# 密码学系列讲座

# 第 2 课：公钥加密与数字签名

**lynndell 博士**
新火科技 密码学专家 lynndell2010@gmail.com

**目录**

# 1 群

## 1.1 素数群

**符号说明：**a^b 是指 a 的 b 次方；a_1 下划线表示下标；a*b 是指 a 乘以 b。

**群定义：**一个**集合 G**，满足以下 6 个条件，则称为群。
**1.非空集：**集合中至少有一个元素。
**2.二元运算：**集合中的元素能够进行一种运算，例如**加法**运算、或**乘法**运算。
**3.封闭性：**集合中的元素进行运算后，得到的结果仍然是集合中的元素。
**4.结合律：**任意 a,b,c 属于 G，则(a+b)+c=a+(b+c)。
**5.单位元 e：**加法情况下 a+e=e+a=a，乘法情况下：a*e=e*a=a。
**6.每个元素 a 都有逆元，记为 $a^{-1}$：**（1）加法情况下：$a+ a^{-1}= a^{-1}+a=e$，
（2）乘法情况下：$a*a^{-1}= a^{-1}*a=e$。
因此集合 G 称为群。
可以简单理解为：**具有封闭运算的集合称为群。**

**举例：**{0,1}集合，**除法**不满足**封闭性**。1 除以 0 等于无穷大，超出了集合范围。

群有 6 个性质，主要用到二元运算、封闭性、单位元、**逆元**这四个性质。
而非空集和结合律很容易满足。

**概念 1：**如果一个群元素 g 能够通过有限次**本身运算**，表达群内其他所有元素，则称为群的**生成元**。
**概念 2：**群内元素个数称为群的**阶**。

**例 1：**集合{0,1,2,3,4,5,6}模系数为 7，就是一个**加法素数群** $\mathbb{Z}_7$ 。

**1.非空集：**群内有 7 个元素。
**2.二元运算：**加法。
**3.封闭性：**群内任意两个元素相加后**模 7** 后仍然是群中的元素，如(5+6)mod7=4；
**4.结合律：**((3+4)+5) mod7=(3+(4+5)) mod7=5，结果相同。
**5.单位元 e=0：**3+0=0+3=3。
**6.每个元素都有逆元：**

- 因为 0+0=e=0，所以 0 的逆元是 0；
- 因为(1+6)mod7=e=0，所以 1 的逆元为 6；
- 因为(2+5)mod7=e=0，所以 2 的逆元为 5；
- 因为 3+4mod7=e=0，所以 3 的逆元为 4；

因此集合{0,1,2,3,4,5,6}模系数为 7 就是一个**加法群**。

**7 是素数，这个加法素数群的性质特别好。因为素数 7 与群元素 i 是互素的，所以每个非零元素都是群的生成元。**
例如：群元素 2 能够通过有限次运算，表达其他所有元素：
(2+2)mod7=4 则表达群元素 4，

(2+2+2)mod7=6 则表达群元素 6，
(2+2+2+2)mod7=1 则表达群元素 1，
(2+2+2+2+2)mod7=3 则表达群元素 3，
(2+2+2+2+2+2)mod7=5 则表达群元素 5，
(2+2+2+2+2+2+2)mod7=0 则表达群元素 0。

群元素 3 能够通过有限次运算，表达其他所有元素：
(3)mod7=3
(3+3)mod7=6
(3+3+3)mod7=2
(3+3+3+3)mod7=5
(3+3+3+3+3)mod7=1
(3+3+3+3+3+3)mod7=4
(3+3+3+3+3+3+3)mod7=0

群元素 4 能够通过有限次运算，表达其他所有元素：
(4)mod7=4
(4+4)mod7=1
(4+4+4)mod7=5
(4+4+4+4)mod7=2
(4+4+4+4+4)mod7=6
(4+4+4+4+4+4)mod7=3
(4+4+4+4+4+4+4)mod7=0

群元素 5 能够通过有限次运算，表达其他所有元素：
(5)mod7=5
(5+5)mod7=3
(5+5+5)mod7=1
(5+5+5+5)mod7=6
(5+5+5+5+5)mod7=4
(5+5+5+5+5+5)mod7=2
(5+5+5+5+5+5+5)mod7=0

群元素 6 能够通过有限次运算，表达其他所有元素：
(6)mod7=6
(6+6)mod7=5
(6+6+6)mod7=4
(6+6+6+6)mod7=3
(6+6+6+6+6)mod7=2
(6+6+6+6+6+6)mod7=1
(6+6+6+6+6+6+6)mod7=0
**群元素 1,2,3,4,5,6 均可以通过有限次运算表达其他群元素，**
**这个加法素数群中，任意非零元素均为生成元。**

**例 2：**集合{1,2,3,4,5,6}模系数为 7，是一个**乘法素数群**，记为$\mathbb{Z}_7^*$

**1.非空集：**群内有 6 个元素。
**2.二元运算：乘法**。
**3.封闭性：**群内任意两个元素相乘后模 7 后仍然是群中的元素，如(5*6)mod7=2；
**4.结合律：**((3*4) *5) mod7=(3* (4*5)) mod7=4，结果相同。
**5.单位元为 1：**1 乘以任意元素等于任意元素;3*1=1*3=3。
**6.每个元素都有逆元：**

- 因为(1*1)mod7=1，1 的逆元为 1；
- 因为(2*4)mod7=1，2 的逆元为 4；
- 因为(3*5)mod7=1，3 的逆元为 5；
- 因为(6*6)mod7=1，6 的逆元为 6；

因此集合{1,2,3,4,5,6}模系数为 7 就是一个**乘法群**。

(2)mod7=2
(2*2)mod7=4
(2*2*2)mod7=1
(2*2*2*2)mod7=2
(2*2*2*2*2)mod7=4
(2*2*2*2*2*2)mod7=1
(2*2*2*2*2*2*2)mod7=2
**只能表达 1,2,4 因此 2 不是生成元**

(3)mod7=3，记为$3^1 \bmod 7 = 3$
(3*3)mod7=2，记为$3^2 \bmod 7 = 2$
(3*3*3)mod7=6，记为$3^3 \bmod 7 = 6$
(3*3*3*3)mod7=4，记为$3^4 \bmod 7 = 4$
(3*3*3*3*3)mod7=5，记为$3^5 \bmod 7 = 5$
(3*3*3*3*3*3)mod7=1，记为$3^6 \bmod 7 = 1$
能表达所有元素，**所以 3 是生成元**

已知私钥 sk=5 和生成元 g=3，则能够快速计算 PK=3^5。计算方法：$PK=g^{sk}$

$3^n : 3, 3^2, 3^4, 3^8, 3^{16}, \ldots$以指数方式快速计算出。

**问题：已知任意一个群元素 1，生成元是 3。生成元 3 通过 n 次运算得到群元素 1，求 n？**
**无法对 1 开 n 次根号。只能正向搜索。**
**解决方案：**需要暴力搜索，**遍历**群元素。**因此，需要指数时间。**
这里的 1 是公开的，记为公钥 PK；私钥就是 n=sk。
已知公钥 PK 和生成元 g，计算私钥 sk，需要指数时间，暴力搜索。如果私钥 sk 的空间是 256bit，则暴力搜索时间是 $2^{256}$，**不可行。**

**1. 离散对数困难问题（DL）：**

已知生成元 $g$ 和公钥 $PK$，不能在多项式时间内求私钥 $sk$ 。

2. **计算性 Diffie-Hellman 困难问题（CDH）：（离散对数困难问题的变形）**

   已知 $g, g^a, g^b \in \mathbb{G}$，求 $g^{ab}$ 是困难的。

3. **q 阶强 Diffie-Hellman 困难问题：**

   已知 $g, g^a, ..., g^{a^q} \in \mathbb{G}$ 和随机数 s，求 $g^{1/(a+s)}$ 是困难的。

4. **q 阶强 Diffie-Hellman 求逆困难问题：**

   已知 $g, g^a, ..., g^{a^q} \in \mathbb{G}$，求 $g^{1/a}$ 是困难的。

5. **双线性 Diffie-Hellman 困难问题：**

   已知 $g, g^a, g^b, g^c \in \mathbb{G}$，求 $e(g,g)^{abc}$ 是困难的。

6. **q 阶双线性 Diffie-Hellman 求逆困难问题：**

   已知 $g, g^a, ..., g^{a^q} \in \mathbb{G}$，求 $e(g,g)^{1/a}$ 是困难的。

7. **判决性 Diffie-Hellman 困难问题（DDH）：**

   已知 $g, g^a, g^b, Z \in \mathbb{G}$，判断 $Z \stackrel{?}{=} g^{ab}$ 是困难的。

8. **双线性判决性 Diffie-Hellman 困难问题：**

   已知 $g, g^a, g^b, g^c \in \mathbb{G}, Z \in \mathbb{G}_T$，判断 $Z \stackrel{?}{=} e(g,g)^{abc}$ 是困难的。

9. **判决线性问题：**

   已知 $g, g^a, g^b, g^{ac_1}, g^{bc_2}, Z \in \mathbb{G}$，判断 $Z \stackrel{?}{=} g^{c_1+c_2}$ 是困难的。

**DL,CDH,DDH 是标准困难问题，是最安全的困难问题。**
**其他困难问题是非标准困难问题，是相对安全的困难问题。**

## 1.3 椭圆曲线群

**椭圆曲线群与素数群，几乎相同，仅仅是底层表达细节不一样，导致计算速度不一样。**
**在相同的安全性等级，需要的私钥长度更小。**

### 10.3.1 Abel 群

由第 4 章可知，Abel 群 $G$ 由元素的集合及其上的二元运算 · 组成，有时记为 $\{G, \cdot\}$。将 $G$ 中元素的序偶 $(a,b)$ 与 $G$ 中元素 $(a \cdot b)$ 对应，使得下述公理①成立：

| | |
|---|---|
| (A1) 封闭性 | 若 $a$ 和 $b$ 属于 $G$，则 $a \cdot b$ 也属于 $G$。 |
| (A2) 结合性 | 对 $G$ 中任意的 $a, b$ 和 $c$，$(a \cdot b) \cdot c = a \cdot (b \cdot c)$ |
| (A3) 单位元 | $G$ 中存在元素 $e$，使得对 $G$ 中所有的 $a$，$e \cdot a = a \cdot e = a$。 |
| (A4) 逆元 | 对 $G$ 中任何 $a$，存在 $G$ 中元素 $a'$，使得 $a' \cdot a = a \cdot a' = e$。 |
| (A5) 交换性 | 对 $G$ 中任何 $a$ 和 $b$，有 $a \cdot b = b \cdot a$。 |

椭圆曲线并不是椭圆，之所以称为椭圆曲线是因为它们与计算椭圆周长的方程相似，也是用三次方程来表示的。一般，椭圆曲线的三次方程形为

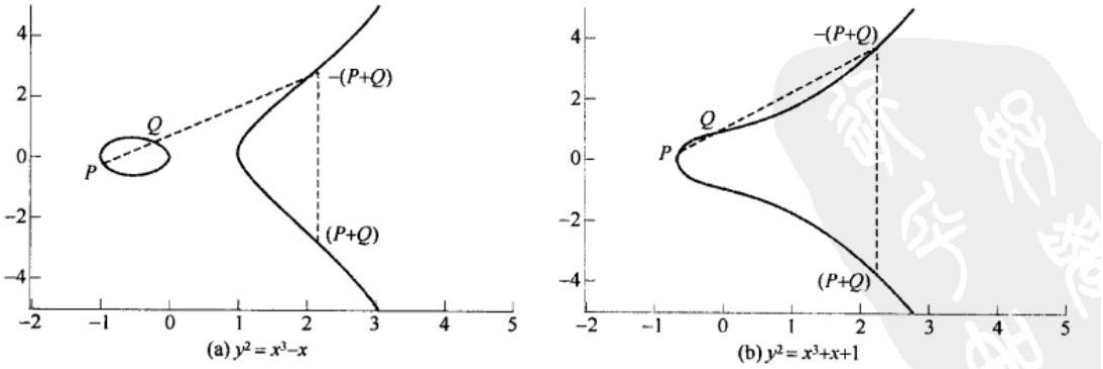
$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

其中 $a,b,c,d$ 和 $e$ 是实数，$x$ 和 $y$ 在实数集上取值[①]。对我们而言将方程限制为下述形式就已足够：

$$y^2 = x^3 + ax + b \tag{10.1}$$

因为方程中的指数最高是3，所以我们称之为三次方程，或者说方程的次数为3。椭圆曲线的定义中还包含一个称为无穷远点或零点的元素，记为 $O$，我们以后再讨论这个概念。为了画出该曲线，我们需要计算：

$$y = \sqrt{x^3 + ax + b}$$




(1) $O$ 是加法的单位元。这样有 $O = -O$；对椭圆曲线上的任何一点 $P$，有 $P + O = P$。下面假定 $P \neq Q$ 且 $Q \neq O$。

(2) 点 $P$ 的负元是具有相同 $x$ 坐标和相反的 $y$ 坐标的点，即若 $P = (x,y)$，则 $-P = (x,-y)$。注意这两个点可用一条垂直的线连接起来，并且 $P + (-P) = P - P = O$。

(3) 要计算 $x$ 坐标不相同的两点 $P$ 和 $Q$ 之和，则在 $P$ 和 $Q$ 间作一条直线并找出第三个交点 $R$，显然存在有唯一的交点 $R$（除非这条直线在 $P$ 或 $Q$ 处与该椭圆曲线相切，此时我们分别取 $R = P$ 或 $R = Q$）。要形成群，需要定义如下三个点上的加法：$P + Q = -R$。也就是说，定义 $P + Q$ 为第三个交点（相对于 $x$ 轴）的镜像。图 10.4 说明了这一情形。

(4) 上述术语的几何解释也适用于具有相同 $x$ 坐标的两个点 $P$ 和 $-P$ 的情形。用一条垂直的线连接这两点，这也可视为在无穷远点处与曲线相交，因此有 $P + (-P) = O$，与上述步骤(2)相一致。

(5) 为计算点 $Q$ 的两倍，画一条切线并找出另一交点 $S$，则 $Q + Q = 2Q = -S$。

利用前述的运算规则，可以证明集合 $E(a,b)$ 是 Abel 群。

**加法的代数描述**

在本小节中，我们给出一些用于椭圆曲线上加法的结论[①]。对不是互为负元的两个不同的点 $P=(x_P,y_P)$ 和 $Q=(x_Q,y_Q)$，连接它们的曲线 $l$ 的斜率 $\Delta=(y_Q-y_P)/(x_Q-x_P)$。$l$ 恰与椭圆曲线相交与另一点，即 $P$ 与 $Q$ 之和的负元。利用某些代数运算，我们可如下表示和 $R=P+Q$：

$$\begin{aligned} x_R &= \Delta^2 - x_P - x_Q \\ y_R &= -y_P + \Delta(x_P - x_R) \end{aligned} \tag{10.3}$$

我们也需要能够计算一个点与它自身相加：$P+P=2P=R$。当 $y_P \neq 0$ 时，该表达式为

$$\begin{aligned} x_R &= \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \\ y_R &= \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P \end{aligned} \tag{10.4}$$

例如，取 $p=23$。考虑椭圆曲线方程 $y^2=x^3+x+1$，这里 $a=b=1$。注意，该方程与图 10.4(b) 中的方程是相同的。图中显示了所有满足方程的实点。对 $E_{23}(1,1)$，我们只对如下非负整数感兴趣，它们位于从 $(0,0)$ 到 $(p-1,p-1)$ 的象限中，满足模 $p$ 的方程。表 10.1 列出了若干点（除 $O$ 外），这些点是 $E_{23}(1,1)$ 的一部分，图 10.5 给出了 $E_{23}(1,1)$ 上的点。注意这些点除了一个之外，均关于 $y=11.5$ 对称。

**表 10.1　椭圆曲线 $E_{23}(1,1)$ 上的点**

| | | |
|---|---|---|
| (0,1) | (6,4) | (12,19) |
| (0,22) | (6,19) | (13,7) |
| (1,7) | (7,11) | (13,16) |
| (1,16) | (7,12) | (17,3) |
| (3,10) | (9,7) | (17,20) |
| (3,13) | (9,16) | (18,3) |
| (4,0) | (11,3) | (18,20) |
| (5,4) | (11,20) | (19,5) |
| (5,19) | (12,4) | (19,18) |

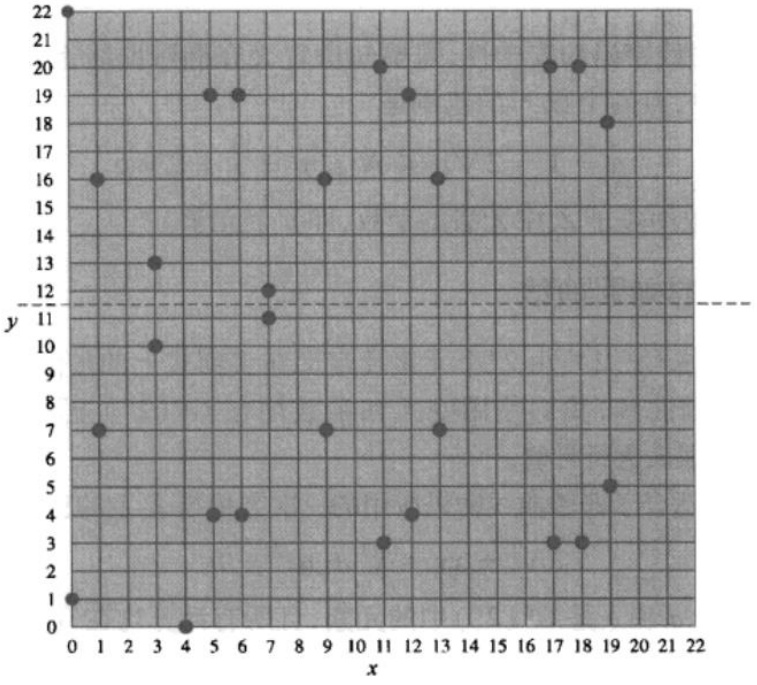



图 10.5　椭圆曲线 $E_{23}(1,1)$

$E_p(a,b)$上的加法运算构造与定义在实数上的椭圆曲线中描述的代数方法是一致的。对任何点$P,Q \in E_p(a,b)$

(1) $P+O=P$。

(2) 若$P=(x_P,y_P)$，则$P+(x_P,-y_P)=O$。点$(x_P,-y_P)$是$P$的负元，记为$-P$。例如，对$E_{23}(1,1)$上的点$P=(13,7)$，有$-P=(13,-7)$，而$-7 \bmod 23=16$，因此，$-P=(13,16)$，该点也在$E_{23}(1,1)$上。

(3) 若$P=(x_P,y_P)$，$Q=(x_Q,y_Q)$，且$P \neq -Q$则$R=P+Q=(x_R,y_R)$由下列规则确定：

$$x_R=(\lambda^2-x_P-x_Q) \bmod p$$
$$y_R=(\lambda(x_P-x_R)-y_P) \bmod p$$

其中

$$\lambda=\begin{cases}\left(\dfrac{y_Q-y_P}{x_Q-x_P}\right) \bmod p, & P \neq Q \\ \left(\dfrac{3x_P^2+a}{2y_P}\right) \bmod p, & P=Q\end{cases}$$

(4) 乘法定义为重复相加。如$4P=P+P+P+P$。

例如取$E_{23}(1,1)$上的$P=(3,10)$，$Q=(9,7)$，那么

$$\lambda=\left(\frac{7-10}{9-3}\right) \bmod 23=\left(\frac{-3}{6}\right) \bmod 23=\left(\frac{-1}{2}\right) \bmod 23=11$$
$$x_R=(11^2-3-9) \bmod 23=109 \bmod 23=17$$
$$y_R=(11(3-17)-10) \bmod 23=-164 \bmod 23=20$$

所以$P+Q=(17,20)$。为计算$2P$，先求

$$\lambda=\left(\frac{3(3^2)+1}{2\times 10}\right) \bmod 23=\left(\frac{5}{20}\right) \bmod 23=\left(\frac{1}{4}\right) \bmod 23=6$$

上述等式的最后一步中需求4在$Z_{23}$中的乘法逆元。这可以用4.4节定义的扩展Euclid算法实现。注意到$(6\times 4) \bmod 23=24 \bmod 23=1$。

$$x_R=(6^2-3-3) \bmod 23=30 \bmod 23=7$$
$$y_R=(6(3-7)-10) \bmod 23=(-34) \bmod 23=12$$

可见$2P=(7,12)$。

为了确定各种椭圆曲线密码的安全性，需要知道定义在椭圆曲线上的有限Abel群中点的个数。在有限群$E_p(a,b)$中，点的个数$N$的范围是

$$p+1-2\sqrt{p} \leqslant N \leqslant p+1+2\sqrt{p}$$

所以，$E_p(a,b)$上点的个数约等于$Z_p$中元素的个数，即$p$个元素。

**已知$P$和$n$，能在多项式是时间内计算$n \cdot P$。**

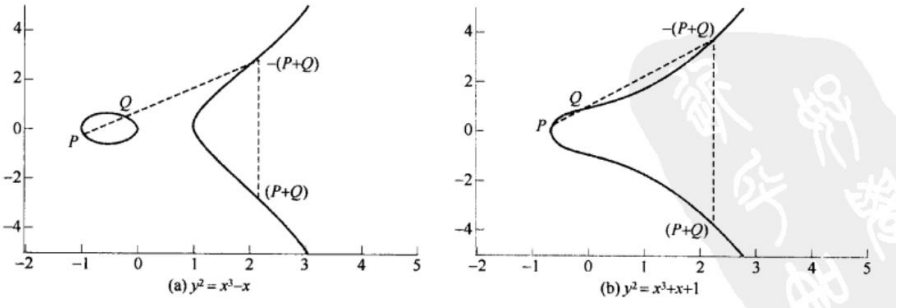
与椭圆曲线相切于第3个点，找对称点。如果$n=2^a$，则需要a次**切线计算**。

如果$n=2^a+2^b+...+2^x$，则需要a+b+…+x次**切线计算**和少量**点加计算**。

**反之，已知$P$和$n\cdot P$，需要指数时间计算$n$。**
**思考：如何反向运算？**

$Q=n\cdot P, n=2^a$找对称点 Q'， Q'与椭圆曲线相切于 1 个或 2 个点 Q1/Q2，Q1/Q2 找对称点，与椭圆曲线相切于 2/4 个点 A1/A2/A3/A4，以此类推， $n=2^{256}$，**计算复杂度和存储空间呈指数增加。**找到$2^{256}$个点，检测否为 P 点。如果是 P 点，则找到了，否则继续找。
对于任意点

$$Y=nP=(2^a+2^b+...+2^x)P$$
$$2^aP, 2^bP, ..., 2^xP$$




**椭圆曲线群上的困难问题：**

1. **离散对数困难问题：**

   已知$G, a\cdot G\in\mathbb{G}$，求$a$是困难的。

2. **计算性 Diffie-Hellman 困难问题：**

   已知$G, a\cdot G, b\cdot G\in\mathbb{G}$，求$ab\cdot G$是困难的。

3. **q 阶强 Diffie-Hellman 困难问题：**

   已知$G, a\cdot G, ..., a^q\cdot G\in\mathbb{G}$和随机数 s，求$(1/(a+s))\cdot G$是困难的。

4. **q 阶强 Diffie-Hellman 求逆困难问题：**

   已知$G, a\cdot G, ..., a^q\cdot G\in\mathbb{G}$，求$(1/a)\cdot G$是困难的。

5. **双线性 Diffie-Hellman 困难问题：**

   已知$G, a\cdot G, b\cdot G, c\cdot G\in\mathbb{G}$，求$e(G,G)^{abc}$是困难的。

6. **q 阶双线性 Diffie-Hellman 求逆困难问题：**

   已知$G, a\cdot G, ..., a^q\cdot G\in\mathbb{G}$，求$e(G,G)^{1/a}$是困难的。

7. **判决性 Diffie-Hellman 困难问题：**

   已知$G, a\cdot G, b\cdot G, Z\in\mathbb{G}$，判断$Z\overset{?}{=}ab\cdot G$是困难的。

8. **双线性判决性 Diffie-Hellman 困难问题：**

   已知$G, a\cdot G, b\cdot G, c\cdot G\in\mathbb{G}, Z\in\mathbb{G}_T$，判断$Z\overset{?}{=}e(G,G)^{abc}$是困难的。

9. **判决线性问题：**

已知$G, a\cdot G, b\cdot G, ac_1\cdot G, bc_2\cdot G, Z\in\mathbb{G}$，判断$Z\overset{?}{=}(c_1+c_2)\cdot G$是困难的。

# 2 公钥加密

## 2.1ElGamal 加密系列

### 2.1.1 基于素数群的 ElGamal 加密

**系统参数：**素数群$\mathbb{G}$的阶为$p$，生成元为$g$。

**密钥生成：**选择随机数$\alpha\in\mathbb{Z}_p$，计算$g_1=g^{\alpha}$，则私钥和公钥为$(\alpha, g_1)$。

**加密：**将消息编码为群元素$m\in\mathbb{G}$，公钥$g_1$，选择随机数$r\in\mathbb{Z}_p$，计算密文

$$C_1=g^r, C_2=g_1^r\cdot m$$

**解密：**对于密文$(C_1, C_2)$，使用私钥$\alpha$，计算$m=C_2\cdot C_1^{-\alpha}$

公式推导过程：$C_2\cdot C_1^{-\alpha}=(g_1^r\cdot m)(g^r)^{-\alpha}=m$

安全性分析：基于 DDH 困难问题，归约损失为 2，1bit。
离散对数是困难的，但是计算逆元是简单的

$$h^p=e$$
$$h\cdot h^{p-1}=e$$
$$h\cdot h^{-1}=e$$
$$h^{-1}=h^{p-1}$$

**方案扩展：**

**应用需求：**将消息编码到群上$m\in\mathbb{G}$ **很难**，**需要**加密**任意长的**数据$m\in\{0,1\}^*$。

**加密：**对于任意长消息$m\in\{0,1\}^*$，公钥$g_1$，选择随机数$r\in\mathbb{Z}_p$和随机群元素$x\in\mathbb{G}$，计算密文

$$C_1=g^r, C_2=g_1^r\cdot x, C_3=GCM\_AES\_Enc(hash(x), m)$$

**解密：**对于密文$(C_1, C_2)$，使用私钥$\alpha$，计算$x=C_2\cdot C_1^{-\alpha}$，然后计算

$$m=GCM\_AES\_Dec(hash(x), C_3)$$

**数字信封：**公钥加密一个对称密钥 x，使用对称密钥 x 进行 AES **对称加密**。

### 2.1.2 基于素数群的 Hashed ElGamal 加密

**系统参数：**素数群$\mathbb{G}$的阶为$p$，生成元为$g$。

**密钥生成：**选择随机数$\alpha \in \mathbb{Z}_p$，计算$g_1 = g^{\alpha}$，则私钥和公钥为$(\alpha, g_1)$。

**加密：**对于消息$m \in \{0,1\}^{256}$，公钥$g_1$，选择随机数$r \in \mathbb{Z}_p$，计算密文

$$C_1 = g^r, C_2 = hash(g_1^r) \oplus m$$

**解密：**对于密文$(C_1, C_2)$，使用私钥$\alpha$，计算$m = C_2 \oplus hash(C_1^{\alpha})$

公式推导过程：$C_2 \cdot C_1^{-\alpha} = \left(hash(g_1^r) \oplus m\right) \oplus hash\left((g^r)^{\alpha}\right) = m$

安全性分析：基于 CDH 困难问题，归约损失为$q_{hash}$，是查询随机预言机的次数，通常是$2^{40}$。

假设私钥长度为 256bit，算法安全性为 216bit。

### 2.1.3 基于椭圆曲线群的 ElGamal 加密

**系统参数：**椭圆曲线群$\mathbb{G}$的阶为$p$，生成元为$G$。

**密钥生成：**选择随机数$\alpha \in \mathbb{Z}_p$，计算$G_1 = \alpha \cdot G$，则私钥和公钥分别为$(\alpha, G_1)$。

**加密：**消息编码到椭圆曲线群$M \in \mathbb{G}$，公钥$G_1$，选择随机数$r \in \mathbb{Z}_p$，计算密文

$$C_1 = r \cdot G, C_2 = M + r \cdot G_1$$

**解密：**对于密文$(C_1, C_2)$，使用私钥$\alpha$，计算$M = C_2 - \alpha \cdot C_1$

**公式过程：**$C_2 - \alpha \cdot C_1 = (M + r \cdot G_1) - \alpha r \cdot G = M$

安全性分析：基于 DDH 困难问题，归约损失为 2。

Zn
Com=g^r * **h**^m
**x=Hash(g)**

### 2.1.4 基于椭圆曲线群的 Hashed ElGamal 加密

**加密：**对于消息$m \in \{0,1\}^{256}$，公钥$G_1$，选择随机数$r \in \mathbb{Z}_p$，计算密文

$$C_1 = r \cdot G, C_2 = m \oplus hash(r \cdot G_1)$$

**解密：**对于密文$(C_1, C_2)$，使用私钥$\alpha$，计算$m = C_2 \oplus hash(\alpha \cdot C_1)$

**公式过程：**$C_2 \oplus hash(\alpha \cdot C_1) = \left(m \oplus hash(r \cdot G_1)\right) \oplus hash(\alpha r \cdot G) = m$

安全性分析：基于 CDH 困难问题，归约损失为 $q_{hash}$，查询随机预言机的次数，通常是 $2^{40}$。

私钥长度为 256bit，算法安全性为 216bit。对于椭圆曲线，通常 70bit 的随机数就足够安全。

### 2.1.5 基于素数群的 Twin Hashed ElGamal 加密

**系统参数：**素数群 $\mathbb{G}$ 的阶为 $p$，生成元为 $g$。

**密钥生成：**选择 2 个随机数 $\alpha,\beta\in\mathbb{Z}_p$，计算 $g_1=g^{\alpha},g_2=g^{\beta}$，则私钥为 $\alpha,\beta$ 和公钥为 $g_1,g_2$。

**加密：**对于消息 $m\in\{0,1\}^{256}$，公钥 $g_1,g_2$，选择随机数 $r\in\mathbb{Z}_p$，计算密文

$$C_1=g^r,C_2=hash(g_1^r,g_2^r)\oplus m$$

**解密：**对于密文 $(C_1,C_2)$，使用私钥 $\alpha,\beta$，计算 $m=C_2\oplus hash(C_1^{\alpha},C_2^{\beta})$

公式推导过程：$C_2\oplus hash(C_1^{\alpha},C_2^{\beta})=(hash(g_1^r,g_2^r)\oplus m)\oplus hash(g^{r\alpha},g^{r\beta})=m$

安全性分析：添加了一个随机因子（私钥），基于 CDH 困难问题，无归约损失。

## 2.2ElGamal 加密安全升级（Cramer-Shoup）

### 2.2.1 基于素数群 Cramer-Shoup 加密

**系统参数：**素数群 $\mathbb{G}$ 的阶为 $p$，生成元为 $g_1,g_2$。

**密钥生成：**选择 5 个随机数 $x_1,x_2,y_1,y_2,z\in\mathbb{Z}_p$，计算 3 个群元素 $c=g_1^{x_1}g_2^{x_2},d=g_1^{y_1}g_2^{y_2},h=g_1^{z}$，则私钥为 $(x_1,x_2,y_1,y_2,z)$ 和公钥为 $(c,d,h)$。

**加密：**对于消息 $m\in\mathbb{G}$，公钥 $(c,d,h)$，选择随机数 $r\in\mathbb{Z}_p$，计算密文 $(u_1,u_2,e,v)$

$$\begin{aligned}&u_1=g_1^r,u_2=g_2^r,e=h^r\cdot m,\\&\alpha=hash(u_1,u_2,e),\\&v=c^rd^{r\alpha}\end{aligned}$$

**解密：**对于密文 $(u_1,u_2,e,v)$，使用私钥 $(x_1,x_2,y_1,y_2,z)$，计算 $\alpha=hash(u_1,u_2,e)$

校验 $v=u_1^{x_1+y_1\alpha}u_2^{x_2+y_2\alpha}$，然后解密 $m=e\cdot u_1^{-z}$。

公式推导过程：
$$u_1^{x_1+y_1\alpha}u_2^{x_2+y_2\alpha}=(g_1^r)^{x_1+y_1\alpha}(g_2^r)^{x_2+y_2\alpha}=(g_1^{x_1}g_2^{x_2})^r(g_1^{y_1}g_2^{y_2})^{r\alpha}=c^rd^{r\alpha}=v$$
$$e\cdot u_1^{-z}=(h^r\cdot m)\cdot(g_1^r)^{-z}=m$$

**优点：**校验使得安全性从 CPA 安全**提升**到 CCA 安全。

**缺点：**私钥、公钥、密文长度增加，计算复杂度增加。

### 2.2.1 基于椭圆曲线群 Cramer-Shoup 加密

**系统参数：**椭圆曲线群$\mathbb{G}$的阶为$p$，生成元为$G_1, G_2$。

**密钥生成：**选择 5 个随机数$x_1, x_2, y_1, y_2, z \in \mathbb{Z}_p$，计算 3 个群元素$C = x_1 \cdot G + x_2 \cdot G_2, D = y_1 \cdot G + y_2 G_2, H = z \cdot G_1$，则私钥为$(x_1, x_2, y_1, y_2, z)$和公钥为$(C, D, H)$。

**加密：**对于消息$M \in \mathbb{G}$，公钥$(C, D, H)$，选择随机数$r \in \mathbb{Z}_p$，计算密文$(U_1, U_2, e, V)$

$$
\begin{aligned}
&U_1 = r \cdot G_1, U_2 = r \cdot G_2, E = r \cdot H + M, \\
&\alpha = hash(U_1, U_2, E), \\
&V = r \cdot C + r\alpha \cdot D
\end{aligned}
$$

**解密：**对于密文$(U_1, U_2, e, V)$，使用私钥$(x_1, x_2, y_1, y_2, z)$，计算$\alpha = hash(U_1, U_2, E)$

校验$V = (x_1 + y_1\alpha) \cdot U_1 + (x_2 + y_2\alpha) \cdot U_2$，然后解密$M = E - z \cdot U_1$

**公式推导过程：**

$$
\begin{aligned}
&(x_1 + y_1\alpha) \cdot U_1 + (x_2 + y_2\alpha) \cdot U_2 = (x_1 + y_1\alpha) r \cdot G_1 + (x_2 + y_2\alpha) r \cdot G_2 \\
&= x_1 r \cdot G_1 + y_1 \alpha r \cdot G_1 + x_2 r \cdot G_2 + y_2 \alpha r \cdot G_2 \\
&= r(x_1 G_1 + x_2 \cdot G_2) + r\alpha(y_1 \cdot G_1 + y_2 \cdot G_2) \\
&= rC + r\alpha D = V \\
&E - zU_1 = (r \cdot H + M) - zr \cdot G_1 = M
\end{aligned}
$$

**优点：**校验使得安全性从 CPA 安全**提升**到 CCA 安全。

**缺点：**私钥、公钥、密文长度增加，计算复杂度增加。

## 3.8 ECIES 加密

**系统参数：**椭圆曲线群$\mathbb{G}$的阶为$n$，生成元为$G$。$h$为余因子常量。$KDF$为密钥派生函数。

**密钥生成：**选择随机数$d \in \mathbb{Z}_p$，计算$Q = d \cdot G$，则私钥和公钥分别为$(d, Q)$。

**加密：**选择随机数$k \in [1, n-1]$，计算$R = k \cdot G, Z = hk \cdot Q$。取$Z$的横坐标为$x_Z$，计算 2 个随机数$(k_1, k_2) = KDF(x_Z, R)$。使用 **CTR-AES 加密模式**对任意长消息$m \in \{0,1\}^*$加密

$$
C = CTR\_AES\_Enc_{k_1}(m), t = MAC_{k_2}(C)
$$

密文为$(R, C, t)$。

**解密：**对于密文$(R,C,t)$，使用私钥$d$，计算$Z'=hd\cdot R$。取$Z$的横坐标为$x_Z'$，计算 2 个随机数$(k_1',k_2')=KDF(x_Z',R)$。计算$t'=MAC_{k_2'}(C)$，校验$t'=t$，然后 **CTR-AES 解密模式：**$m=CTR\_AES\_Dec_{k_1'}(C)$。

**公式推导过程：**$Z=hk\cdot Q=hkd\cdot G=hd\cdot R=Z'$

**分析：**

（1）可以根据该算法设计素数群上的对应的 ECIES 加密算法。

（2）有**检验**过程，但是私钥长度为 1，公钥长度为 1，数据量降低。

（3）**引入** KDF 和 MAC 算法，这些算法安全性会影响整个算法安全性，即添加了额外的假设：要求 KDF 和 MAC 算法是安全的。

（4）也可以使用 GCM 模式，只是 ECIES 已规定使用 CTR 模式。

# 3 数字签名

- **群：**具有封闭运算的集合，加法运算。1
- **环：**群 + 乘法运算，缺乘法逆元。1.5
- **域：**环 + 每个元素均有乘法逆元。2

## 3.1 双线性映射

1. $G_1,G_2,G_T$ 是三个阶为素数 $p$ 的循环群；
2. 群 $G_1$ 生成元为 $g_1$，群 $G_2$ 生成元为 $g_2$；
3. e 为双线性映射 $G_1\times G_2\rightarrow G_T$；
4. 双线性映射具有以下两个性质：
   - 双线性：对任意 $a,b\in Z^*,u\in G_1,v\in G_2$，有 $e(u^a,v^b)=e(u,v)^{ab}$ 均成立；
   - 非退化性：$e(g_1,g_2)\neq 1$。

   因此，$e(g_1,g_2)$ 是群 $G_T$ 的生成元。

**二次剩余【开平方】**

**素数群 $G_1$ 的模系数为 7**，$3^2 \bmod 7=2$，所以 3 就是 2 的模 7“平方根”二次剩余。

理解：$\sqrt{2}\bmod 7=3$

**如果某个非零元素是可以开平方根的，称这样的元素为模 n 的二次剩余，否则就叫模 n 的二次非剩余。**

| n=7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 二次剩余 | - | 1,6 | 3,4 | 二次非剩余 | 2,5 | 二次非剩余 | 二次非剩余 |

$$1^2 \bmod 7 = 1$$
$$2^2 \bmod 7 = 4$$
$$3^2 \bmod 7 = 2$$
$$4^2 \bmod 7 = 2$$
$$5^2 \bmod 7 = 4$$
$$6^2 \bmod 7 = 1$$

二次剩余的个数（1，2，4 共 3 个）和二次非剩余的个数（3，5，6 共 3 个）。二次剩余与非剩余个数相等的（不讨论 0），二次剩余的逆元仍然是二次剩余，二次非剩余的逆元也仍然是二次非剩余；而且每个二次剩余都有两个根，且他们的**和**模 7 为 0。

**域的扩张**

3 在 $F_7$ 中没有“平方根”。

**类比复数对实数的扩展，假设 3 的一个平方根为 j，即 j*jmod7=3。**

$$\sqrt{3} \bmod 7 = j$$

把 j 加入到{0,1,2,3,4,5,6}集合中，然后再加入其他元素，使得新的集合仍然构成一个域。

**一共添加 49 个元素：**

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| j | 1+j | 2+j | 3+j | 4+j | 5+j | 6+j |
| 2j | 1+2j | 2+2j | 3+2j | 4+2j | 5+2j | 6+2j |
| 3j | 1+3j | 2+3j | 3+3j | 4+3j | 5+3j | 6+3j |
| 4j | 1+4j | 2+4j | 3+4j | 4+4j | 5+4j | 6+4j |
| 5j | 1+5j | 2+5j | 3+5j | 4+5j | 5+5j | 6+5j |
| 6j | 1+6j | 2+6j | 3+6j | 4+6j | 5+6j | 6+6j |

举例：

加法 6j+j mod7 = 0 满足**加法封闭性**

乘法(3+j)(5+2j)=15 + 11j + $2j^2$ mod7 =15 + 11j + 2*3 mod7 = 4j **乘法封闭性**

因为(4+4j)(6+1j)= 24-$24j^2$ mod7 = 24-24*3 mod7=-48mod7=1 逆元存在性且唯一

所以**逆元**：$(4+4j)^{-1}$ mod7 = 6+1j 或$(6+1j)^{-1}$ mod7 = 4+4j

这 49 个元素能够完成**封闭的四则运算**

对于**加法和减法**：容易验证任意两个元素的和、差均在集合中；

对于**乘法**：根据扩张规则 **j*jmod7=3**，任何两个元素的积在集合中；

对于**除法**：可以通过如下方式计算 a+bj 的逆元（a 和 b 不同时为 0）：

$$\frac{1}{a+bj} = \frac{a-bj}{(a+bj)(a-bj)} = \frac{a-bj}{a^2-3b^2} = \frac{1}{a^2-3b^2}(a-bj) = c(a-bj)$$

**在 49 个元素中，选择 7 个元素，作为扩张子群，就是群 $G_2$，即群 $G_2$ 的阶也是 7。**

**选择方法：例如以群元素 1+j 作为群 $G_2$ 的生成元 g，进行加法运算，模系数为 7，能够计算每个群元素。**

**双线性映射通过 Millier 循环计算**

$$e(P,Q)=f_r(x_Q,y_Q)^{\frac{p^k-1}{r}}$$

其中素数**域**的阶为 p，椭圆曲线**群 $G_1$ 的阶为 r**，k 为嵌入度（常量），函数 $f_r$ 满足**递归关系：**

$$f_1=1,$$
$$f_{i+1}=f_i\cdot l_{(iP,P)},$$
$$f_i=f_{i/2}^2\cdot l_{(i/2P,-iP)}$$

其中 $l_{R,S}$ 是经过点 R 和点 S 的**直线方程**。

举例： BN254 椭圆曲线群的阶为 $r$ 非常大，假设一个小群的阶 r=17

$$\begin{aligned}
f_{17}&=f_{16}\cdot l_{(16P,P)}\\
&=f_8^2\cdot l_{(8P,-16P)}\cdot l_{(16P,P)}\\
&=(f_4^2\cdot l_{(4P,-8P)})^2\cdot l_{(8P,-16P)}\cdot l_{(16P,P)}\\
&=\left(\left(f_2^2\cdot l_{(2P,-4P)}\right)^2\cdot l_{(4P,-8P)}\right)^2\cdot l_{(8P,-16P)}\cdot l_{(16P,P)}\\
&=\left(\left(\left(f_1^2\cdot l_{(P,-2P)}\right)^2\cdot l_{(2P,-4P)}\right)^2\cdot l_{(4P,-8P)}\right)^2\cdot l_{(8P,-16P)}\cdot l_{(16P,P)}\\
&=\left(\left(\left(f_1^2\right)^2\right)^2\right)^2\cdot l_{(P,-2P)}^8\cdot l_{(2P,-4P)}^4\cdot l_{(4P,-8P)}^2\cdot l_{(8P,-16P)}\cdot l_{(16P,P)}\\
&=l_{(P,-2P)}^8\cdot l_{(2P,-4P)}^4\cdot l_{(4P,-8P)}^2\cdot l_{(8P,-16P)}\cdot l_{(16P,P)}
\end{aligned}$$

椭圆曲线群 $G_1$，阶为 17，基域为 101

| **P=5G** | **2P=10G** | **4P=20G=3G** | **8P=40G=6G** | **16P=12G** |
|---|---|---|---|---|
| (12,32) | (91,66) | (26,45) | (32,42) | (12,69) |
| - | **-2P=(-2 mod17)P= 75G=7G** | **-4P mod17=(-4 mod17)P =65G=14G** | **-8P=(-8 mod17)P=11G** | **-16P=(-16 mod17)P=5G** |
| | (91,35) | (26,56) | (32,59) | (12,32) |

| 椭圆曲线群 $G_1$ 上两个点 (P,-2P)确定直线方程 $l_{R,S}$ | 直线方程 | $l_{R,S}$ 直线方程 | 带入椭圆曲线群 $G_2$ 上的点 $(x_Q,y_Q)=(10,16j)$ |
|---|---|---|---|
| (P,-2P)=(5G,7G) =((12,32),(91,35)) | $y-32=\frac{35-32}{91-12}(x-12)$ | $l_{R,S}=79y-3x+33\bmod 101$ | $l_{R,S}=3+52j$ |

$$
\begin{aligned}
&f_{17} = l^{8}_{(P,-2P)} \cdot l^{4}_{(2P,-4P)} \cdot l^{2}_{(4P,-8P)} \cdot l_{(8P,-16P)} \cdot l_{(16P,P)} \\
&= (3+52j)^{8} \cdot (x_1 + y_1 j)^{4} \cdot (x_2 + y_2 j)^{2} \cdot (x_3 + y_3 j) \cdot (x_4 + y_4 j) \in \mathbb{G}_T \\
&= x + yj \bmod r \in \mathbb{G}_T \\
&j^2 \bmod r = c
\end{aligned}
$$

$$
e(P,Q) = f_r(x_Q, y_Q)^{\frac{p^k-1}{r}} = (x+yj)^{\frac{p^k-1}{r}} \bmod r = n + mj \in \mathbb{G}_T
$$

因此，能够计算出 $e(a \cdot G_1, b \cdot G_2) = n + mj \in \mathbb{G}_T$

$e(G_1, G_2)^{ab} = n + mj \in \mathbb{G}_T$

**因此，双线性映射成立：** $e(a \cdot G_1, b \cdot G_2) = e(G_1, G_2)^{ab}$

## 3.2BLS 签名

$(G_1, G_2)$ 是 co-GDH 群对，且 $|G_1| = |G_2| = p$。全域哈希函数 $H : \{0,1\}^* \to G_1$。BLS 签名包括 3 个算法，分别为：密钥生成，签名和验证。

- **密钥生成：**随机选择群 $\mathbb{Z}_p$ 中的元素 $x$，计算 $v = g_2^x$，则公钥属于群 $G_2$ 中的元素，私钥为 $x$。
- **签名：**给定私钥 $x$ 和消息 $M$，如下计算
$$\sigma = H(M)^x$$
则签名为 $\sigma$。
- **验证：**给定公钥 $v$，消息 $M$，以及签名 $\sigma$，如果以下等式成立：
$$e(H(M), v) = e(\sigma, g_2)$$
则签名有效，否则无效。

$$
e(\sigma_m, g_2) = e(H(m)^x, g_2) = e(H(m), g_2)^x = e(H(m), g_2^x) = e(H(m), v)
$$

BLS 签名仅 1 个随机因子，即私钥。

安全性：如果 CDH 问题是困难的，则签名满足不可伪造性，归约损失为 $q_H$，40 比特损失。

例如：BLS 签名算法，384bit 的私钥长度，则签名安全性仅有 344bit。
EdDSA 和 ECDSA 私钥长度仅 256bit。
**应用场景：BLS 用于区块链共识投票，不用于交易签名。**

**BLS 签名扩展**

选择随机位 $b \in \{0,1\}$ 或选择随机数 $r \in \mathbb{Z}_p$

**扩展 1：**令 $M' = \{M \mid b\}$

**扩展 2：**令 $M' = \{M \mid r\}$

签名：$\sigma = H(M')^x$

广播：$(M, \sigma, pk, b/r)$

数据拼接：$M' \leftarrow \{M \mid b\}, M' \leftarrow \{M \mid r\}$

验证公式：$e(\sigma, g) = e(H(M'), h)$

因为引入了随机数 b 或 r，所以安全性发生了变化。

**扩展 1：**使用随机位 $b \in \{0,1\}$，安全性证明归约损失为 2，损失 1bit。

**扩展 2：**使用随机数 $r \in \mathbb{Z}_p$，安全性证明归约损失为 1，无损失。

## 3.3BLS 聚合签名

上述 BLS 签名方案具有以下三个关键性质：1 允许把不同实体对不同消息的签名集合到一个签名中，且拥有所有签名的任意节点均能够聚集签名。2 允许加入新成员。3 任意两个签名集合后，允许第三个签名聚合到该签名集合中。

对任意用户 $i$，其中 $i = 1, ..., n$，私钥为 $x_i \in \mathbb{Z}_p$，公钥为 $v_i = g_2^{x_i} \in G_2$。

- **聚合签名：**用户 $i$ 对一个消息 $M_i \in \{0,1\}^*$ 签名获得 $\sigma_i = H(M_i)^{x_i} \in G_1$。聚集所有的签名，则仅需要如下计算

$$\sigma_{1,2,...,n} \leftarrow \sigma_1 \cdot \sigma_2 \cdot ... \cdot \sigma_n \in G_1.$$

- **验证：**给定对应的所有公钥 $v_1, .., v_n \in G_2$，所有的消息 $M_1, ..., M_n \in \{0,1\}^*$，聚集签名为 $\sigma_{1,2,...,n} \in G_1$。验证对于所有的用户 $i$ 签名的消息 $M_i$，$i = 1, ..., n$，如下检测：
  1. 消息 $M_1, ..., M_n$ 互不相同；
  2. $e(\sigma_{1,2,...,n}, g_2) = \prod_{i=1}^{n} e(H(M_i), v_i)$。

  如果上述两个条件均成立，则接受聚集签名，否则拒绝。

**双线性映射公式推导：**

$$e(u^x \cdot u^y, v) = e(u^{x+y}, v) = e(u, v)^{x+y} = e(u, v)^x \cdot e(u, v)^y = e(u^x, v) \cdot e(u^y, v)$$

$let: a = u^x, b = u^y$

$then:$

$e(u^x \cdot u^y, v) = e(a \cdot b, v)$

$e(u^x, v) \cdot e(u^y, v) = e(a, v) \cdot e(b, v)$

$therefore:$

$e(a \cdot b, v) = e(a, v) \cdot e(b, v)$

**验证：**

$$e(\sigma_{1,2,...,n},g_2)=e(\sigma_1\cdot\sigma_2\cdot...\cdot\sigma_n,g_2)=e(\sigma_1,g_2)\cdot e(\sigma_2,g_2)\cdot...\cdot e(\sigma_n,g_2)$$
$$=\prod_{i=1}^{n}e(\sigma_i,g_2)=\prod_{i=1}^{n}e(H(M_i)^{x_i},g_2)=\prod_{i=1}^{n}e(H(M_i),g_2)^{x_i}=\prod_{i=1}^{n}e(H(M_i),g_2^{x_i})$$
$$=\prod_{i=1}^{n}e(H(M_i),v_i)$$

**功能：压缩签名数据长度。应用到区块链共识算法。**
**上述方案双线性映射进行了 n+1 次，**
**下述方案双线性映射只进行了 2 次。**

## 3.4BLS 批量验证

假设 $n$ 个用户对同一个消息 $M$ 签名，则能够实现批验证，且验证速度极快。任意用户获得的 $n$ 个签名为 $\sigma_1,...,\sigma_n$，则能够对这 $n$ 个签名进行批验证，其验证速度远远快于逐个验证。

用户 $i$ 的私钥 $x_i\in\mathbb{Z}_p$，公钥 $v_i\in g_2^{x_i}\in G_2$，签名 $\sigma_i=H(M)^{x_i}\in G_1$，则

1. 随机选择 $n$ 个整数 $c_1,...,c_n\in[0,B]$，$B$ 为某个固定值；
2. 计算 $V\leftarrow\prod_{i=1}^{n}v_i^{c_i}\in G_2, U\leftarrow\prod_{i=1}^{n}\sigma_i^{c_i}\in G_1$；
3. 如下检测等式是否成立，$e(U,g_2)=e(H(M),V)$。

如果满足上述两个条件，则接受所有的 $n$ 个签名，否则拒绝。

双线性映射：计算复杂度很高；尽量少算；
其他群运算计算复杂度很低，可以多算。

**双线性映射公式推导：**

$$e(a\cdot b,c)=e(a,c)\cdot e(b,c)$$
$$e\left(\prod_{i=1}^{n}a_i,b\right)=e(a_1\cdot a_2\cdot...\cdot a_n,b)=e\left(a_1,b\right)\cdot e\left(a_2,b\right)\cdot...\cdot e\left(a_n,b\right)=\prod_{i=1}^{n}e(a_i,b)$$

$$V\leftarrow\prod_{i=1}^{n}v_i^{c_i}=\prod_{i=1}^{n}g_2^{x_i\cdot c_i},$$
$$U\leftarrow\prod_{i=1}^{n}\sigma_i^{c_i}=\prod_{i=1}^{n}H(M)^{x_i\cdot c_i}$$

$$e(U,g_2)=e\left(\prod_{i=1}^{n}H(M)^{x_i\cdot c_i},g_2\right)=\prod_{i=1}^{n}e\left(H(M)^{x_i\cdot c_i},g_2\right)=\prod_{i=1}^{n}e\left(H(M),g_2\right)^{x_i\cdot c_i}$$
$$e(H(M),V)=e\left(H(M),\prod_{i=1}^{n}g_2^{x_i\cdot c_i}\right)=\prod_{i=1}^{n}e\left(H(M),g_2^{x_i\cdot c_i}\right)=\prod_{i=1}^{n}e\left(H(M),g_2\right)^{x_i\cdot c_i}$$

随机数 $c_1,...,c_n$ 起随机化作用，防止牛头对马嘴。

**Schwartz–Zippel 引理**

$\mathrm{P}$为有限域$\mathbb{F}$上的多项式$\mathrm{P}=\mathrm{F}(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n})$，阶为$\mathrm{d}$。令$\mathrm{S}$为有限域$\mathbb{F}$的子集，从$\mathrm{S}$中选择随机数$\mathrm{r}_1,\ldots,\mathrm{r}_\mathrm{n}$，则多项式等于零的概率可忽略

$$\Pr[\mathrm{P}(\mathrm{r}_1,\ldots,\mathrm{r}_\mathrm{n})=0]\leq\frac{\mathrm{d}}{|\mathrm{S}|}$$

在单变量情况下，等价于多项式的阶为$\mathrm{d}$，则最多有$\mathrm{d}$个根。

**使用随机数进行对签名进行线性组合**，根据 Schwartz–Zippel 引理，发生碰撞的概率可忽略。

## 3.5BBRO 签名

**系统参数：**群 G 的阶为 p，生成元为 g，$e$ 为双线性映射 $e:G\times G\to G_T$，哈希函数 $Hash:\{0,1\}^*\to G$。

**密钥生成：**选择随机数群元素 $h\in G$，选择随机数 $\alpha\in\mathbb{Z}_p$，计算 $g_1=g^\alpha$，则私钥为 $\alpha$，公钥为 $g_1,h$。

**签名：**对于消息 $m\in\{0,1\}^*$，使用私钥 $\alpha$，计算 $\sigma=(\sigma_1,\sigma_2)=(h^\alpha Hash(m)^r,g^r)$

**验证：**使用公钥 $g_1,h$，校验 $e(\sigma_1,g)=e(g_1,h)e(Hash(m),\sigma_2)$

公式推导过程：

$$e(\sigma_1,g)=e(h^\alpha\cdot Hash(m)^r,g)=e(h^\alpha,g)e(Hash(m)^r,g)=e(g_1,h)e(Hash(m),\sigma_2)$$

安全性：如果 CDH 问题是困难的，则该签名不可伪造，归约损失为 $q_{Hash}$，损失 40bit。

## 3.6ZSS 签名

**系统参数：**群 G 的阶为 p，生成元为 g，$e$ 为双线性映射 $e:G\times G\to G_T$，哈希函数 $Hash:\{0,1\}^*\to\mathbb{Z}_p$。

**密钥生成：**选择随机数群元素 $h\in G$，选择随机数 $\alpha\in\mathbb{Z}_p$，计算 $g_1=g^\alpha$，则私钥为 $\alpha$，公钥为 $g_1,h$。

**签名：**对于消息 $m\in\{0,1\}^*$，使用私钥 $\alpha$，计算 $\sigma=h^{1/(\alpha+Hash(m))}$

（**数据很短**，仅一个群元素）

**验证：**使用公钥 $g_1,h$，校验 $e(\sigma,g_1g^{Hash(m)})=e(h,g)$

公式推导过程：$e(\sigma,g_1g^{Hash(m)})=e(h^{1/(\alpha+Hash(m))},g^\alpha g^{Hash(m)})=e(h,g)$

安全性：如果 **q-SDH 问题是困难的**，则该签名不可伪造，归约损失为 $q_{Hash}$，损失 40bit。

标准困难问题是 CDH/DDH/DL，其他都是非标准困难问题。

**签名扩展 1**

**签名：**对于消息 $m \in \{0,1\}^*$，选择随机位 $c \in \{0,1\}$，使用私钥 $\alpha$，计算

$$\sigma = (\sigma_1, \sigma_2) = (c, h^{1/(\alpha + Hash(m,c))})$$

**验证：**使用公钥 $g_1, h$，校验 $e(\sigma_2, g_1 g^{Hash(m,\sigma_1)}) = e(h, g)$

公式推导过程：

$$e(\sigma_2, g_1 g^{Hash(m,\sigma_1)}) = e(h^{1/(\alpha + Hash(m,c))}, g^{\alpha} g^{Hash(m,c)}) = e(h, g)$$

安全性：如果 q-SDH 问题是困难的，则该签名不可伪造，**归约损失为 2，1bit。**

**签名扩展 2**

**签名：**对于消息 $m \in \{0,1\}^*$，选择随机数 $r \in \mathbb{Z}_p$，使用私钥 $\alpha$，计算

$$\sigma = (\sigma_1, \sigma_2) = (r, h^{1/(\alpha + Hash(m,r))})$$

**验证：**使用公钥 $g_1, h$，校验 $e(\sigma_2, g_1 g^{Hash(m,\sigma_1)}) = e(h, g)$

公式推导过程：

$$e(\sigma_2, g_1 g^{Hash(m,\sigma_1)}) = e(h^{1/(\alpha + Hash(m,r))}, g^{\alpha} g^{Hash(m,r)}) = e(h, g)$$

安全性：如果 CDH 问题是困难的，则该签名不可伪造，归约损失为 1（无损失）。

## 3.7BB 短签名

系统参数：群 G 的阶为 p，生成元为 g， $e$ 为双线性映射 $e: G \times G \to G_T$。

密钥生成：选择随机数群元素 $h \in G$，选择随机数 $\alpha, \beta \in \mathbb{Z}_p$，计算 $g_1 = g^{\alpha}, g_2 = g^{\beta}$，则私钥为 $\alpha, \beta$，公钥为 $g_1, g_2, h$。

**签名：**对于消息 $m \in \{0,1\}^*$，选择随机数 $r \in \mathbb{Z}_p$，使用私钥 $\alpha, \beta$，计算

$$\sigma = (\sigma_1, \sigma_2) = (r, h^{1/(\alpha + \beta m + r)})$$

**验证：**使用公钥 $g_1, g_2, h$，校验 $e(\sigma_2, g_1 g_2^{m} g^{\sigma_1}) = e(g, h)$

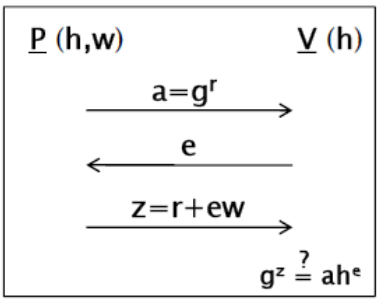
公式推导过程：$e(\sigma_2, g_1 g_2^{m} g^{\sigma_1}) = e(h^{1/(\alpha + \beta m + r)}, g^{\alpha} g^{m\beta} g^{r}) = e(h, g)$

安全性： 如果 **q-SDH 问题是困难的（非标准困难问题）**，则该签名不可伪造，归约损失为 2。

**注意：**私钥 2 个，随机数 1 个，一共 3 个随机因子，不涉及哈希函数。

**密码学核心思想：**

对 NP 问题进行 Sigma 零知识证明 h=g^w，**标准 4 步骤：承诺、挑战、响应、校验。**




**Sigma 零知识证明：**证明知道秘密 w，**且**秘密 w 与公开参数 **h** 满足 NP 关系。

其中，离散对数和大整数因子分解，是典型的 NP 关系。

**数字签名：**证明知道私钥 x，**且**私钥 x 与公开参数**公钥 PK** 满足离散对数 NP 关系。

**所以，得出核心结论：可以认为：Sigma 零知识证明≈数字签名。**

zkSNARK 和 zkSTARK 是对任意关系（NP 关系和 P 关系）的零知识证明，证明的**范围更大**。

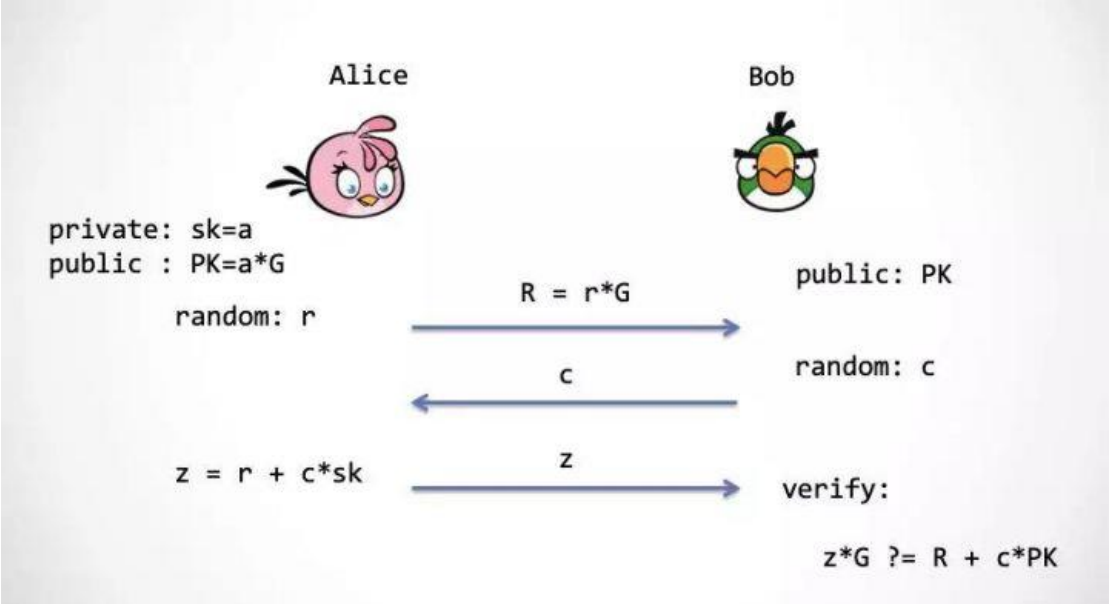
zkSNARK 简洁非交互零知识**论证，**而不是**证明。浅显理解：论证≈证明**

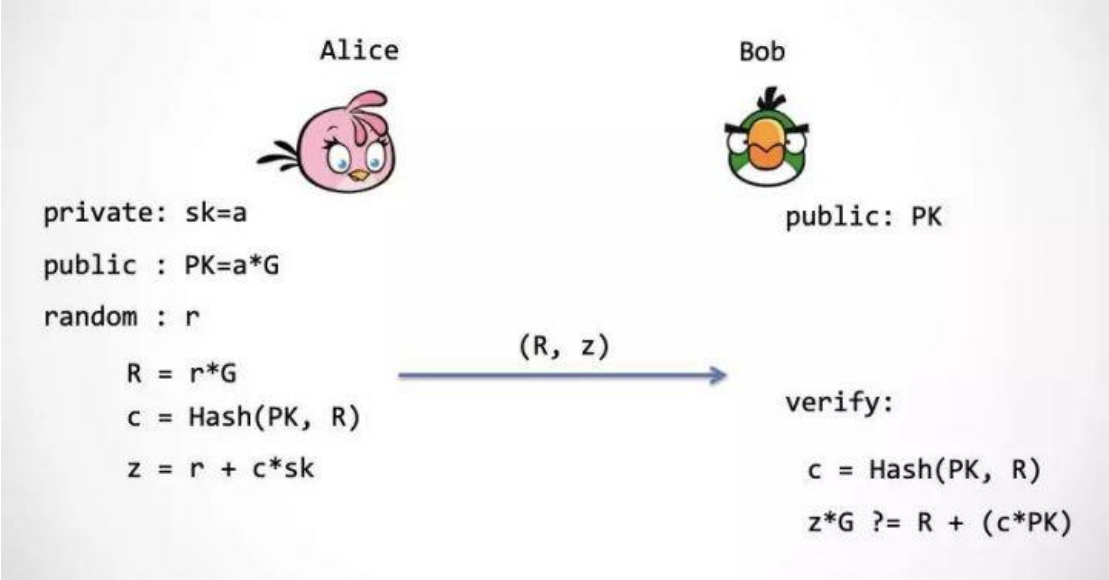
**实际上：零知识证明就是数字签名的进一步扩展。**

**如果扩展可忽略，则零知识证明≈数字签名**

**如果扩展非常多，则零知识证明就是 zkSNARK,成为了一个新工具。**

下图是 schnorr 零知识证明拥有正确的私钥，且私钥与公钥满足离散对数关系。






## 3.8Schnorr 签名

**初始化：**椭圆曲线生成元为$G$，阶为$n$。

**密钥生成：**私钥$u \in [1, n-1]$，公钥为$Y$，满足离散对数关系$Y = u \cdot G$。

**签名：**消息为$m$，**选择随机数**$k \in [1, n-1]$，计算**承诺**$R := k \cdot G$，计算**挑战**$e = hash(m, R)$，

计算**响应**$z = k + e \cdot u \bmod n$，则签名为$(R, z)$

**验证：**输入消息$m$、签名$(R, z)$和公钥$Y$，计算**挑战**$e = hash(m, R)$，校验

$$z \cdot G == R + e \cdot Y$$

一致性过程如下：$z \cdot G = (k + e \cdot u) \cdot G = R + e \cdot Y$

**签名缺点：**对**不同**消息签名，使用相同的随机数 k，则能够解方程求出私钥$u$。

**攻击方法：**

对$m_1$，承诺$R := k \cdot G$，挑战$e_1 = hash(m_1, R)$，响应$z_1 = k + e_1 \cdot u \bmod n$，签名$(R, z_1)$

对$m_2$，承诺$R := k \cdot G$，挑战$e_2 = hash(m_2, R)$，响应$z_2 = k + e_2 \cdot u \bmod n$，签名$(R, z_2)$

攻击者获得签名$(R, z_1), (R, z_2)$，能够计算$e_1 = hash(m_1, R)$和$e_2 = hash(m_2, R)$

解方程$\begin{matrix} z_1 = k + e_1 \cdot u \bmod n \\ z_2 = k + e_2 \cdot u \bmod n \end{matrix}$，计算出私钥$u$。

**k1 k2 u**

## 3.9EdDSA 签名算法

**初始化：**椭圆曲线生成元为$G$，阶为$n$。$c$为曲线常量参数。

**密钥生成：**私钥为$d$，计算$(low_{bit}, hi_{bit}) := \text{sha512}(d)$，令$y = low_{bit}$，公钥为$PK = y \cdot G$

**签名：**消息为$m$，**计算随机数**$k=\text{sha256}(hi_{bit},m) \bmod n$，计算**承诺**$R=k\cdot G$，

计算**挑战**$e:=\text{sha256}(R,PK,m) \bmod n$

计算**响应**$s:=(k+e\cdot y) \bmod n$

**签名**为$(R,s)$

**验证：**重新计算**挑战**$e:=\text{sha256}(R,PK,m) \bmod n$，然后校验

$$2^c s\cdot G == 2^c\cdot R+2^c e\cdot PK$$

**方案优势：**解决了 Schnorr 签名的缺点。不同消息，计算出的随机数 r 肯定不同，不能解方程求 $y$。注释：上述方案来自**维基百科**。

## 3.10ECDSA

### 3.10.1 素数群上的 ECDSA

素数群$G$的阶为$q$，生成元为$g$，哈希函数$H:\{0,1\}^* \to \mathbb{Z}_q$，哈希函数$H':G\to\mathbb{Z}_q$

密钥生成：私钥为随机数$x\in\mathbb{Z}_q$，公钥为$y$，其中$y=g^x$

签名：对于消息 m，选择随机数$k\in\mathbb{Z}_q$，计算承诺$R=g^{k^{-1}}$，计算挑战$r=H'(R)$，计算响应$s=k(m+xr) \bmod q$，则签名为$(r,s)$

验证：校验$(r,s)$的取值范围$\mathbb{Z}_q$，计算$R'=g^{ms^{-1} \bmod q}y^{rs^{-1} \bmod q}$，检测$r=H'(R')$

公式推导过程：

$$s=k(m+xr) \bmod q$$
$$k^{-1}=s^{-1}(m+xr) \bmod q$$
$$g^{k^{-1}}=g^{s^{-1}(m+xr) \bmod q}$$
$$R'=g^{ms^{-1} \bmod q}y^{rs^{-1} \bmod q}=g^{ms^{-1} \bmod q}g^{xrs^{-1} \bmod q}=g^{s^{-1}(m+xr) \bmod q}=g^{k^{-1}}=R$$

### 3.10.2 椭圆曲线群上的 ECDSA

**初始化：**椭圆曲线生成元为$G$，阶为$n$，基域为$F_q$。椭圆曲线点的横坐标和纵坐标的取值空间为$F_q$。

**密钥生成：**私钥$x\in[1,n-1]$和公钥$PK$，满足离散对数关系$PK=x\cdot G$

**签名：**输入消息 $M$，计算 $m := hash(M) \bmod n$；选择随机数 $k \in [1, n-1]$，计算**承诺** $R := k^{-1} \cdot G$，取 $R$ 横坐标**挑战** $r := x_R \bmod n$；计算**响应** $s := k(m + xr) \bmod n$，则签名为 $(r, s)$。

**签名的另一种描述：**选择随机数 $k \in [1, n-1]$，计算**承诺** $R := k \cdot G$，取 $R$ 横坐标为**挑战** $r := x_R \bmod n$；计算**响应** $s := k^{-1}(m + xr) \bmod n$，则签名为 $(r, s)$。

注释：两种描述是一样。

**验证：**输入消息 $M$，计算 $m := hash(M) \bmod n$；校验 $r, s \in [1, n-1]$，计算 $R' := (s^{-1}m) \cdot G + (s^{-1}r) \cdot PK$，取 $R'$ 的横坐标为 $r' := x_{R'} \bmod n$；校验 $r == r'$。如果相等，则接受，否则拒绝。

公式推导过程：

$$
\begin{aligned}
R' &= (s^{-1}m) \cdot G + (s^{-1}r) \cdot PK \\
&= (s^{-1}m) \cdot G + (s^{-1}rx) \cdot G \\
&= \left(s^{-1}(m + rx)\right) \cdot G \\
&= k^{-1} \cdot G \\
&= R
\end{aligned}
$$

**ECDSA 具有延展性，是唯一的延展性（r,s） (r, n-s)**

$$
\begin{aligned}
n - s &= k^{-1}(m + xr) \\
k(n - s) &= (m + xr) \\
k(n - s) \cdot G &= m \cdot G + xr \cdot G \\
-ks \cdot G &= m \cdot G + r \cdot PK \\
-R &= s^{-1}m \cdot G + s^{-1}r \cdot PK
\end{aligned}
$$

计算出 $-R$，纵坐标是负的无所谓，取横坐标得到的就是 $r' := R'\_x \bmod |F_r|$，校验 $r == r'$。如果相等，则接受，否则拒绝。因此，$(r, n-s)$ 是合法签名。既然有 2 个合法签名，所以两方签名里面计算 $s = \min\{s', n - s'\}$ 确定一个小的，就不会有延展攻击了。

**算法缺点：**

重复使用随机数 k 会导致私钥泄露

$$
\begin{aligned}
s_1 &= k(m_1 + xr) \\
s_2 &= k(m_2 + xr)
\end{aligned}
$$

所以通常推荐 k 的计算方法：$k = hash(sk, m)$

所以，ecdsa 可以看做 schnorr 签名的额外规定，规定 k 的计算方法。

**算法分析：**

Schnorr/EdDSA 的 $s := (k + e \cdot y) \bmod n$ 仅有加法，多签协议没难度！

$$s = s_1 + s_2 = (k_1 + k_2) + e(y_1 + y_2) \bmod n$$
$$s_1 = (k_1 + e \cdot y_1) \bmod n$$
$$s_2 = (k_2 + e \cdot y_2) \bmod n$$

ECDSA 的 $s := k(m + xr) \bmod n$ 具有乘法，多签协议难度大！

$$s = (k_1 + k_2)(m + r(x_1 + x_2)) \bmod n$$
$$= (k_1 m + k_2 m + r k_1 x_1 + r k_1 x_2 + r k_2 x_1 + r k_2 x_2) \bmod n$$

需要密码工具：**同态加密**。拆开这两个保密随机数，实现**门限签名**。

## 3.11 门罗币环签名

**初始化：**椭圆曲线群为$\mathbb{G}$，生成元为$G$，阶为$n$。椭圆曲线点的横坐标和纵坐标的取值空间为$F_q$，基域为$\mathbb{F}_q$。哈希函数$H_s : \{0,1\}^* \to \mathbb{F}_q, H_p : \mathbb{G} \to \mathbb{G}$

**密钥生成：**私钥$x \in [1, n-1]$，计算公钥$P = x \cdot G$，计算**密钥镜像**$I = x \cdot H_p(P)$。

*（理解：只能出现一次的假公钥、假身份。）（1）假身份，则已知$I$无法计算公钥$P$，满足匿名性。（2）只能出现一次，出现第 2 次，则是双重花费。*

**签名：**使用 n 个 UTXO，假如 n=5。其中 4 个 UTXO 是其他用户的，用户需要花费的真实 UTXO 是第 3 个。每个 UTXO 对应的公钥为$P_i, i = 1,2,3,4,5$。

这 5 个 UTXO 记为消息$m$，选择 5 随机数$q_1, q_2, q_3, q_4, q_5$，选择 4 个随机数$w_1, w_2, w_4, w_5$（注意没有$w_3$），计算 10 个**承诺**

$$\{L_1, L_2, L_3, L_4, L_5\} = \{q_1 G + w_1 P_1, q_2 G + w_2 P_2, q_3 G, q_4 G + w_4 P_4, q_5 G + w_5 P_5\}$$
$$\{R_1, R_2, R_3, R_4, R_5\} = \{q_1 H_p(P_1) + w_1 I, q_2 H_p(P_2) + w_2 I, q_3 H_p(P_3), q_4 H_p(P_4) + w_4 I, q_5 H_p(P_5) + w_5 I\}$$

密钥镜像使用了 4 次。

计算**挑战**$c = H_s(m, L_1, ..., L_5, R_1, ..., R_5)$

计算**响应**

$$\{c_1, c_2, c_3, c_4, c_5\} = \{w_1, w_2, c - (c_1 + c_2 + c_4 + c_5), w_4, w_5\}$$
$$\{r_1, r_2, r_3, r_4, r_5\} = \{q_1, q_2, q_3 - c_3 \cdot x, q_4, q_5\}$$

环签名为$\sigma = \{I, c_1, ,, .c_5, r_1, ..., r_5\}$。注意：n 越大，签名越长。

验证：验证方**计算**

$$\{L_1', L_2', L_3', L_4', L_5'\} = \{r_1G + c_1P_1, r_2G + c_2P_2, r_3G + c_3P_3, r_4G + r_4P_4, r_5G + r_5P_5\}$$
$$\{R_1', R_2', R_3', R_4', R_5'\} = \{r_1H_p(P_1) + c_1I, r_2H_p(P_2) + c_2I, r_3H_p(P_3) + c_3I, r_4H_p(P_4) + c_4I, r_5H_p(P_5) + c_5I\}$$

密钥镜像使用了 5 次，其中第 3 次必须使用密钥镜像对应的私钥 x。否则验证肯定失败。

**校验：** $c_1 + ... + c_5 == H_s(m, L_1', ..., L_5', R_1', ..., R_5')$

公式推导：**非用户 UTXO 情况下：**

$$L_1 = q_1G + w_1P_1 = r_1G + c_1P_1 = L_1'$$
$$R_1 = q_1H_p(P_1) + w_1I = r_1H_p(P_1) + c_1I = R_1'$$

同理有 $L_2 = L_2', L_4 = L_4', L_5 = L_5'$
$R_2 = R_2', R_4 = R_4', R_5 = R_5'$

**用户 UTXO 情况下：**

$$L_3 = q_3G = r_3G + c_3P_3 = L_3'$$
$$R_3 = q_3H_p(P_3) = (r_3 + c_3 \cdot x)H_p(P_3) = r_3H_p(P_3) + c_3 \cdot xH_p(P_3) = r_3H_p(P_3) + c_3 \cdot I = R_3'$$

**链接：**如果私钥 $x$ 使用第 2 次，则**密钥镜像** $I = x \cdot H_p(P_3)$ 一定会出现第 2 次，则双重花费。

**分析：**

（1）每次支付仅使用一个 UTXO，而不能批量使用 UTXO，应该使用**门限环签名**，每次支付使用多个密钥镜像，即使用多个私钥计算响应，则能够实现多个 UTXO 批量支付，节约存储 gas。

**（2）密钥镜像**唯一对应私钥，使用某个密钥镜像，就必须要知道对应的私钥 $x$。如果不知道，则不能计算出正确的响应 $r_3$。则不能花费该 UTXO。

**（3）密钥镜像**是**唯一标识符**，用于防止双重花费攻击。已知密钥镜像，无法在多项式时间内计算公钥 $P_3$。密钥镜像是唯一标识符，UTXO 和公钥 Pi 可以被任意用户使用多次。

**lynndell 新火科技 密码学专家** lynndell2010@gmail.com