会话密钥或进一步加密所需的密钥数据。

密钥派生函数需要调用密码杂凑函数。

设密码杂凑函数为 $H_v(\cdot)$ ，其输出是长度恰为 v 比特的杂凑值。

密钥派生函数 $KDF(Z, klen)$ ：

输入：比特串 Z ，整数 $klen$ (表示要获得的密钥数据的比特长度，要求该值小于 $(2^{32}-1)v$)。

输出：长度为 $klen$ 的密钥数据比特串 K 。

a)初始化一个32比特构成的计数器 $ct=0x00000001$ ；

b)对 i 从1到 $\lceil klen/v \rceil$ 执行：

b.1)计算 $Ha_i = H_v(Z \parallel ct)$ ；

b.2) $ct++$ ；

c)若 $klen/v$ 是整数，令 $Ha!_{\lceil klen/v \rceil} = Ha_{\lceil klen/v \rceil}$ ，否则令 $Ha!_{\lceil klen/v \rceil}$ 为 $Ha_{\lceil klen/v \rceil}$ 最左边的 $(klen - (v \times \lceil klen/v \rceil))$ 比特；

d)令 $K = Ha_1 \parallel Ha_2 \parallel \dots \parallel Ha_{\lceil klen/v \rceil - 1} \parallel Ha!_{\lceil klen/v \rceil}$ 。

5.4.4 随机数发生器

本部分规定使用国家密码管理局批准的随机数发生器。

5.5 用户其它信息

用户A具有长度为 $entlen_A$ 比特的可辨别标识 ID_A ，记 $ENTL_A$ 是由整数 $entlen_A$ 转换而成的两个字节；用户B具有长度为 $entlen_B$ 比特的可辨别标识 ID_B ，记 $ENTL_B$ 是由整数 $entlen_B$ 转换而成的两个字节。在本部分规定的椭圆曲线密钥交换协议中，参与密钥协商的A、B双方都需要用密码杂凑函数求得用户A的杂凑值 Z_A 和用户B的杂凑值 Z_B 。按本文本第1部分4.2.5和4.2.4给出的方法，将椭圆曲线方程参数 a 、 b 、 G 的坐标 x_G 、 y_G 和 P_A 的坐标 x_A 、 y_A 的数据类型转换为比特串， $Z_A = H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ ；按本文本第1部分4.2.5和4.2.4给出的方法，将椭圆曲线方程参数 a 、 b 、 G 的坐标 x_G 、 y_G 和 P_B 的坐标 x_B 、 y_B 的数据类型转换为比特串， $Z_B = H_{256}(ENTL_B \parallel ID_B \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_B \parallel y_B)$ 。

6 密钥交换协议及流程

6.1 密钥交换协议

设用户A和B协商获得密钥数据的长度为 $klen$ 比特，用户A为发起方，用户B为响应方。

用户A和B双方为了获得相同的密钥，应实现如下运算步骤：

记 $w = \lceil (\lceil \log_2(n) \rceil / 2) \rceil - 1$ 。

用户A：

A1：用随机数发生器产生随机数 $r_A \in [1, n-1]$ ；

A2：计算椭圆曲线点 $R_A = [r_A]G = (x_1, y_1)$ ；

A3：将 R_A 发送给用户B；

用户B:

B1: 用随机数发生器产生随机数 $r_B \in [1, n-1]$;

B2: 计算椭圆曲线点 $R_B = [r_B]G=(x_2, y_2)$;

B3: 从 R_B 中取出域元素 x_2 , 按本文第1部分4.2.7给出的方法将 x_2 的数据类型转换为整数, 计算 $\bar{x}_2 = 2^w + (x_2 \& (2^w - 1))$;

B4: 计算 $t_B = (d_B + \bar{x}_2 \cdot r_B) \bmod n$;

B5: 验证 R_A 是否满足椭圆曲线方程, 若不满足则协商失败; 否则从 R_A 中取出域元素 x_1 , 按本文第1部分4.2.7给出的方法将 x_1 的数据类型转换为整数, 计算 $\bar{x}_1 = 2^w + (x_1 \& (2^w - 1))$;

B6: 计算椭圆曲线点 $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$, 若 V 是无穷远点, 则B协商失败; 否则按本文第1部分4.2.5和4.2.4给出的方法将 x_V 、 y_V 的数据类型转换为比特串;

B7: 计算 $K_B = KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$;

B8: (选项)按本文第1部分4.2.5和4.2.4给出的方法将 R_A 的坐标 x_1 、 y_1 和 R_B 的坐标 x_2 、 y_2 的数据类型转换为比特串, 计算 $S_B = Hash(0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$;

B9: 将 R_B 、(选项 S_B)发送给用户A;

用户A:

A4: 从 R_A 中取出域元素 x_1 , 按本文第1部分4.2.7给出的方法将 x_1 的数据类型转换为整数, 计算 $\bar{x}_1 = 2^w + (x_1 \& (2^w - 1))$;

A5: 计算 $t_A = (d_A + \bar{x}_1 \cdot r_A) \bmod n$;

A6: 验证 R_B 是否满足椭圆曲线方程, 若不满足则协商失败; 否则从 R_B 中取出域元素 x_2 , 按本文第1部分4.2.7给出的方法将 x_2 的数据类型转换为整数, 计算 $\bar{x}_2 = 2^w + (x_2 \& (2^w - 1))$;

A7: 计算椭圆曲线点 $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$, 若 U 是无穷远点, 则A协商失败; 否则按本文第1部分4.2.5和4.2.4给出的方法将 x_U 、 y_U 的数据类型转换为比特串;

A8: 计算 $K_A = KDF(x_U \parallel y_U \parallel Z_A \parallel Z_B, klen)$;

A9: (选项)按本文第1部分4.2.5和4.2.4给出的方法将 R_A 的坐标 x_1 、 y_1 和 R_B 的坐标 x_2 、 y_2 的数据类型转换为比特串, 计算 $S_1 = Hash(0x02 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$, 并检验 $S_1 = S_B$ 是否成立, 若等式不成立则从B到A的密钥确认失败;

A10: (选项)计算 $S_A = Hash(0x03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$, 并将 S_A 发送给用户B。

用户B:

B10: (选项)计算 $S_2 = Hash(0x03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$, 并检验 $S_2 = S_A$ 是否成立, 若等式不成立则从A到B的密钥确认失败。

注: 如果 Z_A 、 Z_B 不是用户A和B所对应的杂凑值, 则自然不能达成一致的共享秘密值。密钥交换协议过程的示例参见附录A。

6.2 密钥交换协议流程

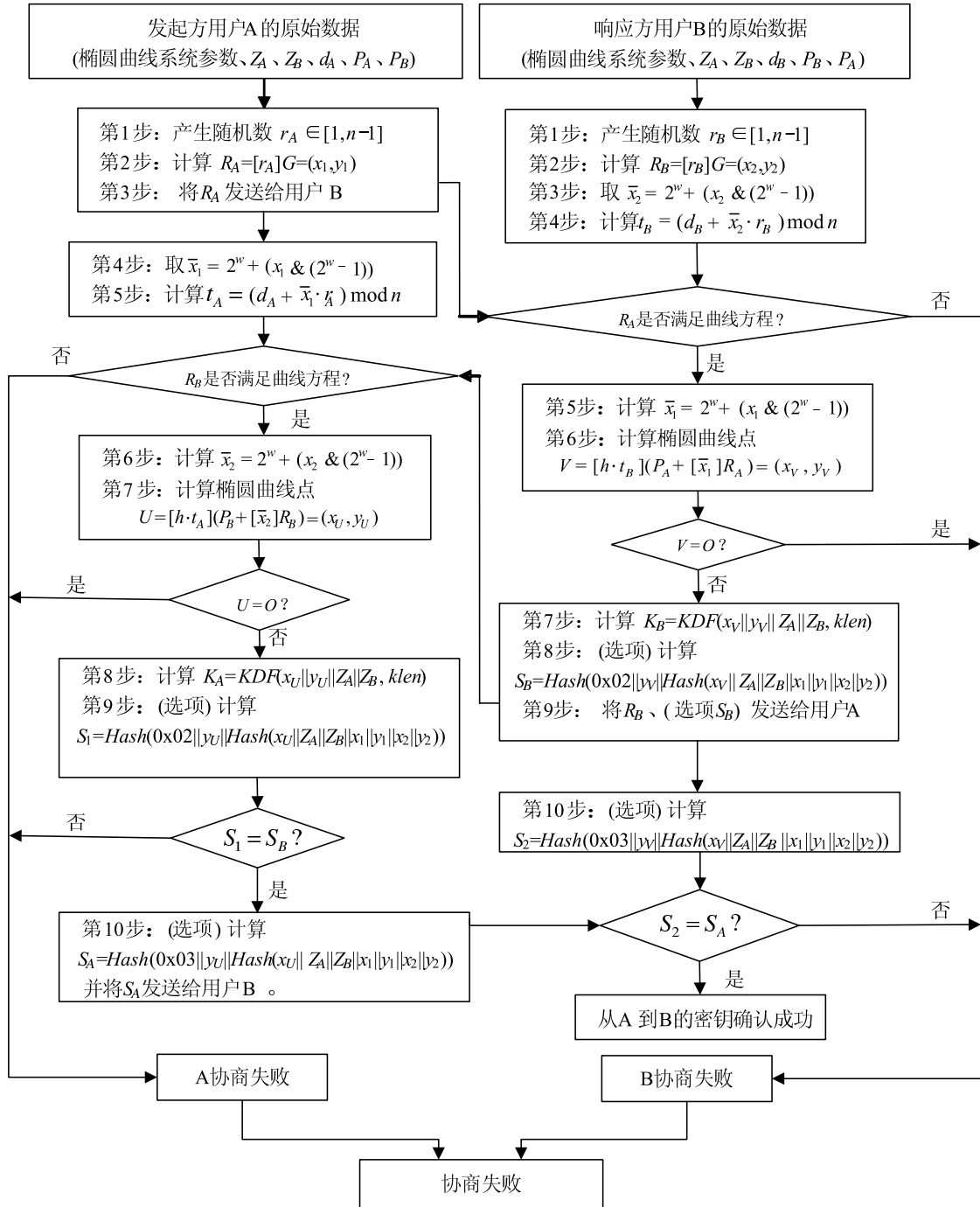


图1 密钥交换协议流程

附录A
(资料性附录)
密钥交换及验证示例

A.1 一般要求

本附录选用《SM3密码杂凑算法》给出的密码杂凑函数，其输入是长度小于 2^{64} 的消息比特串，输出是长度为256比特的杂凑值，记为 $H_{256}()$ 。

本附录中，所有用16进制表示的数，左边为高位，右边为低位。

设用户A的身份是：ALICE123@YAHOO.COM。用ASCII编码记 ID_A ：

414C 49434531 32334059 41484F4F 2E434F4D。 $ENTL_A=0090$ 。

设用户B的身份是：BILL456@YAHOO.COM。用ASCII编码记 ID_B ：

42 494C4C34 35364059 41484F4F 2E434F4D。 $ENTL_B=0088$ 。

A.2 F_p 上椭圆曲线密钥交换协议

椭圆曲线方程为： $y^2=x^3+ax+b$

示例1: F_p -256

素数 p ：

8542D69E 4C044F18 E8B92435 BF6FF7DE 45728391 5C45517D 722EDB8B 08F1DFC3

系数 a ：

787968B4 FA32C3FD 2417842E 73BBFEFF 2F3C848B 6831D7E0 EC65228B 3937E498

系数 b ：

63E4C6D3 B23B0C84 9CF84241 484BFE48 F61D59A5 B16BA06E 6E12D1DA 27C5249A

余因子 h ： 1

基点 $G=(x_G, y_G)$ ，其阶记为 n 。

坐标 x_G ：

421DEBD6 1B62EAB6 746434EB C3CC315E 32220B3B ADD50BDC 4C4E6C14 7FEDD43D

坐标 y_G ：

0680512B CBB42C07 D47349D2 153B70C4 E5D7FDFC BFA36EA1 A85841B9 E46E09A2

阶 n ：

8542D69E 4C044F18 E8B92435 BF6FF7DD 29772063 0485628D 5AE74EE7 C32E79B7

用户A的私钥 d_A ：

6FCBA2EF 9AE0AB90 2BC3BDE3 FF915D44 BA4CC78F 88E2F8E7 F8996D3B 8CCEEDEE

用户A的公钥 $P_A=(x_A, y_A)$ ：

坐标 x_A ：

3099093B F3C137D8 FCBBCDF4 A2AE50F3 B0F216C3 122D7942 5FE03A45 DBFE1655

坐标 y_A ：

3DF79E8D AC1CF0EC BAA2F2B4 9D51A4B3 87F2EFAF 48233908 6A27A8E0 5BAED98B

用户B的私钥 d_B ：

5E35D7D3 F3C54DBA C72E6181 9E730B01 9A84208C A3A35E4C 2E353DFC CB2A3B53

用户B的公钥 $P_B=(x_B, y_B)$ ：

坐标 x_B ：

245493D4 46C38D8C C0F11837 4690E7DF 633A8A4B FB3329B5 ECE604B2 B4F37F43

坐标 y_B ：

53C0869F 4B9E1777 3DE68FEC 45E14904 E0DEA45B F6CECF99 18C85EA0 47C60A4C
 杂凑值 $Z_A = H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ 。

Z_A :
 E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
 杂凑值 $Z_B = H_{256}(ENTL_B \parallel ID_B \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_B \parallel y_B)$ 。

Z_B :
 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

密钥交换A1-A3步骤中的有关值:
 产生随机数 r_A :
 83A2C9C8 B96E5AF7 0BD480B4 72409A9A 327257F1 EBB73F5B 073354B2 48668563
 计算椭圆曲线点 $R_A = [r_A]G = (x_1, y_1)$:
 坐标 x_1 :
 6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0
 坐标 y_1 :
 0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A

密钥交换B1-B9步骤中的有关值:
 产生随机数 r_B :
 33FE2194 0342161C 55619C4A 0C060293 D543C80A F19748CE 176D8347 7DE71C80
 计算椭圆曲线点 $R_B = [r_B]G = (x_2, y_2)$:
 坐标 x_2 :
 1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5
 坐标 y_2 :
 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4
 取 $\bar{x}_2 = 2^{127} + (x_2 \& (2^{127} - 1))$: B8F2B533 7B3DCF45 14E8BBC1 9D900EE5
 计算 $t_B = (d_B + \bar{x}_2 \cdot r_B) \bmod n$:
 2B2E11CB F03641FC 3D939262 FC0B652A 70ACAA25 B5369AD3 8B375C02 65490C9F
 取 $\bar{x}_1 = 2^{127} + (x_1 \& (2^{127} - 1))$: E856C095 05324A6D 23150C40 8F162BF0
 计算椭圆曲线点 $[\bar{x}_1]R_A = (x_{A0}, y_{A0})$:
 坐标 x_{A0} :
 2079015F 1A2A3C13 2B67CA90 75BB2803 1D6F2239 8DD8331E 72529555 204B495B
 坐标 y_{A0} :
 6B3FE6FB 0F5D5664 DCA16128 B5E7FCFD AFA5456C 1E5A914D 1300DB61 F37888ED
 计算椭圆曲线点 $P_A + [\bar{x}_1]R_A = (x_{A1}, y_{A1})$:
 坐标 x_{A1} :
 1C006A3B FF97C651 B7F70D0D E0FC09D2 3AA2BE7A 8E9FF7DA F32673B4 16349B92
 坐标 y_{A1} :
 5DC74F8A CC114FC6 F1A75CB2 86864F34 7F9B2CF2 9326A270 79B7D37A FC1C145B
 计算 $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$:
 坐标 x_V :
 47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
 坐标 y_V :
 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
 计算 $K_B = KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$:
 $x_V \parallel y_V \parallel Z_A \parallel Z_B$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

$klen = 128$

共享密钥 K_B : 55B0AC62 A6B927BA 23703832 C853DED4

计算选项 $S_B = Hash(0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67
6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0
0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A
1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5
54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

$Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

02 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

选项 S_B :

284C8F19 8F141B50 2E81250F 1581C7E9 EEB4CA69 90F9E02D F388B454 71F5BC5C

密钥交换A4-A10步骤中的有关值:

取 $\bar{x}_1 = 2^{127} + (x_1 \& (2^{127} - 1))$: E856C095 05324A6D 23150C40 8F162BF0

计算 $t_A = (d_A + \bar{x}_1 \cdot r_A) \bmod n$:

236CF0C7 A177C65C 7D55E12D 361F7A6C 174A7869 8AC099C0 874AD065 8A4743DC

取 $\bar{x}_2 = 2^{127} + (x_2 \& (2^{127} - 1))$: B8F2B533 7B3DCF45 14E8BBC1 9D900EE5

计算椭圆曲线点 $[\bar{x}_2]R_B = (x_{B0}, y_{B0})$:

坐标 x_{B0} :

66864274 6BFC066A 1E731ECF FF51131B DC81CF60 9701CB8C 657B25BF 55B7015D

坐标 y_{B0} :

1988A7C6 81CE1B50 9AC69F49 D72AE60E 8B71DB6C E087AF84 99FEEF4C CD523064

计算椭圆曲线点 $P_B + [\bar{x}_2]R_B = (x_{B1}, y_{B1})$:

坐标 x_{B1} :

7D2B4435 10886AD7 CA3911CF 2019EC07 078AFF11 6E0FC409 A9F75A39 01F306CD

坐标 y_{B1} :

331F0C6C 0FE08D40 5FFEDB30 7BC255D6 8198653B DCA68B9C BA100E73 197E5D24

计算 $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$:

坐标 x_U :

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905

坐标 y_U :

2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295

计算 $K_A = KDF(x_U \parallel y_U \parallel Z_A \parallel Z_B, klen)$:

$x_U \parallel y_U \parallel Z_A \parallel Z_B$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

$klen=128$

共享密钥 K_A : 55B0AC62 A6B927BA 23703832 C853DED4

计算选项 $S_1=Hash(0x02 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2))$:

$x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2$

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67
6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0
0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A
1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5
54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

$Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0x02 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

02 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

选项 S_1 :

284C8F19 8F141B50 2E81250F 1581C7E9 EEB4CA69 90F9E02D F388B454 71F5BC5C

计算选项 $S_A=Hash(0x03 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2))$:

$x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67
6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0
0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A
1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5
54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

$Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0x03 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

03 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

选项 S_A :

23444DAF 8ED75343 66CB901C 84B3BDBB 63504F40 65C1116C 91A4C006 97E6CF7A

密钥交换B10步骤中的有关值:

计算选项 $S_2=Hash(0x03 || y_V || Hash(x_V || Z_A || Z_B || x_1 || y_1 || x_2 || y_2))$:

$x_V || Z_A || Z_B || x_1 || y_1 || x_2 || y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905
E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31
6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0
 0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A
 1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5
 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4
Hash($x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$):
 FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647
0x03 $\parallel y_V \parallel \text{Hash}(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:
 03 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295
 FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647
 选项 S_2 :
 23444DAF 8ED75343 66CB901C 84B3BDBB 63504F40 65C1116C 91A4C006 97E6CF7A

A.3 F_{2^m} 上椭圆曲线密钥交换协议

椭圆曲线方程为: $y^2 + xy = x^3 + ax^2 + b$

示例2: F_{2^m} -257

基域生成多项式: $x^{257} + x^{12} + 1$

系数 a : 0

系数 b :

00 E78BCD09 746C2023 78A7E72B 12BCE002 66B9627E CB0B5A25 367AD1AD 4CC6242B

余因子 h : 4

基点 $G=(x_G, y_G)$, 其阶记为 n 。

坐标 x_G :

00 CDB9CA7F 1E6B0441 F658343F 4B10297C 0EF9B649 1082400A 62E7A748 5735FADD

坐标 y_G :

01 3DE74DA6 5951C4D7 6DC89220 D5F7777A 611B1C38 BAE260B1 75951DC8 060C2B3E

阶 n :

7FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF BC972CF7 E6B6F900 945B3C6A 0CF6161D

用户A的私钥 d_A :

4813903D 254F2C20 A94BC570 42384969 54BB5279 F861952E F2C5298E 84D2CEAA

用户A的公钥 $P_A = (x_A, y_A)$:

坐标 x_A :

00 8E3BDB2E 11F91933 88F1F901 CCC857BF 49CFC065 FB38B906 9CAAE6D5 AFC3592F

坐标 y_A :

00 4555122A AC0075F4 2E0A8BBD 2C0665C7 89120DF1 9D77B4E3 EE4712F5 98040415

用户B的私钥 d_B :

08F41BAE 0922F47C 212803FE 681AD52B 9BF28A35 E1CD0EC2 73A2CF81 3E8FD1DC

用户B的公钥 $P_B = (x_B, y_B)$:

坐标 x_B :

00 34297DD8 3AB14D5B 393B6712 F32B2F2E 938D4690 B095424B 89DA880C 52D4A7D9

坐标 y_B :

01 99BBF11A C95A0EA3 4BBD00CA 50B93EC2 4ACB6833 5D20BA5D CFE3B33B DBD2B62D

杂凑值 $Z_A = H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ 。

Z_A :

ECF00802 15977B2E 5D6D61B9 8A99442F 03E8803D C39E349F 8DCA5621 A9ACDF2B

杂凑值 $Z_B=H_{256}(ENTL_B \parallel ID_B \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_B \parallel y_B)$ 。

Z_B :

557BAD30 E183559A EEC3B225 6E1C7C11 F870D22B 165D015A CF9465B0 9B87B527

密钥交换A1-A3步骤中的有关值:

产生随机数 r_A :

54A3D667 3FF3A6BD 6B02EBB1 64C2A3AF 6D4A4906 229D9BFC E68CC366 A2E64BA4

计算椭圆曲线点 $R_A=[r_A]G=(x_1, y_1)$:

坐标 x_1 :

01 81076543 ED19058C 38B313D7 39921D46 B80094D9 61A13673 D4A5CF8C 7159E304

坐标 y_1 :

01 D8CFFF7C A27A01A2 E88C1867 3748FDE9 A74C1F9B 45646ECA 0997293C 15C34DD8

密钥交换B1-B9步骤中的有关值:

产生随机数 r_B :

1F219333 87BEF781 D0A8F7FD 708C5AE0 A56EE3F4 23DBC2FE 5BDF6F06 8C53F7AD

计算椭圆曲线点 $R_B=[r_B]G=(x_2, y_2)$:

坐标 x_2 :

00 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46 8D322A2A

坐标 y_2 :

00 16599BB5 2ED9EAFD D01CFA45 3CF3052E D60184D2 EECFD42B 52DB7411 0B984C23

取 $\bar{x}_2 = 2^{127} + (x_2 \& (2^{127} - 1))$: E8CC43FF A5F2623B 9BD04E46 8D322A2A

计算 $t_B = (d_B + \bar{x}_2 \cdot r_B) \bmod n$:

3D51D331 14A453A0 5791DB63 5B45F8DB C54686D7 E2212D49 E4A717C6 B10DEDB0

计算 $h \cdot t_B \bmod n$:

75474CC4 52914E81 5E476D8D 6D17E36F 5882EE67 A1CDBC26 FE4122B0 B741A0A3

取 $\bar{x}_1 = 2^{127} + (x_1 \& (2^{127} - 1))$: B80094D9 61A13673 D4A5CF8C 7159E304

计算椭圆曲线点 $[\bar{x}_1]R_A=(x_{A0}, y_{A0})$:

坐标 x_{A0} :

01 98AB5F14 349B6A46 F77FBFCB DDBFCD34 320DC1F4 C546D13C 3A9F0E83 0C39B579

坐标 y_{A0} :

00 BFB49224 ACCE2E51 04CD4519 C0CBE3AD 0C19BF11 805BE108 59069AA6 9317A2B7

计算椭圆曲线点 $P_A + [\bar{x}_1]R_A=(x_{A1}, y_{A1})$:

坐标 x_{A1} :

00 24A92F64 66A37C5C 12A2C68D 58BFB0F0 32F2B976 60957CB0 5E63F961 F160FE57

坐标 y_{A1} :

00 F74A4F17 DC560A55 FDE0F1AB 168BCBF7 6502E240 BA2D6BD6 BE6E5D79 16B288FC

计算 $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$:

坐标 x_V :

00 DADD0874 06221D65 7BC3FA79 FF329BB0 22E9CB7D DFCFCCFE 277BE8CD 4AE9B954

坐标 y_V :

01 F0464B1E 81684E5E D6EF281B 55624EF4 6CAA3B2D 37484372 D91610B6 98252CC9

计算 $K_B=KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$:

$x_V \parallel y_V \parallel Z_A \parallel Z_B$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9

5401F046 4B1E8168 4E5ED6EF 281B5562 4EF46CAA 3B2D3748 4372D916 10B69825

2CC9ECF0 08021597 7B2E5D6D 61B98A99 442F03E8 803DC39E 349F8DCA 5621A9AC
DF2B557B AD30E183 559AEEC3 B2256E1C 7C11F870 D22B165D 015ACF94 65B09B87
B527

$klen=128$

共享密钥 K_B : 4E587E5C 66634F22 D973A7D9 8BF8BE23

计算选项 $S_B = Hash(0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9
54ECF008 0215977B 2E5D6D61 B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF
2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01 5ACF9465 B09B87B5
27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159
E30401D8 CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15
C34DD800 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46
8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4 2B52DB74
110B984C 23

$Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

02 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C
C9E05FE2 87B73B0C E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

选项 S_B :

4EB47D28 AD3906D6 244D01E0 F6AEC73B 0B51DE15 74C13798 184E4833 DBAE295A

密钥交换A4-A10步骤中的有关值:

取 $\bar{x}_1 = 2^{127} + (x_1 \& (2^{127} - 1))$: B80094D9 61A13673 D4A5CF8C 7159E304

计算 $t_A = (d_A + \bar{x}_1 \cdot r_A) \bmod n$:

18A1C649 B94044DF 16DC8634 993F1A4A EE3F6426 DFE14AC1 3644306A A5A94187

计算 $h \cdot t_A \bmod n$:

62871926 E501137C 5B7218D2 64FC692B B8FD909B 7F852B04 D910C1AA 96A5061C

取 $\bar{x}_2 = 2^{127} + (x_2 \& (2^{127} - 1))$: E8CC43FF A5F2623B 9BD04E46 8D322A2A

计算椭圆曲线点 $[\bar{x}_2]R_B = (x_{B0}, y_{B0})$:

坐标 x_{B0} :

01 0AA3BAC9 7786B629 22F93414 57AC64F7 2552AA15 D9321677 A10C7021 33B16735

坐标 y_{B0} :

00 C10837F4 8F53C46B 714BCFBF AA1AD627 11FCB03C 0C25B366 BF176A2D C7B8E62E

计算椭圆曲线点 $P_B + [\bar{x}_2]R_B = (x_{B1}, y_{B1})$:

坐标 x_{B1} :

00 C7A446E1 98DB4278 60C3BB50 ED2197DE B8161973 9141CA61 03745035 9FAD9A99

坐标 y_{B1} :

00 602E5A42 17427EAB C5E3917D E81BFFA1 D806591A F949DD7C 97EF90FD 4CF0A42D

计算 $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$:

坐标 x_U :

00 DADD0874 06221D65 7BC3FA79 FF329BB0 22E9CB7D DFCFCFFE 277BE8CD 4AE9B954

坐标 y_U :

01 F0464B1E 81684E5E D6EF281B 55624EF4 6CAA3B2D 37484372 D91610B6 98252CC9

计算 $K_A=KDF(x_U || y_U || Z_A || Z_B, klen)$:

$x_U || y_U || Z_A || Z_B$

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9
5401F046 4B1E8168 4E5ED6EF 281B5562 4EF46CAA 3B2D3748 4372D916 10B69825
2CC9ECF0 08021597 7B2E5D6D 61B98A99 442F03E8 803DC39E 349F8DCA 5621A9AC
DF2B557B AD30E183 559AEEC3 B2256E1C 7C11F870 D22B165D 015ACF94 65B09B87
B527

$klen=128$

共享密钥 K_A : 4E587E5C 66634F22 D973A7D9 8BF8BE23

计算选项 $S_1=Hash(0x02 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2))$:

$x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2$

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9
54ECF008 0215977B 2E5D6D61 B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF
2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01 5ACF9465 B09B87B5
27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159
E30401D8 CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15
C34DD800 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46
8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4 2B52DB74
110B984C 23

$Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0x02 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

02 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C
C9E05FE2 87B73B0C E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

选项 S_1 :

4EB47D28 AD3906D6 244D01E0 F6AEC73B 0B51DE15 74C13798 184E4833 DBAE295A

计算选项 $S_A=Hash(0x03 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2))$:

$x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9
54ECF008 0215977B 2E5D6D61 B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF
2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01 5ACF9465 B09B87B5
27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159
E30401D8 CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15
C34DD800 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46
8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4 2B52DB74
110B984C 23

$Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0x03 || y_U || Hash(x_U || Z_A || Z_B || x_1 || y_1 || x_2 || y_2)$:

03 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C
C9E05FE2 87B73B0C E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

选项 S_A :

588AA670 64F24DC2 7CCAA1FA B7E27DFE 811D500A D7EF2FB8 F69DDF48 CC0FECB7

密钥交换B10步骤中的有关值:

计算选项 $S_2 = Hash(0x03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9
54ECF008 0215977B 2E5D6D61 B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF
2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01 5ACF9465 B09B87B5
27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159
E30401D8 CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15
C34DD800 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46
8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4 2B52DB74
110B984C 23

$Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0x03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

03 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C
C9E05FE2 87B73B0C E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

选项 S_2 :

588AA670 64F24DC2 7CCAA1FA B7E27DFF 811D500A D7EF2FB8 F69DDF48 CC0FECB7

SM2椭圆曲线公钥密码算法 第4部分：公钥加密算法

Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves

Part 4: Public Key Encryption Algorithm

国家密码管理局

2010年12月

目 次

前 言	II
引 言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 符号	2
5 算法参数与辅助函数	3
5.1 总则	3
5.2 椭圆曲线系统参数	3
5.3 用户密钥对	3
5.4 辅助函数	3
5.4.1 概述	3
5.4.2 密码杂凑函数	3
5.4.3 密钥派生函数	4
5.4.4 随机数发生器	4
6 加密算法及流程	4
6.1 加密算法	4
6.2 加密算法流程	5
7 解密算法及流程	5
7.1 解密算法	5
7.2 解密算法流程	6
附录A (资料性附录) 消息加解密示例	7
A.1 一般要求	7
A.2 F_p 上椭圆曲线消息加解密	7
A.3 F_{2^m} 上椭圆曲线消息加解密	8

前 言

《SM2椭圆曲线公钥密码算法》分为四个部分：

- 第1部分：总则
 - 第2部分：数字签名算法
 - 第3部分：密钥交换协议
 - 第4部分：公钥加密算法
- 本部分为第4部分。
本部分的附录A为资料性附录。

引言

N.Koblitz和V.Miller在1985年各自独立地提出将椭圆曲线应用于公钥密码系统。椭圆曲线公钥密码所基于的曲线性质如下:

- 有限域上椭圆曲线在点加运算下构成有限交换群,且其阶与基域规模相近;
- 类似于有限域乘法群中的乘幂运算,椭圆曲线多倍点运算构成一个单向函数。

在多倍点运算中,已知多倍点与基点,求解倍数的问题称为椭圆曲线离散对数问题。对于一般椭圆曲线的离散对数问题,目前只存在指数级计算复杂度的求解方法。与大数分解问题及有限域上离散对数问题相比,椭圆曲线离散对数问题的求解难度要大得多。因此,在相同安全程度要求下,椭圆曲线密码较其它公钥密码所需的密钥规模要小得多。

本部分描述了基于椭圆曲线的公钥加密算法。

SM2椭圆曲线公钥密码算法

第4部分：公钥加密算法

1 范围

本部分规定了SM2椭圆曲线公钥密码算法的公钥加密算法，并给出了消息加解密示例和相应的流程。

本部分适用于国家商用密码应用中的消息加解密，消息发送者可以利用接收者的公钥对消息进行加密，接收者用对应的私钥进行解密，获取消息。同时，本部分还可为安全产品生产商提供产品和技术标准定位以及标准化的参考，提高安全产品的可信性与互操作性。

2 规范性引用文件

下列文件中的条款通过本部分引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

SM2椭圆曲线公钥密码算法 第1部分：总则

3 术语和定义

下列术语和定义适用于本部分。

3.1

非对称密码算法 asymmetric cryptographic algorithm

使用公钥进行加密而使用私钥进行解密的一类密码算法，已知公钥求私钥在计算上不可行。

[ANSI X9.63-2001]

3.2

密钥 key

确定密码函数运算的一个参数，它用于：

- a) 加密或解密变换；
- b) 同步产生共享秘密；
- c) 数字签名的生成或验证。

[ANSI X9.63-2001]

3.3

私钥 private key

在非对称密码体制中，实体自己保持的、只有实体自身知道的一种密钥。

[ISO/IEC 15946-3 3.24]

3.4

公钥 public key

在非对称密码体制中，实体的一种可以公开的密钥。

[ISO/IEC 15946-3 3.25]

3.5

秘密密钥 secret key

在密码体制中收发双方共同拥有的、而第三方不知道的一种密钥。

[ISO/IEC 15946-3 3.26]

3.6

消息 message

任意有限长度的比特串。

[ISO/IEC 15946-4 3.7]

3.7

明文 plaintext

无需利用密码技术即可得出语义内容的数据。

[GB/T 5271.8-2001 03.07]

3.8

密文 ciphertext

利用加密技术产生的数据，若不使用密码技术，则得不到其语义内容。

[GB/T 5271.8-2001 03.08]

3.9

加密 encipherment

为了产生密文，即隐藏数据的信息内容，由密码算法对数据进行(可逆)变换。

[GB/T 15843.1-1999 3.3.10]

3.10

解密 decipherment

加密对应的逆过程。

[GB/T 15843.1-1999 3.3.8]

3.11

杂凑函数 hash function

将一个比特串映射为一个固定长度比特串的函数。该函数满足如下性质：

- a) 对于任意给定的输出，要找到其对应的输入，在计算上是不可行的；
- b) 对于任意给定的输入，要找到输出相同的另一个输入，在计算上是不可行的。

注：计算可行性依赖于具体的安全需求和环境。

[ISO/IEC 15946-2 3.1.3]

3.12

杂凑值 hash value

杂凑函数作用于一条消息时输出的比特串。

[ISO/IEC 15946-2 3.1.2]

3.13

密钥派生函数 key derivation function

通过作用于共享秘密和双方都知道的其它参数，产生一个或多个共享秘密密钥的函数。

[ANSI X9.63-2001]

4 符号

下列符号适用于本部分。

A,B: 使用公钥密码系统的两个用户。

a, b : F_q 中的元素，它们定义 F_q 上的一条椭圆曲线 E 。

d_B : 用户B的私钥。

$E(F_q)$: F_q 上椭圆曲线 E 的所有有理点(包括无穷远点 O)组成的集合。

F_q : 包含 q 个元素的有限域。

G : 椭圆曲线的一个基点，其阶为素数。

$Hash()$: 密码杂凑函数。

$H_v()$: 消息摘要长度为 v 比特的密码杂凑函数。

$KDF()$: 密钥派生函数。

M : 待加密的消息。

M' : 解密得到的消息。

n : 基点 G 的阶(n 是 $\#E(F_q)$ 的素因子)。

O : 椭圆曲线上的一个特殊点, 称为无穷远点或零点, 是椭圆曲线加法群的单位元。

P_B : 用户 B 的公钥。

q : 有限域 F_q 中元素的数目。

$x||y$: x 与 y 的拼接, 其中 x 、 y 可以是比特串或字节串。

$[k]P$: 椭圆曲线上点 P 的 k 倍点, 即, $[k]P = \underbrace{P + P + \cdots + P}_{k\text{个}}$, k 是正整数。

$[x,y]$: 大于或等于 x 且小于或等于 y 的整数的集合。

$\lceil x \rceil$: 顶函数, 大于或等于 x 的最小整数。例如 $\lceil 7 \rceil = 7$, $\lceil 8.3 \rceil = 9$ 。

$\lfloor x \rfloor$: 底函数, 小于或等于 x 的最大整数。例如 $\lfloor 7 \rfloor = 7$, $\lfloor 8.3 \rfloor = 8$ 。

$\#E(F_q)$: $E(F_q)$ 上点的数目, 称为椭圆曲线 $E(F_q)$ 的阶。

5 算法参数与辅助函数

5.1 总则

公钥加密算法规定发送者用接收者的公钥将消息加密成密文, 接收者用自己的私钥对收到的密文进行解密还原成原始消息。

5.2 椭圆曲线系统参数

椭圆曲线系统参数包括有限域 F_q 的规模 q (当 $q=2^m$ 时, 还包括元素表示法的标识和约化多项式); 定义椭圆曲线 $E(F_q)$ 的方程的两个元素 $a, b \in F_q$; $E(F_q)$ 上的基点 $G=(x_G, y_G)(G \neq O)$, 其中 x_G 和 y_G 是 F_q 中的两个元素; G 的阶 n 及其它可选项(如 n 的余因子 h 等)。

椭圆曲线系统参数及其验证应符合本文本第1部分第5章的规定。

5.3 用户密钥对

用户 B 的密钥对包括其私钥 d_B 和公钥 $P_B=[d_B]G$ 。

用户密钥对的生成算法与公钥验证算法应符合本文本第1部分第6章的规定。

5.4 辅助函数

5.4.1 概述

本部分规定的椭圆曲线公钥加密算法涉及到三类辅助函数: 密码杂凑函数、密钥派生函数和随机数发生器。这三类辅助函数的强弱直接影响加密算法的安全性。

5.4.2 密码杂凑函数

本部分规定使用国家密码管理局批准的密码杂凑算法, 如SM3密码杂凑算法。

5.4.3 密钥派生函数

密钥派生函数的作用是从一个共享的秘密比特串中派生出密钥数据。在密钥协商过程中，密钥派生函数作用在密钥交换所获共享的秘密比特串上，从中产生所需的会话密钥或进一步加密所需的密钥数据。

密钥派生函数需要调用密码杂凑函数。

设密码杂凑函数为 $H_v()$ ，其输出是长度恰为 v 比特的杂凑值。

密钥派生函数 $KDF(Z, klen)$ ：

输入：比特串 Z ，整数 $klen$ (表示要获得的密钥数据的比特长度，要求该值小于 $(2^{32}-1)v$)。

输出：长度为 $klen$ 的密钥数据比特串 K 。

- a) 初始化一个32比特构成的计数器 $ct=0x00000001$ ；
- b) 对 i 从1到 $\lceil klen/v \rceil$ 执行：
 - b.1) 计算 $H_{a_i}=H_v(Z \parallel ct)$ ；
 - b.2) $ct++$ ；
- c) 若 $klen/v$ 是整数，令 $Ha!_{\lceil klen/v \rceil} = Ha_{\lceil klen/v \rceil}$ ，否则令 $Ha!_{\lceil klen/v \rceil}$ 为 $Ha_{\lceil klen/v \rceil}$ 最左边的 $(klen - (v \times \lceil klen/v \rceil))$ 比特；
- d) 令 $K = Ha_1 \parallel Ha_2 \parallel \dots \parallel Ha_{\lceil klen/v \rceil - 1} \parallel Ha!_{\lceil klen/v \rceil}$ 。

5.4.4 随机数发生器

本部分规定使用国家密码管理局批准的随机数发生器。

6 加密算法及流程

6.1 加密算法

设需要发送的消息为比特串 M ， $klen$ 为 M 的比特长度。

为了对明文 M 进行加密，作为加密者的用户A应实现以下运算步骤：

A1：用随机数发生器产生随机数 $k \in [1, n-1]$ ；

A2：计算椭圆曲线点 $C_1 = [k]G = (x_1, y_1)$ ，按本文本第1部分4.2.8和4.2.4给出的细节，将 C_1 的数据类型转换为比特串；

A3：计算椭圆曲线点 $S = [h]P_B$ ，若 S 是无穷远点，则报错并退出；

A4：计算椭圆曲线点 $[k]P_B = (x_2, y_2)$ ，按本文本第1部分4.2.5和4.2.4给出的细节，将坐标 x_2 、 y_2 的数据类型转换为比特串；

A5：计算 $t = KDF(x_2 \parallel y_2, klen)$ ，若 t 为全0比特串，则返回A1；

A6：计算 $C_2 = M \oplus t$ ；

A7：计算 $C_3 = Hash(x_2 \parallel M \parallel y_2)$ ；

A8：输出密文 $C = C_1 \parallel C_2 \parallel C_3$ 。

注：加密过程的示例参见附录A。

6.2 加密算法流程

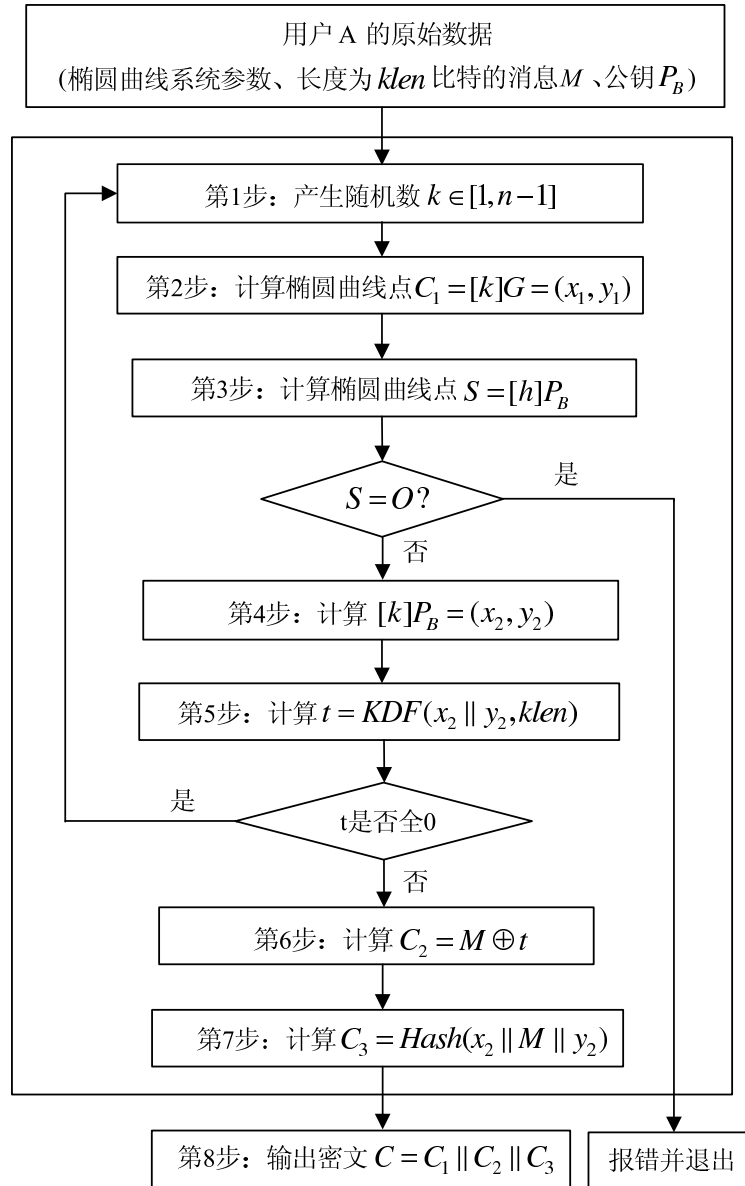


图1 加密算法流程

7 解密算法及流程

7.1 解密算法

设 $klen$ 为密文中 C_2 的比特长度。

为了对密文 $C = C_1 || C_2 || C_3$ 进行解密，作为解密者的用户 B 应实现以下运算步骤：

B1: 从 C 中取出比特串 C_1 ，按本文本第1部分4.2.3和4.2.9给出的细节，将 C_1 的数据类型转换为椭圆曲线上的点，验证 C_1 是否满足椭圆曲线方程，若不满足则报错并退出；

B2: 计算椭圆曲线点 $S = [h]C_1$ ，若 S 是无穷远点，则报错并退出；

B3: 计算 $[d_B]C_1 = (x_2, y_2)$ ，按本文本第1部分4.2.5和4.2.4给出的细节，将坐标 x_2 、 y_2 的数据类型转换为比特串；

B4: 计算 $t = KDF(x_2 || y_2, klen)$ ，若 t 为全0比特串，则报错并退出；

B5: 从 C 中取出比特串 C_2 ，计算 $M' = C_2 \oplus t$ ；

B6: 计算 $u = Hash(x_2 || M' || y_2)$ ，从 C 中取出比特串 C_3 ，若 $u \neq C_3$ ，则报错并退出；

B7: 输出明文 M' 。

注：解密过程的示例参见附录A。

7.2 解密算法流程

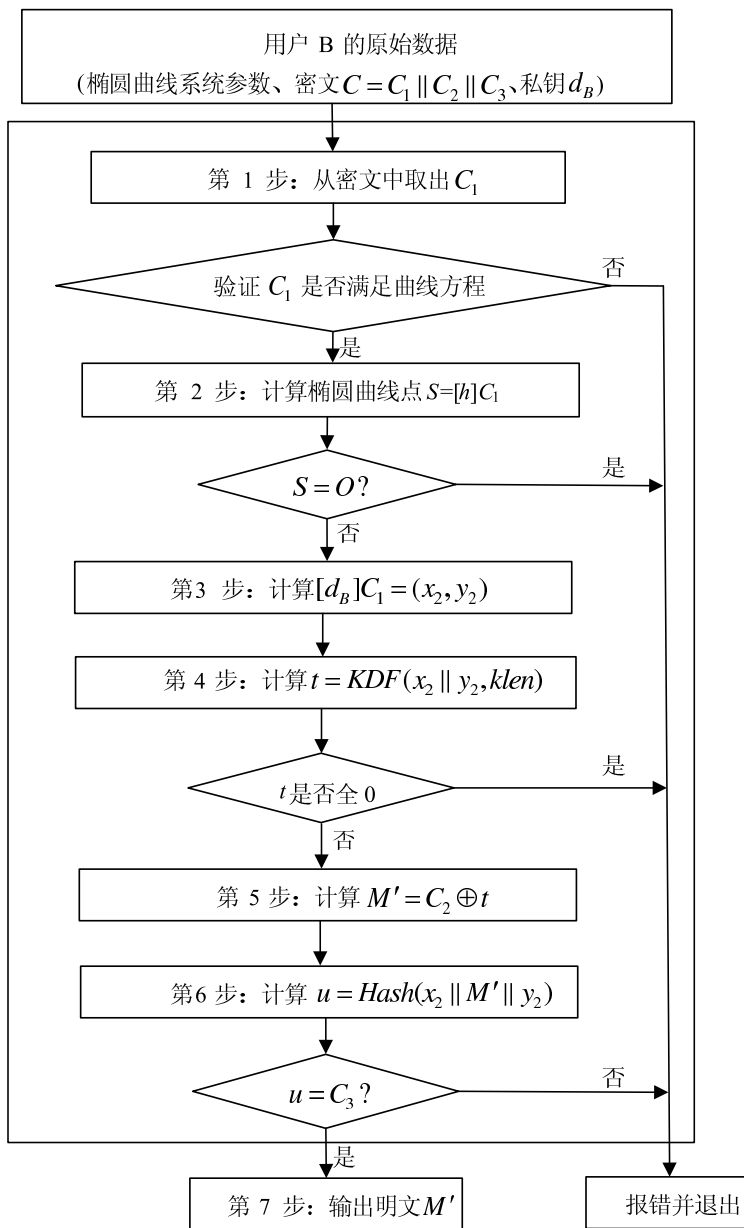


图2 解密算法流程

附 录 A
(资料性附录)
消息加解密示例

A.1 一般要求

本附录选用《SM3密码杂凑算法》给出的密码杂凑函数，其输入是长度小于 2^{64} 的消息比特串，输出是长度为256比特的杂凑值，记为 $H_{256}()$ 。

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，明文采用ASCII编码。

A.2 F_p 上椭圆曲线消息加解密

椭圆曲线方程为： $y^2 = x^3 + ax + b$

示例1: F_p -256

素数 p :

8542D69E 4C044F18 E8B92435 BF6FF7DE 45728391 5C45517D 722EDB8B 08F1DFC3

系数 a :

787968B4 FA32C3FD 2417842E 73BBFEFF 2F3C848B 6831D7E0 EC65228B 3937E498

系数 b :

63E4C6D3 B23B0C84 9CF84241 484BFE48 F61D59A5 B16BA06E 6E12D1DA 27C5249A

基点 $G=(x_G, y_G)$ ，其阶记为 n 。

坐标 x_G :

421DEBD6 1B62EAB6 746434EB C3CC315E 32220B3B ADD50BDC 4C4E6C14 7FEDD43D

坐标 y_G :

0680512B CBB42C07 D47349D2 153B70C4 E5D7FD7C BFA36EA1 A85841B9 E46E09A2

阶 n :

8542D69E 4C044F18 E8B92435 BF6FF7DD 29772063 0485628D 5AE74EE7 C32E79B7

待加密的消息 M : encryption standard

消息 M 的16进制表示: 656E63 72797074 696F6E20 7374616E 64617264

私钥 d_B :

1649AB77 A00637BD 5E2EFE28 3FBF3535 34AA7F7C B89463F2 08DDBC29 20BB0DA0

公钥 $P_B=(x_B, y_B)$:

坐标 x_B :

435B39CC A8F3B508 C1488AFC 67BE491A 0F7BA07E 581A0E48 49A5CF70 628A7E0A

坐标 y_B :

75DDBA78 F15FEECB 4C7895E2 C1CDF5FE 01DEBB2C DBADF453 99CCF77B BA076A42

加密各步骤中的有关值:

产生随机数 k :

4C62EEFD 6ECFC2B9 5B92FD6C 3D957514 8AFA1742 5546D490 18E5388D 49DD7B4F

计算椭圆曲线点 $C_1=[k]G=(x_1, y_1)$:

坐标 x_1 :

245C26FB 68B1DDDD B12C4B6B F9F2B6D5 FE60A383 B0D18D1C 4144ABF1 7F6252E7

坐标 y_1 :

76CB9264 C2A7E88E 52B19903 FDC47378 F605E368 11F5C074 23A24B84 400F01B8

在此 C_1 选用未压缩的表示形式，点转换成字节串的形式为 $PC\|x_1\|y_1$ ，其中 PC 为单一字节且 $PC=04$ ，仍记为 C_1 。

计算椭圆曲线点 $[k]P_B=(x_2, y_2)$:

坐标 x_2 :

64D20D27 D0632957 F8028C1E 024F6B02 EDF23102 A566C932 AE8BD613 A8E865FE

坐标 y_2 :

58D225EC A784AE30 0A81A2D4 8281A828 E1CEDF11 C4219099 84026537 5077BF78

消息 M 的比特长度 $klen=152$

计算 $t=KDF(x_2\|y_2, klen)$:

006E30 DAE231B0 71DFAD8A A379E902 64491603

计算 $C_2=M\oplus t$:

650053 A89B41C4 18B0C3AA D00D886C 00286467

计算 $C_3=Hash(x_2\|M\|y_2)$:

$x_2\|M\|y_2$:

64D20D27 D0632957 F8028C1E 024F6B02 EDF23102 A566C932 AE8BD613 A8E865FE

656E6372 79707469 6F6E2073 74616E64 61726458 D225ECA7 84AE300A 81A2D482

81A828E1 CEDF11C4 21909984 02653750 77BF78

C_3 :

9C3D7360 C30156FA B7C80A02 76712DA9 D8094A63 4B766D3A 285E0748 0653426D

输出密文 $C=C_1\|C_2\|C_3$:

04245C26 FB68B1DD DDB12C4B 6BF9F2B6 D5FE60A3 83B0D18D 1C4144AB F17F6252

E776CB92 64C2A7E8 8E52B199 03FDC473 78F605E3 6811F5C0 7423A24B 84400F01

B8650053 A89B41C4 18B0C3AA D00D886C 00286467 9C3D7360 C30156FA B7C80A02

76712DA9 D8094A63 4B766D3A 285E0748 0653426D

解密各步骤中的有关值:

计算椭圆曲线点 $[d_B]C_1=(x_2, y_2)$:

坐标 x_2 :

64D20D27 D0632957 F8028C1E 024F6B02 EDF23102 A566C932 AE8BD613 A8E865FE

坐标 y_2 :

58D225EC A784AE30 0A81A2D4 8281A828 E1CEDF11 C4219099 84026537 5077BF78

计算 $t=KDF(x_2\|y_2, klen)$:

006E30 DAE231B0 71DFAD8A A379E902 64491603

计算 $M'=C_2\oplus t$:

656E63 72797074 696F6E20 7374616E 64617264

计算 $u=Hash(x_2\|M'\|y_2)$:

9C3D7360 C30156FA B7C80A02 76712DA9 D8094A63 4B766D3A 285E0748 0653426D

明文 M' : 656E63 72797074 696F6E20 7374616E 64617264, 即为: encryption standard

A.3 F_{2^m} 上椭圆曲线消息加解密

椭圆曲线方程为: $y^2+xy=x^3+ax^2+b$

示例2: F_{2^m} -257

基域生成多项式为: $y^{257}+x^{12}+1$

系数 a : 0

系数 b :

00 E78BCD09 746C2023 78A7E72B 12BCE002 66B9627E CB0B5A25 367AD1AD 4CC6242B

基点 $G=(x_G, y_G)$, 其阶记为 n 。

坐标 x_G :

00 CDB9CA7F 1E6B0441 F658343F 4B10297C 0EF9B649 1082400A 62E7A748 5735FADD

坐标 y_G :

01 3DE74DA6 5951C4D7 6DC89220 D5F7777A 611B1C38 BAE260B1 75951DC8 060C2B3E

阶 n :

7FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF BC972CF7 E6B6F900 945B3C6A 0CF6161D

待加密的消息 M : encryption standard

消息 M 的16进制表示: 656E63 72797074 696F6E20 7374616E 64617264

私钥 d_B :

56A270D1 7377AA9A 367CFA82 E46FA526 7713A9B9 1101D077 7B07FCE0 18C757EB

公钥 $P_B=(x_B, y_B)$:

坐标 x_B :

00 A67941E6 DE8A6180 5F7BCFF0 985BB3BE D986F1C2 97E4D888 0D82B821 C624EE57

坐标 y_B :

01 93ED5A67 07B59087 81B86084 1085F52E EFA7FE32 9A5C8118 43533A87 4D027271

加密各步骤中的有关值:

产生随机数 k :

6D3B4971 53E3E925 24E5C122 682DBDC8 705062E2 0B917A5F 8FCDB8EE 4C66663D

计算椭圆曲线点 $C_1=[k]G=(x_1, y_1)$:

坐标 x_1 :

01 9D236DDB 305009AD 52C51BB9 32709BD5 34D476FB B7B0DF95 42A8A4D8 90A3F2E1

坐标 y_1 :

00 B23B938D C0A94D1D F8F42CF4 5D2D6601 BF638C3D 7DE75A29 F02AFB7E 45E91771

在此 C_1 选用未压缩的表示形式, 点转换成字节串的形式为 $PC\|x_1\|y_1$, 其中 PC 为单一字节

且 $PC=04$, 仍记为 C_1 。

计算椭圆曲线点 $[k]P_B=(x_2, y_2)$:

坐标 x_2 :

00 83E628CF 701EE314 1E8873FE 55936ADF 24963F5D C9C64805 66C80F8A 1D8CC51B

坐标 y_2 :

01 524C647F 0C0412DE FD468BDA 3AE0E5A8 0FCC8F5C 990FEE11 60292923 2DCD9F36

消息 M 的比特长度 $klen=152$

计算 $t=KDF(x_2\|y_2, klen)$:

983BCF 106AB2DC C92F8AEA C6C60BF2 98BB0117

计算 $C_2=M \oplus t$:

FD55AC 6213C2A8 A040E4CA B5B26A9C FCDA7373 FCDA7373

计算 $C_3=Hash(x_2 \parallel M \parallel y_2)$:

$x_2 \parallel M \parallel y_2$:

0083E628 CF701EE3 141E8873 FE55936A DF24963F 5DC9C648 0566C80F 8A1D8CC5

1B656E63 72797074 696F6E20 7374616E 64617264 01524C64 7F0C0412 DEF468B

DA3AE0E5 A80FCC8F 5C990FEE 11602929 232DCD9F 36

C_3 :

73A48625 D3758FA3 7B3EAB80 E9CFCABA 665E3199 EA15A1FA 8189D96F 579125E4

输出密文 $C=C_1\|C_2\|C_3$:

04019D23 6DDB3050 09AD52C5 1BB93270 9BD534D4 76FBB7B0 DF9542A8 A4D890A3
F2E100B2 3B938DC0 A94D1DF8 F42CF45D 2D6601BF 638C3D7D E75A29F0 2AFB7E45
E91771FD 55AC6213 C2A8A040 E4CAB5B2 6A9CFCDA 737373A4 8625D375 8FA37B3E
AB80E9CF CABA665E 3199EA15 A1FA8189 D96F5791 25E4

解密各步骤中的有关值:

计算椭圆曲线点 $[d_B]C_1=(x_2,y_2)$:

坐标 x_2 :

00 83E628CF 701EE314 1E8873FE 55936ADF 24963F5D C9C64805 66C80F8A 1D8CC51B

坐标 y_2 :

01 524C647F 0C0412DE FD468BDA 3AE0E5A8 0FCC8F5C 990FEE11 60292923 2DCD9F36

计算 $t=KDF(x_2\|y_2, klen)$:

983BCF 106AB2DC C92F8AEA C6C60BF2 98BB0117

计算 $M'=C_2\oplus t$:

656E63 72797074 696F6E20 7374616E 64617264

计算 $u=Hash(x_2\|M'\|y_2)$:

73A48625 D3758FA3 7B3EAB80 E9CFCABA 665E3199 EA15A1FA 8189D96F 579125E4

明文 M' : 656E63 72797074 696F6E20 7374616E 64617264, 即为: encryption standard
