2026-2027 学年短学期

课程综合实践 I：信息安全综合实践

Crypto 专题一：非对称加密与后量子密码

朱柯宇　inuebisu@outlook.com

浙江大学计算机科学与技术学院

CONTENT

目录

对称密码

通信双方事先共享同一个密钥 k。Alice 将明文 m 用密钥 k 加密为密文 c，Bob 收到 c 后用相同的 k 解密得到 m。即

$$c = E(k, m),$$
$$m = D(k, c).$$

加密和解密使用同一个密钥——双方都知道 k，任何不知道 k 的人无法解密密文。

对称密码

通信双方事先共享同一个密钥 k。Alice 将明文 m 用密钥 k 加密为密文 c，Bob 收到 c 后用相同的 k 解密得到 m。即

$$c = E(k, m),$$
$$m = D(k, c).$$

加密和解密使用同一个密钥——双方都知道 k，任何不知道 k 的人无法解密密文。

问题是：Alice 和 Bob 怎么在事先没有共享过秘密的情况下，把 k 安全地传给对方？如果有一条安全的信道可以传密钥，那为什么不直接用那条信道传消息本身？这就是对称密码的密钥分发难题。

非对称密码

每个人生成一对密钥：一个公钥（公开，任何人都可获取），一个私钥（自己秘密保存）。公钥用于加密，私钥用于解密。

$$c = E(pk, m),$$
$$m = D(sk, c).$$

Alice 想给 Bob 发消息——她从公开目录查到 Bob 的公钥，用 Bob 的公钥加密消息；只有持有对应私钥的 Bob 能解密。

任何人都可以向 Bob 发送加密消息，但只有 Bob 能阅读。

对比一下对称密码体制：

1. 密钥分发：对称密码需要先在可信信道上共享密钥；公钥密码不需要事先共享秘密——每个人将公钥公开即可，私钥从不出现在信道上。
2. 密钥数量：n 个人两两通信，对称体制需要 $\frac{n(n-1)}{2}$ 对密钥，公钥体制每人只需一对公私钥，密钥总数 $O(n)$。

$31 \times 29 = ?$

$31 \times 29 = ?$

899。

公钥密码学的引入

$31 \times 29 = ?$

899。

899 是哪两个素数的乘积?

公钥密码学的引入

$31 \times 29 = ?$

899。

899 是哪两个素数的乘积?

试除 2, 3, 5, 7, 11, 13, 17, 19, 23, 29... 答案是 29×31。

$31 \times 29 = ?$

899。

899 是哪两个素数的乘积?

试除 2, 3, 5, 7, 11, 13, 17, 19, 23, 29... 答案是 29×31。

两个三位数的乘法，心算几秒即可完成；而一个三位数的因数分解，即使手算也需要数十次试除。

定义

设映射 $f : X \to Y$。若

- 正向计算 $y = f(x)$ 是容易的；
- 逆向计算 $x = f^{-1}(y)$ 是困难的（在多项式时间内不可行）；
- 但存在辅助信息 t，使得从 (y, t) 恢复 x 是容易的，

则称 f 为带陷门 t 的**单向函数**。

密码学中用困难问题充当单向函数，将陷门作为私钥。

常见的困难问题有三类：

- 大整数分解：已知 $n = pq$（p, q 为大素数），求 p, q；
- 离散对数：已知 $y \equiv g^x \bmod p$ 中的 g, y, p，求 x；
- 格中问题：最短向量问题（SVP）、最近向量问题（CVP）、错误学习问题（LWE）。

密码学中大量运算在模意义下进行。

回顾模运算的基本性质：

模运算性质

设 m 为正整数，a, b, c, d 为整数：

- $a \equiv b \bmod m$ 且 $c \equiv d \bmod m \to a \pm c \equiv b \pm d \bmod m$；
- $a \equiv b \bmod m$ 且 $c \equiv d \bmod m \to ac \equiv bd \bmod m$；
- $ac \equiv bc \bmod m$ 且 $\gcd(c, m) = 1 \to a \equiv b \bmod m$；
- $a \equiv b \bmod m \to a^n \equiv b^n \bmod m$ $(n \in \mathbb{N})$。

定义

若 $ab \equiv 1 \bmod p$，则称 b 为 a 模 p 下的逆元，记作 $a^{-1} \bmod p$。

模意义下的倒数。

裴蜀 (Bézout) 定理

设 a, b 是不全为零的整数。那么

1. 对任意整数 x, y，都有 $\gcd(a, b) \mid ax + by$；
2. 存在整数 x, y 满足 $ax + by = \gcd(a, b)$。

证明：考虑集合 $S = \{au + bv : u, v \in \mathbb{Z}\}$。令 $d = ax_0 + by_0$ 为 S 中最小的正整数。

对任意 $s = au + bv \in S$，带余除法给出 $s = qd + r$，其中 $0 \leqslant r < d$。则

$$r = s - qd = a(u - qx_0) + b(v - qy_0) \in S.$$

由 d 的最小性知 $r = 0$，故 $d \mid s$。即 d 整除 S 中一切元素。特别地，取 $(u, v) = (1, 0)$ 得 $d \mid a$，取 $(u, v) = (0, 1)$ 得 $d \mid b$。故 $d \mid \gcd(a, b)$。

另一方面，$\gcd(a, b)$ 整除 a 和 b，从而整除任意 $au + bv$。特别地，$\gcd(a, b) \mid d$。因此 $d = \gcd(a, b)$。第一部分也同时得证：既然 $d = \gcd(a, b)$ 整除 S 中一切元素，则对任意 x, y 有 $\gcd(a, b) \mid ax + by$。 □

推论

整数 a, b 互素，当且仅当存在整数 x, y 使得 $ax + by = 1$。

证明：（从左到右）若 $\gcd(a, b) = 1$，由 Bézout 定理第二部分直接得存在 x, y 满足 $ax + by = 1$。

（从右到左）若存在 x, y 使 $ax + by = 1$，由 Bézout 定理第一部分知 $\gcd(a, b) \mid 1$，故 $\gcd(a, b) = 1$。 □

定理

a 在模 p 下存在逆元当且仅当 $\gcd(a,p)=1$。

证明：充分性：若 $\gcd(a,p)=1$，由推论存在 x,y 满足 $ax+py=1$。模 p 即得 $ax\equiv 1 \bmod p$，x 即为 a 模 p 的逆元。

必要性：若 a 有模 p 逆元 x，即 $ax\equiv 1 \bmod p$，则存在整数 k 使 $ax-1=kp$，即 $ax+(-k)p=1$。由推论知 $\gcd(a,p)=1$。 □

定理

a 在模 p 下存在逆元当且仅当 $\gcd(a,p)=1$。

证明：充分性：若 $\gcd(a,p)=1$，由推论存在 x,y 满足 $ax+py=1$。模 p 即得 $ax\equiv 1 \bmod p$，x 即为 a 模 p 的逆元。

必要性：若 a 有模 p 逆元 x，即 $ax\equiv 1 \bmod p$，则存在整数 k 使 $ax-1=kp$，即 $ax+(-k)p=1$。由推论知 $\gcd(a,p)=1$。 □

定理

扩展 Euclid 算法可以在 $O(\log \min(a,p))$ 时间内同时计算 $\gcd(a,p)$ 和逆元。

离散对数问题

在整数模 p 乘法群 $\mathbb{Z}_p^*$ 中引入如下问题：

定义

设 G 为有限循环群，g 为 G 的生成元。给定 $h \in G$，求 x 使得 $g^x = h$，称为**离散对数问题**（Discrete Logarithm Problem, DLP）。

离散对数问题

在整数模 p 乘法群 $\mathbb{Z}_p^*$ 中引入如下问题：

定义

设 G 为有限循环群，g 为 G 的生成元。给定 $h \in G$，求 x 使得 $g^x = h$，称为**离散对数问题**（Discrete Logarithm Problem, DLP）。

一个合理的疑问是：g 一定是生成元吗？也即，给定 g 和 p，g 的幂能否「遍历」模 p 的简化剩余系？

记群 $\mathbb{Z}_p^* = \{a \in \mathbb{Z}_p : \gcd(a,p) = 1\}$。当 p 为素数时，$|\mathbb{Z}_p^*| = p-1$。

定义

群 G 中元素 g 的阶，是满足 $g^n = 1$ 的最小正整数 n。

若 g 的阶等于 $p-1$，则 g 是 $\mathbb{Z}_p^*$ 的生成元（也称原根），此时 $\{g^0, g^1, \ldots, g^{p-2}\} = \mathbb{Z}_p^*$。

定理

对任意素数 p，$\mathbb{Z}_p^*$ 是循环群，即存在阶为 $p-1$ 的元素。

离散对数问题

记群 $\mathbb{Z}_p^* = \{a \in \mathbb{Z}_p : \gcd(a,p) = 1\}$。当 p 为素数时，$|\mathbb{Z}_p^*| = p-1$。

定义

群 G 中元素 g 的阶，是满足 $g^n = 1$ 的最小正整数 n。

若 g 的阶等于 $p-1$，则 g 是 $\mathbb{Z}_p^*$ 的生成元（也称原根），此时 $\{g^0, g^1, \ldots, g^{p-2}\} = \mathbb{Z}_p^*$。

定理

对任意素数 p，$\mathbb{Z}_p^*$ 是循环群，即存在阶为 $p-1$ 的元素。

对于实际密码系统，p 取数百甚至数千比特的素数，目前没有多项式时间的算法能在一般情形下求解 DLP。

Diffie–Hellman 密钥交换

回到本讲开头提出的问题：两个从未见过面的人，如何在不安全的信道上协商出一个共享密钥？

对称密码要求双方事先持有 k，而公钥密码的加密方案又不能直接拿来协商密钥。我们需要一种新的协议，它不要求事前共享秘密，却能通过公开的交换让双方得到同一个秘密值。

Diffie–Hellman 密钥交换

考虑 DLP 的一个性质：对任意 a, b，有 $\left(g^a\right)^b = g^{ab} = \left(g^b\right)^a$。如果 Alice 持有 a，Bob 持有 b，两人分别计算 g^a 和 g^b 并发送给对方，那么他们都能独立计算出 g^{ab}。

而窃听者只能看到 g、g^a 和 g^b；计算 g^{ab} 意味着解 CDH 问题，已知 g^a 求 a 意味着解 DLP。

Diffie–Hellman 密钥交换

由这一观察直接得到 Diffie–Hellman 协议：

定义

Diffie–Hellman 密钥交换（Diffie & Hellman, 1976）：

1. 公开参数：大素数 p 和生成元 $g \in \mathbb{Z}_p^*$。
2. Alice 选取随机数 $a \in \{1, 2, \ldots, p-2\}$，计算 $A = g^a \bmod p$，发送 A 给 Bob。
3. Bob 选取随机数 $b \in \{1, 2, \ldots, p-2\}$，计算 $B = g^b \bmod p$，发送 B 给 Alice。
4. Alice 计算 $K = B^a \bmod p = g^{ba} \bmod p$。
5. Bob 计算 $K = A^b \bmod p = g^{ab} \bmod p$。

Diffie–Hellman 密钥交换

由这一观察直接得到 Diffie–Hellman 协议：

定义

Diffie–Hellman 密钥交换（Diffie & Hellman, 1976）：

1. 公开参数：大素数 p 和生成元 $g \in \mathbb{Z}_p^*$。
2. Alice 选取随机数 $a \in \{1, 2, \ldots, p-2\}$，计算 $A = g^a \bmod p$，发送 A 给 Bob。
3. Bob 选取随机数 $b \in \{1, 2, \ldots, p-2\}$，计算 $B = g^b \bmod p$，发送 B 给 Alice。
4. Alice 计算 $K = B^a \bmod p = g^{ba} \bmod p$。
5. Bob 计算 $K = A^b \bmod p = g^{ab} \bmod p$。

被动窃听的敌手能截获 p, g, A, B，恢复 $K = g^{ab}$ 对应求解 CDH 问题。

Diffie–Hellman 密钥交换

上述基于 DLP 的假设可以做更精细的划分：

- **DLP**：给定 g 和 g^a，求 a；
- **CDH**（Computational DH）：给定 g, g^a, g^b，求 g^{ab}；
- **DDH**（Decisional DH）：给定 g, g^a, g^b, g^c，判断 $c = ab \bmod (p-1)$ 是否成立。

可以证明 DLP 的困难性不弱于 CDH，CDH 的困难性不弱于 DDH。在实际安全论证中，通常要求 DDH 假设成立。

ElGamal 加密

DH 协议解决的是密钥协商问题，而 ElGamal（1985）给出了一个直接基于 DLP 的公钥加密方案：

定义

ElGamal 加密：

- 密钥生成：选取大素数 p 和生成元 g。私钥为随机数 $x \in \{1, \ldots, p-2\}$，公钥为 $h = g^x \bmod p$。
- 加密（用公钥 (p, g, h) 加密消息 $m \in \mathbb{Z}_p^*$）：选取随机数 y，计算 $c_1 = g^y \bmod p$，$c_2 = m \cdot h^y \bmod p$；密文为 (c_1, c_2)。
- 解密（用私钥 x）：$m = c_2 \cdot (c_1^x)^{-1} \bmod p$。

ElGamal 加密

DH 协议解决的是密钥协商问题，而 ElGamal（1985）给出了一个直接基于 DLP 的公钥加密方案：

定义

ElGamal 加密：

- 密钥生成：选取大素数 p 和生成元 g。私钥为随机数 $x \in \{1, \ldots, p-2\}$，公钥为 $h = g^x \bmod p$。
- 加密（用公钥 (p, g, h) 加密消息 $m \in \mathbb{Z}_p^*$）：选取随机数 y，计算 $c_1 = g^y \bmod p$，$c_2 = m \cdot h^y \bmod p$；密文为 (c_1, c_2)。
- 解密（用私钥 x）：$m = c_2 \cdot (c_1^x)^{-1} \bmod p$。

正确性验证：$c_2 \cdot (c_1^x)^{-1} = m \cdot h^y \cdot (g^{xy})^{-1} = m \cdot g^{xy} \cdot g^{-xy} = m \bmod p$。

攻击者从公钥 $h = g^x$ 中恢复 x 即面临 DLP。ElGamal 的 IND-CPA 安全性等价于 DDH 假设：若 DDH 在所用群中困难，则 ElGamal 是语义安全的。

DLP 的攻击方法

离散对数是困难问题，但若群的参数选择不当，仍可能遭受攻击。

Shanks 在 1971 年提出的**大步小步算法（BSGS）**是一个通用的、以 $O(\sqrt{n})$ 时间和空间求解任意群中 DLP 的方法。

大步小步算法（BSGS）

在 n 阶群 G 中求解 $g^x = h$：

1. 令 $m = \lceil \sqrt{n} \rceil$。
2. 预计算 $\{g^0, g^1, \ldots, g^{m-1}\}$（Baby steps），存入哈希表。
3. 计算 $u = g^{-m}$，依次检验 $h, hu, hu^2, \ldots$（Giant steps）是否出现在哈希表中。
4. 若 $hu^j = g^i$，则 $x = i + jm$。

证明：将 x 表为 $x = i + jm$，其中 $0 \leqslant i < m$。则

$$g^x = h \Leftrightarrow g^{i+jm} = h \Leftrightarrow g^i = h \cdot (g^{-m})^j = hu^j.$$

等号左边来自 Baby steps，右边来自 Giant steps，两者相遇即给出 x。 □

DLP 的攻击方法

当群的阶 n 的素因子都很小时（$p-1$ 是光滑的），**Pohlig–Hellman 算法**可将 DLP 分解为小阶子群上的 DLP，再用 CRT 合并。

Pohlig–Hellman 算法

设 $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$。

1. 对每个素因子幂 $p_i^{e_i}$，计算 $x \bmod p_i^{e_i}$：将 x 用 p_i 进制展开，逐位确定系数；
2. 使用 CRT 从 $\{x \bmod p_i^{e_i}\}$ 恢复 $x \bmod n$。

DLP 的攻击方法

当群的阶 n 的素因子都很小时（$p-1$ 是光滑的），**Pohlig–Hellman 算法**可将 DLP 分解为小阶子群上的 DLP，再用 CRT 合并。

Pohlig–Hellman 算法

设 $n = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$。

1. 对每个素因子幂 $p_i^{e_i}$，计算 $x \bmod p_i^{e_i}$：将 x 用 p_i 进制展开，逐位确定系数；
2. 使用 CRT 从 $\{x \bmod p_i^{e_i}\}$ 恢复 $x \bmod n$。

具体地，以 $p_i^{e_i} = q^e$ 为例，设 $x \equiv x_0 + x_1 q + x_2 q^2 + \ldots + x_{e-1} q^{e-1} \bmod q^e$。将 $h = g^x$ 两边取 (n/q) 次幂：

$$h^{n/q} = g^{xn/q} = \left(g^{n/q}\right)^{x_0}.$$

这是因为对 $j \geqslant 1$，$\left(g^{n/q}\right)^{x_j q^j} = g^{x_j n q^{j-1}} = 1$。于是 x_0 可以通过在小阶子群（阶为 q）中求解 DLP 得到。后续 $x_1, x_2, \ldots$ 通过类似的迭代消去已知低位来逐次确定。

DLP 的攻击方法

所以攻击者若能让 $p-1$ 的光滑性足够高，DLP 便不再是困难问题。因此安全实现中 $p-1$ 必须包含一个大素因子。

DLP 的攻击方法

所以攻击者若能让 $p-1$ 的光滑性足够高，DLP 便不再是困难问题。因此安全实现中 $p-1$ 必须包含一个大素因子。

小结：DLP 的安全性高度依赖群的选取。

对于 $\mathbb{Z}_p^*$，当 $p-1$ 含有大素因子时 BSGS 和 Pohlig–Hellman 均不可行；但若 $p-1$ 光滑，Pohlig–Hellman 可在多项式时间内求解。实际系统因此通常选用大素数阶子群。

CONTENT

目录

费马小定理

费马 (Fermat) 小定理

若 p 为素数且 $p \nmid a$，则 $a^{p-1} \equiv 1 \bmod p$。

证明：考虑 $a, 2a, ..., (p-1)a$ 这 $p-1$ 个数模 p。

断言它们两两不同余：若 $ia \equiv ja \bmod p$ 且 $p \nmid a$，则 $p \mid (i-j)a$，而 p 是素数且 $p \nmid a$，故 $p \mid (i-j)$。但 $|i-j| < p-1$，故 $i = j$。

因此 $\{a \bmod p, 2a \bmod p, ..., (p-1)a \bmod p\} = \{1, 2, ..., p-1\}$。

将两边所有元素相乘：

$$a \cdot 2a \cdot ... \cdot (p-1)a \equiv 1 \cdot 2 \cdot ... \cdot (p-1) \pmod p.$$

即 $a^{p-1}(p-1)! \equiv (p-1)! \bmod p$。

由于 $\gcd((p-1)!, p) = 1$，两边消去 $(p-1)!$ 即得 $a^{p-1} \equiv 1 \bmod p$。 □

欧拉定理与欧拉函数

定义

对正整数 n，欧拉函数 $\varphi(n)$ 定义为 $\{1, 2, ..., n\}$ 中与 n 互素的元素个数。

欧拉定理与欧拉函数

欧拉函数的计算公式可由容斥原理推出：

欧拉函数的计算

设 $n = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$，则

$$\varphi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right).$$

欧拉函数的计算公式可由容斥原理推出：

欧拉函数的计算

设 $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$，则

$$\varphi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right).$$

证明：在 1 到 n 中，p_i 的倍数有 $\frac{n}{p_i}$ 个，$p_i p_j$ 的倍数有 $\frac{n}{p_i p_j}$ 个，以此类推。由容斥原理：

$$\varphi(n) = n - \sum_i \frac{n}{p_i} + \sum_{i<j} \frac{n}{p_i p_j} - \ldots + (-1)^k \frac{n}{p_1 p_2 \cdots p_k} = n \sum_{I \subseteq \{1,\ldots,k\}} \frac{(-1)^{|I|}}{\prod_{i \in I} p_i}.$$

右端恰好是 $n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)$ 的展开。 □

欧拉定理与欧拉函数

定理

欧拉函数具有积性：若 $\gcd(a,b)=1$，则 $\varphi(ab)=\varphi(a)\varphi(b)$。

证明：$\mathbb{Z}_{ab}^*$ 与 $\mathbb{Z}_a^*\times\mathbb{Z}_b^*$ 之间存在由中国剩余定理给出的双射（见后），故二者阶相等。 □

欧拉定理与欧拉函数

欧拉定理

若 $\gcd(a,n)=1$，则 $a^{\varphi(n)}\equiv 1 \bmod n$。

证明：考虑所有与 n 互素的剩余类构成集合 $U_n=\{u_1,u_2,\ldots,u_{\varphi(n)}\}$。乘以 a 模 n 给出了 U_n 上的一个置换（类似费马小定理证明中的推理）。将 $au_1,au_2,\ldots,au_{\varphi(n)}$ 模 n 后相乘：

$$\prod_{i=1}^{\varphi(n)}(au_i)\equiv\prod_{i=1}^{\varphi(n)}u_i \pmod n.$$

即 $a^{\varphi(n)}\prod u_i\equiv\prod u_i \bmod n$。由于每个 u_i 与 n 互素，乘积亦然，消去即得 $a^{\varphi(n)}\equiv 1 \bmod n$。 □

中国剩余定理 (CRT)

设 $m_1, m_2, ..., m_r$ 两两互素，则同余方程组

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \vdots \\ x \equiv a_r \pmod{m_r} \end{cases}$$

在模 $M = m_1 m_2 ... m_r$ 下有唯一解

$$x = \sum_{i=1}^{r} a_i M_i y_i \pmod{M},$$

其中 $M_i = M/m_i$，$y_i = M_i^{-1} \bmod m_i$。

证明：先构造特殊情形：对固定的 i，求 x_i 满足

$$\begin{cases} x_i \equiv 1 \pmod{m_i} \\ x_i \equiv 0 \pmod{m_j} \\ \quad \forall j \neq i. \end{cases}$$

令 $x_i = M_i y_i$。显然对 $j \neq i$ 有 $m_j \mid M_i$，故 $x_i \equiv 0 \bmod m_j$。由于 $\gcd(M_i, m_i) = 1$（m_i 与所有 $m_j (j \neq i)$ 互素），可取 $y_i = M_i^{-1} \bmod m_i$，此时 $x_i \equiv M_i M_i^{-1} \equiv 1 \bmod m_i$。满足条件的 x_i 构造完毕。之后只需令 $x = \sum_{i=1}^{r} a_i x_i$，由线性叠加即得原方程组的解。唯一性：若 x 和 x' 均是解，则对所有 i 有 $m_i \mid (x - x')$，由 m_i 两两互素知 $M \mid (x - x')$。 □

中国剩余定理

中国剩余定理在 RSA 中有两个重要应用：

1. **解密加速**：解密时不必直接计算 $m = c^d \bmod n$（其中 $n = pq$），而是分别计算 $m_p = c^{d_p} \bmod p$ 和 $m_q = c^{d_q} \bmod q$（其中 $d_p = d \bmod (p-1), d_q = d \bmod (q-1)$），再用 CRT 合成 m。由于模数减半，指数运算提速约 4 倍。
2. **攻击工具**：广播攻击中，同一明文 m 以相同的 e 加密为模不同 n_i 的密文，CRT 可将 e 个方程合并为 $m^e \bmod N$（$N = \prod n_i$），从而退化为小公钥指数攻击场景。

大整数分解问题

RSA 的安全性立足于如下问题的困难性：

大整数分解

给定合数 $n = pq$（p, q 为大素数），求 p 和 q。

目前已知的最优经典分解算法是数域筛法（GNFS），其渐近复杂度为

$$\exp\left(\left(\frac{64}{9}\right)^{\frac{1}{3}}(\ln n)^{\frac{1}{3}}(\ln\ln n)^{\frac{2}{3}}\right),$$

是亚指数级的，对于足够大的 n（2048 比特以上）尚不可行。

然而在实际攻击中，若 p, q 的选取不当，比 GNFS 简单得多的方法就可能成功。

另外，若 Shor 算法（一种量子算法）在足够大规模的量子计算机上运行，分解问题可在多项式时间内求解。因此在量子计算机得到充分发展的未来，RSA 不再安全！这正是后量子密码的驱动力之一。

RSA 加密算法

RSA 加密（Rivest–Shamir–Adleman, 1977）

- 密钥生成：
  - 选取两个大素数 p, q，计算 $n = pq$；
  - 计算 $\varphi(n) = (p-1)(q-1)$；
  - 选择公钥指数 e 满足 $1 < e < \varphi(n)$ 且 $\gcd(e, \varphi(n)) = 1$；
  - 计算私钥指数 $d = e^{-1} \bmod \varphi(n)$；
  - 销毁 p, q，公钥为 (n, e)，私钥为 (n, d)。
- 加密：对于明文 $m \in \mathbb{Z}_n^*$，密文 $c = m^e \bmod n$。
- 解密：$m = c^d \bmod n$。

实践中 e 通常取第 5 个 Fermat 素数 $65537 = 2^{16} + 1$。它的二进制表示中仅含两个 1，因此加密速度快。

这里 $\varphi(n) = (p-1)(q-1)$ 是保密的：任何知道 $\varphi(n)$ 的人可以立即计算出 d，进而解密密文。反之，如果有人知道 $\varphi(n)$，由 $\varphi(n) = n - (p+q) + 1$ 可以求解 $p+q$，结合 $pq = n$ 用韦达定理即可恢复 p, q。因此，计算 $\varphi(n)$ 与分解 n 在计算上是等价的。

RSA 加密算法

RSA 加密算法

RSA 加密（Rivest–Shamir–Adleman, 1977）

- **密钥生成**：
  - 选取两个大素数 p, q，计算 $n = pq$；
  - 计算 $\varphi(n) = (p-1)(q-1)$；
  - 选择公钥指数 e 满足 $1 < e < \varphi(n)$ 且 $\gcd(e, \varphi(n)) = 1$；
  - 计算私钥指数 $d = e^{-1} \bmod \varphi(n)$；
  - 销毁 p, q，公钥为 (n, e)，私钥为 (n, d)。
- **加密**：对于明文 $m \in \mathbb{Z}_n^*$，密文 $c = m^e \bmod n$。
- **解密**：$m = c^d \bmod n$。

实践中 e 通常取第 5 个 Fermat 素数 $65537 = 2^{16} + 1$。它的二进制表示中仅含两个 1，因此加密速度快。

这里 $\varphi(n) = (p-1)(q-1)$ 是保密的：任何知道 $\varphi(n)$ 的人可以立即计算出 d，进而解密密文。反之，如果有人知道 $\varphi(n)$，由 $\varphi(n) = n - (p+q) + 1$ 可以求解 $p+q$，结合 $pq = n$ 用韦达定理即可恢复 p, q。因此，计算 $\varphi(n)$ 与分解 n 在计算上是等价的。

RSA 加密算法

Textbook RSA 也可用来说明签名的基本运算（后文详述）：签名者对消息代表元 M 计算 $s = M^d \bmod n$，验证者检验 $s^e = M \bmod n$。实际方案还需对消息做哈希和编码。

加密和签名是对陷门的两种对偶用法。

RSA 正确性证明

RSA 正确性

若 p, q 为不同素数，$n = pq$，$ed \equiv 1 \bmod \varphi(n)$，则对任意 $m \in \mathbb{Z}_n$，有 $m^{ed} \equiv m \bmod n$。

RSA 正确性证明

证明：由 $ed \equiv 1 \bmod \varphi(n)$，存在整数 k 使得 $ed = 1 + k\varphi(n)$。

情形一：$\gcd(m, n) = 1$。由欧拉定理，$m^{\varphi(n)} \equiv 1 \bmod n$，故

$$m^{ed} = m^{1+k\varphi(n)} = m \cdot \left(m^{\varphi(n)}\right)^k \equiv m \cdot 1^k = m \pmod{n}.$$

情形二：$\gcd(m, n) \neq 1$。由于 $n = pq$ 是两个不同素数的乘积，不妨设 $p \mid m$。此时：

- 模 p 下：$m \equiv 0 \bmod p$，显然 $m^{ed} \equiv 0 \equiv m \bmod p$。
- 模 q 下：因为 $p \mid m$ 而 $p \neq q$，故 $g = \gcd(m, q) = 1$（否则 $q \mid m$，从而 $n = pq \mid m$，但 $m < n$）。由费马小定理 $m^{q-1} \equiv 1 \bmod q$，于是

$$m^{ed} = m^{1+k(p-1)(q-1)} = m \cdot \left(m^{q-1}\right)^{k(p-1)} \equiv m \cdot 1^{k(p-1)} = m \pmod{q}.$$

综上，$p \mid \left(m^{ed} - m\right)$ 且 $q \mid \left(m^{ed} - m\right)$。由于 p, q 是不同的素数，$n = pq \mid \left(m^{ed} - m\right)$，即 $m^{ed} \equiv m \bmod n$。 □

RSA 常见攻击

分解 n 确实是困难的，但参数选取不当会导致典型攻击。

分解攻击

当 p 和 q 的生成存在结构缺陷时，专用分解算法远优于 GNFS。

- **Fermat 分解**：适用于 $|p-q|$ 很小的情况。从 $x=\lceil\sqrt{n}\rceil$ 开始，检验 x^2-n 是否为完全平方数 y^2。若成功则 $n=x^2-y^2=(x+y)(x-y)$，即得 p,q。
- **Pollard $p-1$**：若 $p-1$ 是 B-光滑的（素因子均不超过某个界 B），则对任意与 p 互素的 a，$a^L \bmod p=1$，其中 $L=\mathrm{lcm}(1,2,\ldots,B)$ 或所有不大于 B 的素数幂的乘积。计算 $\gcd(a^L-1,n)$ 即得到 p 的非平凡因子。
- **Williams $p+1$**：与 $p-1$ 方法类似，但作用于 $p+1$ 的光滑性，在 $p-1$ 不光滑但 $p+1$ 光滑时有效。
- **Pollard ρ**：通过迭代 $x_{i+1}=f(x_i) \bmod n$ 构造伪随机序列，当两处 x_i,x_j 模 p 同余但模 n 不同时，$\gcd(|x_i-x_j|,n)$ 即为 p。Floyd 循环检测将空间降为 $O(1)$。

小公钥指数攻击与广播攻击

当 e 很小（如 $e = 3$）且 m 也较小时，$m^e < n$ 可能成立，此时加密未触发模约化，直接对 c 开 e 次方即可恢复 m。若 $m^e = c + kn$ 且 k 不大，可枚举 k 反复尝试。

小公钥指数攻击与广播攻击

当 e 很小（如 $e=3$）且 m 也较小时，$m^e < n$ 可能成立，此时加密未触发模约化，直接对 c 开 e 次方即可恢复 m。若 $m^e = c + kn$ 且 k 不大，可枚举 k 反复尝试。

广播攻击是小指数的延伸：设同一明文 m 以相同的 e、不同的 n_i 加密，得到 $c_i = m^e \bmod n_i$（$i = 1, 2, ..., e$）。用 CRT 求解 $x \equiv c_i \bmod n_i$，在模 $\prod n_i$ 下得到 $x = m^e$。当 $m^e < \prod n_i$ 时（通常成立），$m^e = x$ 作为整数，开 e 次方即得 m。

小私钥指数攻击（Wiener）

当私钥 d 较小时（$d < \frac{1}{3}n^{\frac{1}{4}}$），可以利用连分数理论恢复 d。

由 $ed \equiv 1 \bmod \varphi(n)$，存在整数 k 满足 $ed - k\varphi(n) = 1$。两边除以 $d\varphi(n)$ 并移项：

$$\left|\frac{e}{\varphi(n)} - \frac{k}{d}\right| = \frac{1}{d\varphi(n)}.$$

由于 $\varphi(n) = n - p - q + 1 \approx n$，近似有 $\left|\frac{e}{n} - \frac{k}{d}\right| \approx \frac{1}{d\varphi(n)} < \frac{1}{2d^2}$（当 d 足够小）。Legendre 定理指出：若 $\left|\alpha - \frac{a}{b}\right| < \frac{1}{2b^2}$ 且 $\gcd(a, b) = 1$，则 $\frac{a}{b}$ 必为 α 连分数展开的一个收敛子。由此展开 $\frac{e}{n}$ 为连分数，逐一检验每个收敛子的分母是否为 d，即可在多项式时间内恢复私钥。

Boneh–Durfee 将 Wiener 的界提升至 $d < n^{0.292}$，使用了 Coppersmith 方法（后详）。

共模攻击

若同一模数 n 下以不同的 e_1, e_2 加密同一消息，获得 $c_1 = m^{e_1} \bmod n$ 和 $c_2 = m^{e_2} \bmod n$。当 $\gcd(e_1, e_2) = 1$ 时，Bézout 定理给出 $se_1 + te_2 = 1$，直接计算 $c_1^s c_2^t \equiv m^{se_1+te_2} \equiv m \bmod n$。若 $\gcd(e_1, e_2) = g > 1$，则得到 m^g，当 g 较小时可开根恢复 m。

共模攻击

若同一模数 n 下以不同的 e_1, e_2 加密同一消息，获得 $c_1 = m^{e_1} \bmod n$ 和 $c_2 = m^{e_2} \bmod n$。当 $\gcd(e_1, e_2) = 1$ 时，Bézout 定理给出 $se_1 + te_2 = 1$，直接计算 $c_1^s c_2^t \equiv m^{se_1+te_2} \equiv m \bmod n$。若 $\gcd(e_1, e_2) = g > 1$，则得到 m^g，当 g 较小时可开根恢复 m。

d_p, d_q 泄露

CRT 加速解密使用 $d_p = d \bmod (p-1)$ 和 $d_q = d \bmod (q-1)$。若 d_p 泄露：

由定义 $ed_p \equiv 1 \bmod (p-1)$，即 $ed_p - 1 = k(p-1)$。d_p 已知（与 p 量级相近），k 与 e 量级相近。当 e 较小（如 65537），枚举 k 即可求出 $p-1$，进而得到 p。

相关消息攻击（Franklin–Reiter）

若以相同模数加密两个线性相关的消息 m 和 $m+t$（t 已知），分别得到 c_1, c_2。在多项式环 $\mathbb{Z}_{n[x]}$ 中，$f_1(x) = x^e - c_1$ 和 $f_2(x) = (x+t)^e - c_2$ 都以 m 为公共根，故有公因式 $x - m$。当 e 不大时，直接求 $\gcd(f_1, f_2)$ 即可恢复 m。

相关消息攻击（Franklin–Reiter）

若以相同模数加密两个线性相关的消息 m 和 $m+t$（t 已知），分别得到 c_1, c_2。在多项式环 $\mathbb{Z}_{n[x]}$ 中，$f_1(x) = x^e - c_1$ 和 $f_2(x) = (x+t)^e - c_2$ 都以 m 为公共根，故有公因式 $x - m$。当 e 不大时，直接求 $\gcd(f_1, f_2)$ 即可恢复 m。

高位 / 低位泄露攻击

已知 $m = m_0 \cdot 2^h + m_1$（m_0 已知，m_1 未知且较小），可构造多项式 $f(x) = \left(m_0 \cdot 2^h + x\right)^e - c \bmod n$。需要求解使 $f(x_0) \equiv 0 \bmod n$ 的小根 $x_0 = m_1$。类似地，若 p 的高位泄露，也可构造对应的多项式。

这类问题由 Coppersmith 定理（1996）处理，其核心证明使用了格基规约技术。下一章（格中密码学）将系统给出 Coppersmith 方法的构造与分析。

在 RSA 攻击实践中，SageMath 的 `small_roots()` 方法可直接调用单变元 Coppersmith。

CONTENT

目录

椭圆曲线

RSA 的安全性基于大整数分解，DH / ElGamal 的安全性基于 $\mathbb{Z}_p^*$ 上的 DLP。椭圆曲线密码学则在另一类代数结构上构造公钥密码，其安全性依赖椭圆曲线群上的 DLP。

先在实数平面上考虑三次方程

$$y^2 = x^3 + ax + b$$

的点 (x, y)。当判别式 $\Delta = -16(4a^3 + 27b^2) \neq 0$ 时，曲线没有尖点或自交点——称为非奇异椭圆曲线。

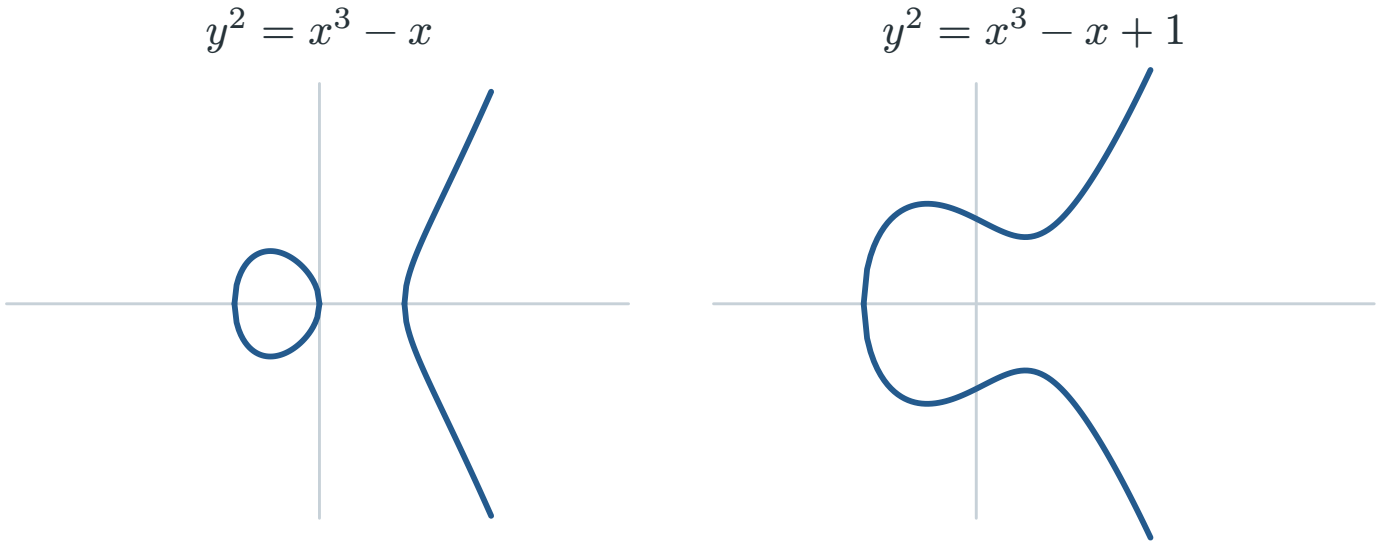

图 1　不同参数下的非奇异椭圆曲线具有不同的实数形态。

椭圆曲线最特别的性质是：可以为曲线上的点赋予一种「加法」运算。取曲线上两点 P, Q，过这两点作直线，该直线与曲线交于第三点 R。取 R 关于 x 轴的对称点 R'，定义 $P + Q = R'$。

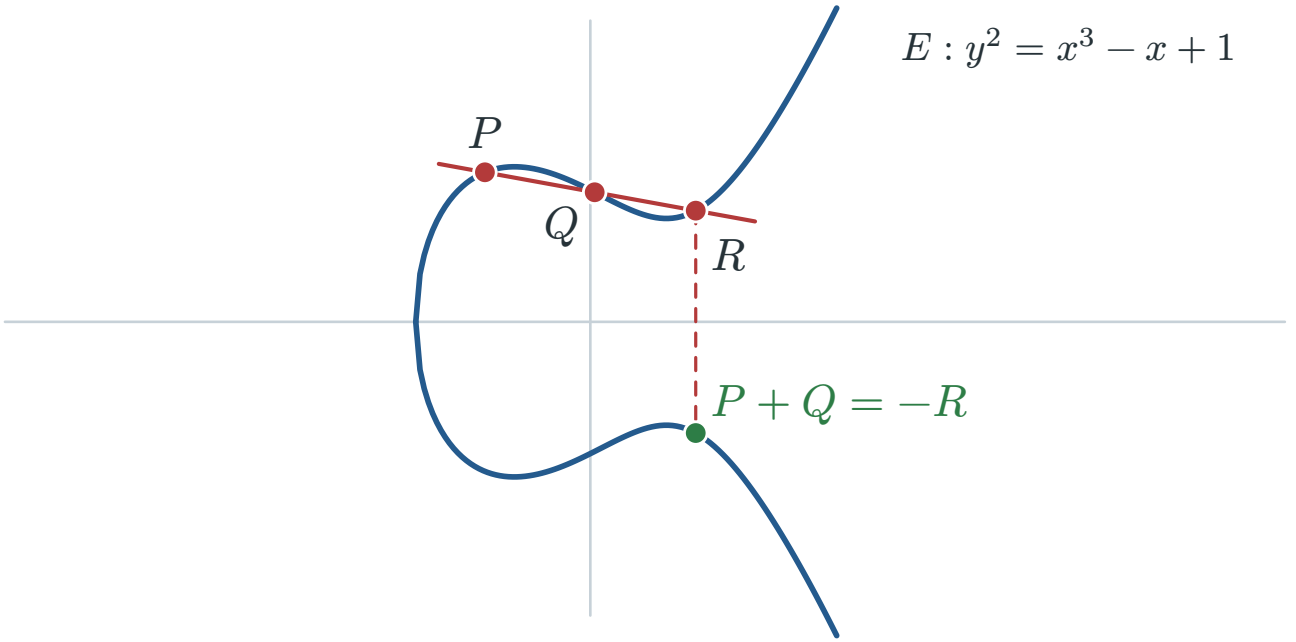

图 2　割线与曲线的第三个交点为 R，关于 x 轴反射得到 $P + Q = -R$。

如果 $P = Q$，则用过 P 的切线代替上述割线——切线交曲线于另一点（重数 2），对称后得到 $2P$，即为倍点运算。

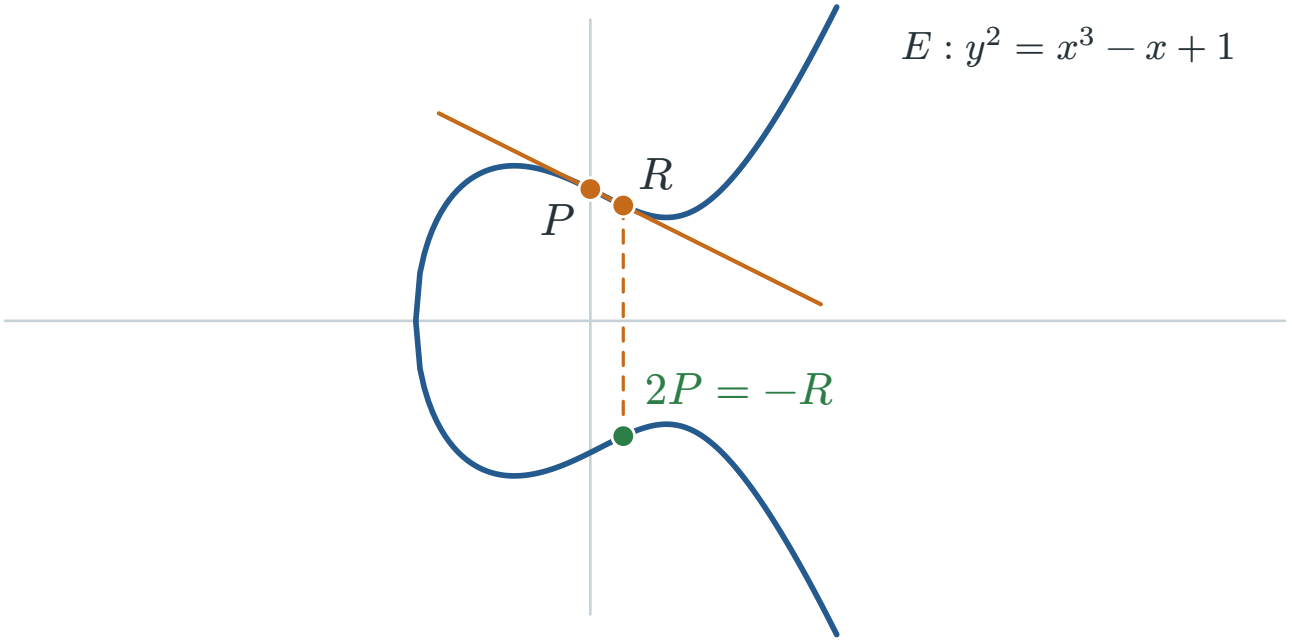

图 3　当两点重合时，以切线替代割线；第三交点反射后得到 $2P$。

无穷远点 $\mathcal{O}$ 充当加法单位元：想象 y 轴方向无限远处的一点，直线与曲线的「竖直」交点即为此点。$P + \mathcal{O} = P$ 成立；$P + (-P) = \mathcal{O}$ 对应竖直直线。

椭圆曲线的代数定义

将几何构造翻译为代数公式，得到 Weierstrass 形式下的显式运算。

定义

设域 K 的特征不为 2 或 3。非奇异椭圆曲线 E 的短 (仿射) Weierstrass 方程为

$$E : y^2 = x^3 + ax + b,$$

其中 $a, b \in K$，且判别式 $\Delta = -16(4a^3 + 27b^2) \neq 0$。

曲线上的点集为 $E(K) = \{(x, y) \in K^2 : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$，其中 $\mathcal{O}$ 为无穷远点。

$\Delta \neq 0$ 保证了曲线是光滑的，即方程 $x^3 + ax + b = 0$ 在代数闭包中没有重根。若 $\Delta = 0$，曲线退化，群结构会破坏（出现不可逆点）。

椭圆曲线上的群结构

定义

点加法（$P \neq \pm Q$）：设 $P=(x_1,y_1)$，$Q=(x_2,y_2)$，$P+Q=(x_3,y_3)$。计算

$$\lambda=\frac{y_2-y_1}{x_2-x_1},\quad x_3=\lambda^2-x_1-x_2,\quad y_3=\lambda(x_1-x_3)-y_1.$$

定义

倍点（$P=Q$ 且 $y_1\neq 0$）：$2P=(x_3,y_3)$，其中

$$\lambda=\frac{3x_1^2+a}{2y_1},\quad x_3=\lambda^2-2x_1,\quad y_3=\lambda(x_1-x_3)-y_1.$$

若 $y_1=0$，则 $2P=\mathcal{O}$（此时切线竖直）。

椭圆曲线上的群结构

(仿射) 椭圆曲线上的点构成加法群

$(E(K), +)$ 是一个加法群，单位元为 $\mathcal{O}$，P 的逆为 $-P$（关于 x 轴对称）

结合律的严格证明涉及代数几何（除子理论），此处从略。交换律可由几何构造的对称性看出；单位元和逆元的性质也可直接验证。上述短 Weierstrass 方程及运算公式适用于特征不为 $2, 3$ 的域。

有限域上的椭圆曲线

密码学中使用有限域 $\mathbb{F}_q$ 上的椭圆曲线，其中 q 可为大素数 p，也可为 2^k。此时点集 $E(\mathbb{F}_q)$ 是有限集合。本节给出的短 Weierstrass 公式直接适用于 $p > 3$ 的素域；特征为 2 或 3 时需使用一般 Weierstrass 方程及相应的运算公式。

Hasse 定理

设 E 为定义在 $\mathbb{F}_q$ 上的椭圆曲线，则

$$\left|\left|E(\mathbb{F}_q)\right| - (q+1)\right| \leqslant 2\sqrt{q}.$$

即曲线上的点数满足 $q + 1 - 2\sqrt{q} \leqslant \left|E(\mathbb{F}_q)\right| \leqslant q + 1 + 2\sqrt{q}$。

直观理解：对每个 $x \in \mathbb{F}_q$，方程 $y^2 = x^3 + ax + b$ 在 $\mathbb{F}_q$ 中有解的 x 约占一半；加上 y 的两个取值（$\pm$）和无穷远点，大致得到 $q + 1$ 个点。

有限域上的椭圆曲线

密码学中使用有限域 $\mathbb{F}_q$ 上的椭圆曲线，其中 q 可为大素数 p，也可为 2^k。此时点集 $E(\mathbb{F}_q)$ 是有限集合。本节给出的短 Weierstrass 公式直接适用于 $p > 3$ 的素域；特征为 2 或 3 时需使用一般 Weierstrass 方程及相应的运算公式。

Hasse 定理

设 E 为定义在 $\mathbb{F}_q$ 上的椭圆曲线，则

$$\left|\left|E(\mathbb{F}_q)\right| - (q+1)\right| \leqslant 2\sqrt{q}.$$

即曲线上的点数满足 $q + 1 - 2\sqrt{q} \leqslant \left|E(\mathbb{F}_q)\right| \leqslant q + 1 + 2\sqrt{q}$。

直观理解：对每个 $x \in \mathbb{F}_q$，方程 $y^2 = x^3 + ax + b$ 在 $\mathbb{F}_q$ 中有解的 x 约占一半；加上 y 的两个取值（$\pm$）和无穷远点，大致得到 $q + 1$ 个点。

$E(\mathbb{F}_q)$ 是一个有限加法群，但未必是循环群。在实际应用中，选择子群 $G \subset E(\mathbb{F}_q)$ 满足 $|G|$ 为大素数 r，运算限制在此子群中进行。这样做的理由有二：(1) 避免 Pohlig–Hellman 降阶攻击；(2) r 的大小直接决定了 ECDLP 的困难程度。

椭圆曲线离散对数问题

定义

设 E 为有限域 $\mathbb{F}_q$ 上的椭圆曲线，$P \in E(\mathbb{F}_q)$，$Q \in \langle P \rangle$。求整数 k 使得 $Q = kP$，称为**椭圆曲线离散对数问题**（ECDLP）。

ECDLP 与 $\mathbb{Z}_p^*$ 上的 DLP 在形式上完全一致——都是在有限循环群中「还原指数」。差异在于：在 $\mathbb{Z}_p^*$ 中，Index Calculus 方法利用整数分解的平滑性求解 DLP，带来了亚指数级的攻击复杂度。而椭圆曲线群中不存在类似的「分解」操作（点与点之间无法像整数那样做素因子分解）。因此，Index Calculus 无法移植到椭圆曲线上，ECDLP 的最优通用攻击（Pollard ρ）复杂度为全指数级 $O(\sqrt{r})$，其中 r 是子群阶。

这一差异带来**密钥长度优势**：有限域 DLP 需使用 3072-bit 的模数才能抵抗数域筛离散对数算法（NFS-DL）；椭圆曲线群中，256-bit 的子群阶可提供约 128-bit 安全强度。

最直接的应用是 ECDH——将 Diffie–Hellman 协议迁移到椭圆曲线群：

定义

ECDH 密钥交换：

1. 公开参数：有限域 $\mathbb{F}_q$ 上的椭圆曲线 E 与基点 P（P 生成大素数阶子群）。
2. Alice 选取私钥 d_A，计算 $Q_A = d_A P$，发送 Q_A 给 Bob。
3. Bob 选取私钥 d_B，计算 $Q_B = d_B P$，发送 Q_B 给 Alice。
4. 共享秘密：$d_A Q_B = d_B Q_A = d_A d_B P$（取 x 坐标或经过 KDF 处理）。

椭圆曲线密码体制

最直接的应用是 ECDH——将 Diffie–Hellman 协议迁移到椭圆曲线群：

定义

ECDH 密钥交换：

1. 公开参数：有限域 $\mathbb{F}_q$ 上的椭圆曲线 E 与基点 P（P 生成大素数阶子群）。
2. Alice 选取私钥 d_A，计算 $Q_A = d_A P$，发送 Q_A 给 Bob。
3. Bob 选取私钥 d_B，计算 $Q_B = d_B P$，发送 Q_B 给 Alice。
4. 共享秘密：$d_A Q_B = d_B Q_A = d_A d_B P$（取 x 坐标或经过 KDF 处理）。

类似地，ElGamal 加密可以自然地迁移为 EC-ElGamal。ECDSA 签名则是对 DSA 的椭圆曲线对应（将在签名章节详述）。

目前 NIST 推荐的标准椭圆曲线包括 P-256、P-384、P-521 等（参数由 NIST 选取）。Curve25519（Bernstein, 2006）是近年来广泛使用的高性能替代方案，其设计经过严格的安全审查，避免了参数选取不可信的风险。

ECDLP 攻击方法

通用攻击：BSGS 和 Pollard ρ（$O(\sqrt{r})$）对任何群成立，椭圆曲线群也不例外。Pohlig–Hellman 同样适用：若子群阶 r 有小的素因子，ECDLP 可被分解。因此安全的椭圆曲线参数要求 r 为素数或几乎素数。

MOV / Frey–Rück 攻击：Menezes–Okamoto–Vanstone（1993）发现，利用 Weil 对或 Tate 对，可以将 ECDLP 约化为 $\mathbb{F}_{q^k}^*$ 上的 DLP。若嵌入次数 k 较小（如 $k = 1, 2$），则此映射可「拉低」到经典 DLP，进而借助 Index Calculus 求解。因此，超奇异椭圆曲线（嵌入次数小）不适用于常规 ECC，但在配对密码（pairing-based cryptography）中有其专门用途。

异常曲线攻击：当 $\left|E(\mathbb{F}_p)\right| = p$（曲线为异常曲线）时，Smart（1999）给出了多项式时间求解 ECDLP 的算法，利用 p-adic 数的提升构造同构。此类曲线在密码实现中必须避开。

小结：ECDLP 的安全条件归结为——在阶接近素数大小的非超奇异、非异常椭圆曲线上，选用 256-bit 以上阶的子群。

CONTENT

目录

线性空间要求系数可以取任意（实）数。

若限制线性空间中的系数只能取整数，得到的离散点集就是格。

定义

设 $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in \mathbb{R}^m$ 为 n 个线性无关的向量。由它们生成的格定义为

$$L = \left\{ \sum_{i=1}^{n} k_i \boldsymbol{v}_i : k_i \in \mathbb{Z} \right\}.$$

$\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\}$ 称为格 L 的一组基，n 为格的维数（秩）。

格的基本概念与几何直观

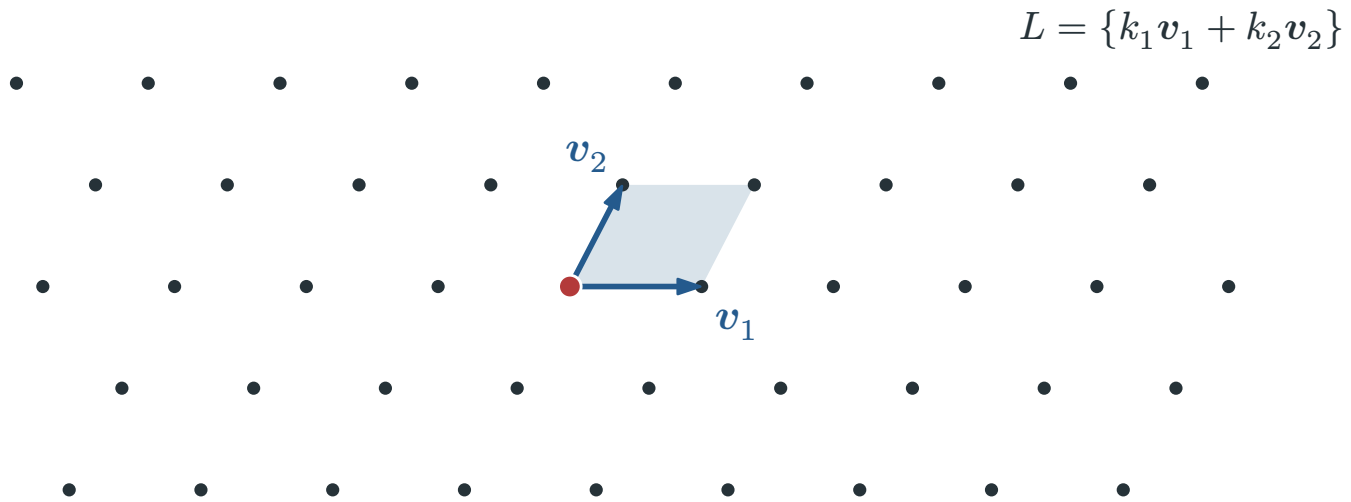

图 4　整数线性组合形成离散点阵；阴影区域是一块基本平行四边形。

格的基本概念与几何直观

格不同于一般向量空间的两个关键性质：

- 系数必须为整数——格中的点是离散分布的，任意两点之间有最小距离；
- 同一个格可以有多组不同的基——不同的基向量线性组合张成的是同一个格（当且仅当两组基之间的变换矩阵为幺模矩阵）。

格的离散性使我们可以问「格中最短的非零向量有多长」。这正是**最短向量问题（SVP）**。

定义

设 n 维格 $L \subset \mathbb{R}^m$ 的基向量按行排成矩阵 $B \in \mathbb{R}^{n\times m}$。格的行列式（determinant）定义为

$$\det(L) = \sqrt{\det(BB^T)}.$$

它等于基本平行多面体在 L 所张成的实向量空间中的 n 维体积。当 $n = m$ 时，上式化为 $\det(L) = |\det(B)|$。

行列式与基的选取无关。若 $B' = UB$ 为另一组基，则 U 为幺模整数矩阵，$\det(U) = \pm 1$。因此

$$\det(B'B'^T) = \det(UBB^TU^T) = \det(BB^T),$$

两组基给出相同的 $\det(L)$。

格基与格的代数性质

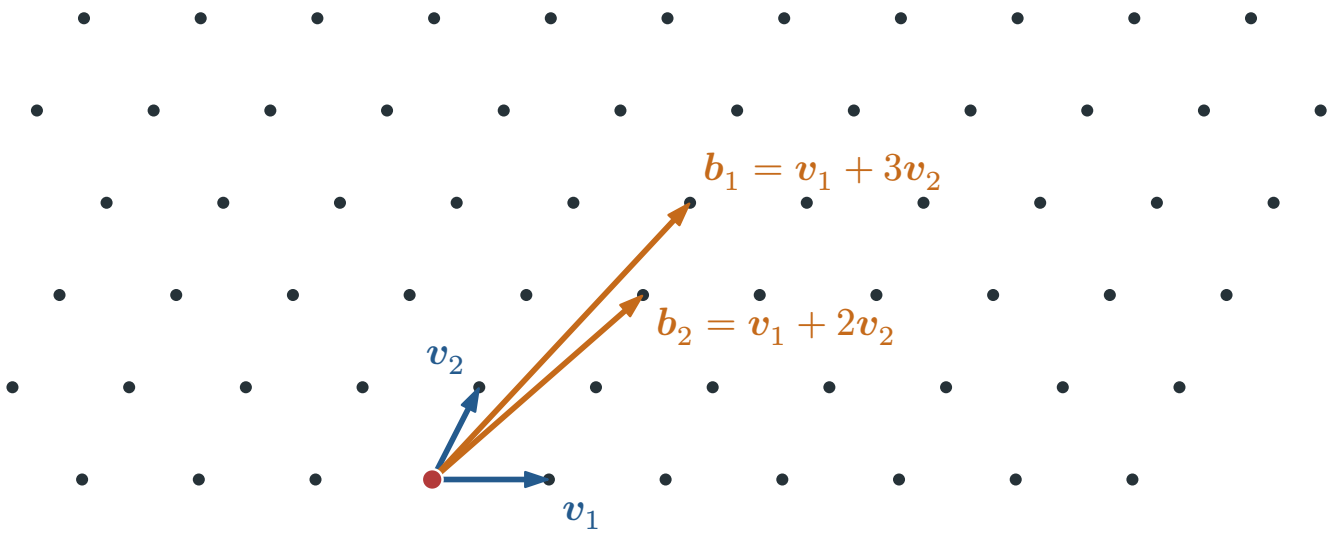

图 5　蓝色短基与橙色长基通过幺模变换联系，张成完全相同的格。

格中困难问题

最短向量问题（Shortest Vector Problem, SVP）

给定格 L 的一组基，求非零向量 $\boldsymbol{v} \in L$ 使得 $\|\boldsymbol{v}\|$ 最小。

SVP 是 NP-hard 的（在确切的近似因子下）。

证明从略，但可以尝试从直观上理解：当格基很「坏」（向量长且方向接近）时，最短向量可能对应基向量的一小整系数组合，枚举这些组合的搜索空间极大。

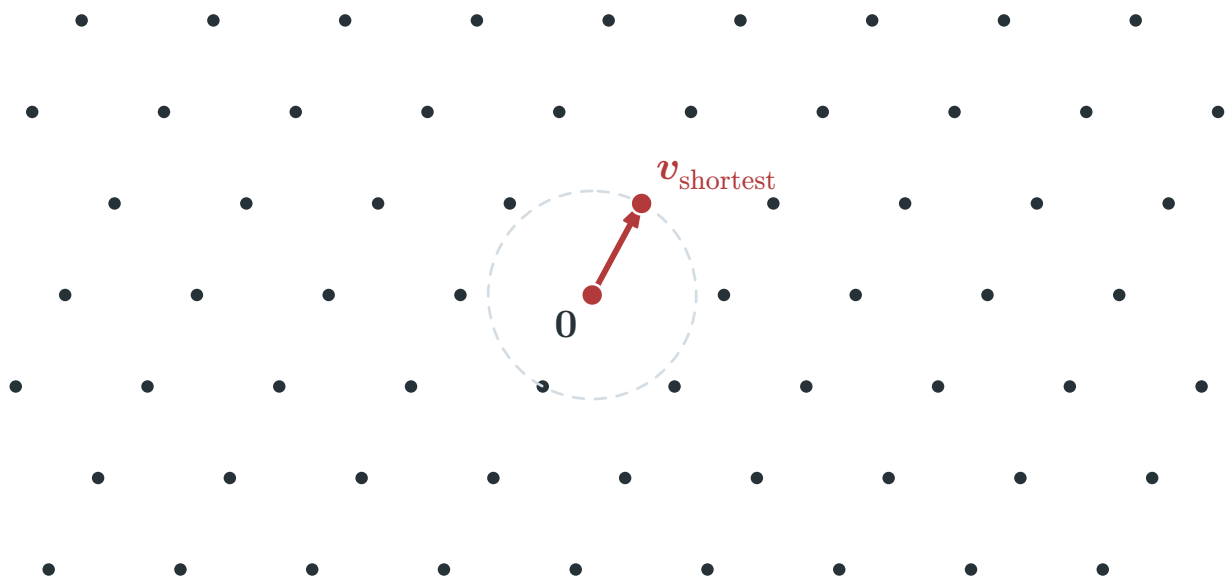

图 6　SVP 寻找离原点最近的非零格点。

最近向量问题（Closest Vector Problem, CVP）

给定格 L 的一组基和目标点 $\boldsymbol{t} \in \mathbb{R}^n$（$\boldsymbol{t}$ 未必在格上），求 $\boldsymbol{v} \in L$ 使得 $\|\boldsymbol{v} - \boldsymbol{t}\|$ 最小。

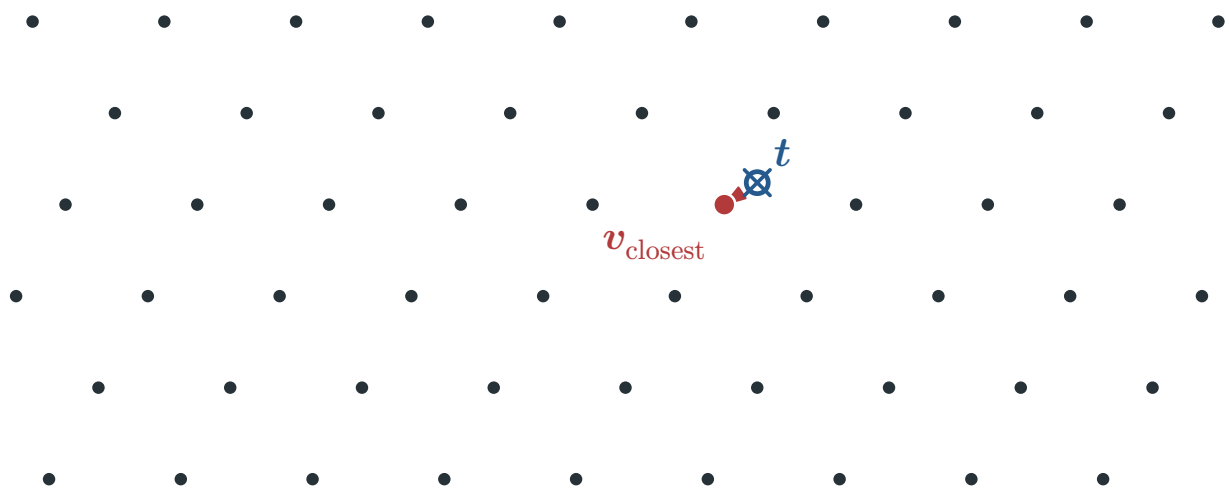

图 7　其实就是寻找距离 $\boldsymbol{t}$ 最近的格点。

SVP 可在多项式时间内归约到 CVP，因此 CVP 难度不低于 SVP。

格中困难问题

错误学习问题（Learning with Errors, LWE）

给定 $A \in \mathbb{Z}_q^{m \times n}$、$\boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e} \bmod q$（$\boldsymbol{s} \in \mathbb{Z}_q^n$ 未知，$\boldsymbol{e}$ 为短噪声向量），求 $\boldsymbol{s}$。

LWE 是格密码中最常用的困难性假设之一。若 $\boldsymbol{e} = \boldsymbol{0}$，问题退化为线性方程组，用 Gauss 消元即可求解；加入小噪声 $\boldsymbol{e}$ 后，求解难度显著上升。Regev（2005）在适当的参数条件下，给出了从最坏情形近似 GapSVP / SIVP 到平均情形 LWE 的量子归约。后续研究（Peikert, Brakerski 等）又对部分参数范围建立了经典归约。

因此，若有算法能高效求解相应的 LWE 实例，便可用它求解归约中的最坏情形格问题。

Hermite 定理与最短向量估计

最短向量到底有多短？以下两个结果提供了上界和期望值（证明从略）：

Hermite 定理

n 维格 L 中必定存在非零向量 $\boldsymbol{v}$ 满足

$$\|\boldsymbol{v}\| \leqslant \sqrt{n} \cdot \det(L)^{\frac{1}{n}}.$$

Hermite 定理给出的界反映了格的「密度」——行列式越大、维数越低，理论上越可能存在较长的最短向量。但该定理未给出寻找最短向量的有效方法。

Hermite 定理与最短向量估计

高斯启发式 (Gaussian Heuristic)

对于随机格，最短非零向量长度的期望值约为

$$\|\boldsymbol{v}(\text{shortest})\| \approx \sqrt{\frac{n}{2\pi e}} \cdot \det(L)^{\frac{1}{n}}.$$

Gaussian Heuristic 刻画了随机格的平均行为。大多数格攻击中，如果目标向量显著短于 Gaussian Heuristic 给出的期望值，那么它很可能是唯一的最短向量，有较大概率被 LLL 或 BKZ 找回。

LLL 格基规约算法

Lenstra–Lenstra–Lovász（1982）提出的 LLL 算法，以多项式时间将任意格基规约为一组「较优」的基——基向量的长度接近最短向量且彼此接近正交。

线性代数中常用的 Gram–Schmidt 正交化可以得到一组正交基，但其系数通常不是整数，因此正交化之后的向量不属于原格。LLL 的思路是在保持整系数组合的前提下，模仿 Gram–Schmidt 结构的优良性质：

- 约束相邻两个 Gram–Schmidt 向量长度不能相差过大（Lovász 条件）；
- 对基向量反复做「取整减法」——从 $\boldsymbol{b}_i$ 中减去 $\boldsymbol{b}_j$ 的适当整数倍以减小长度。

LLL 格基规约算法

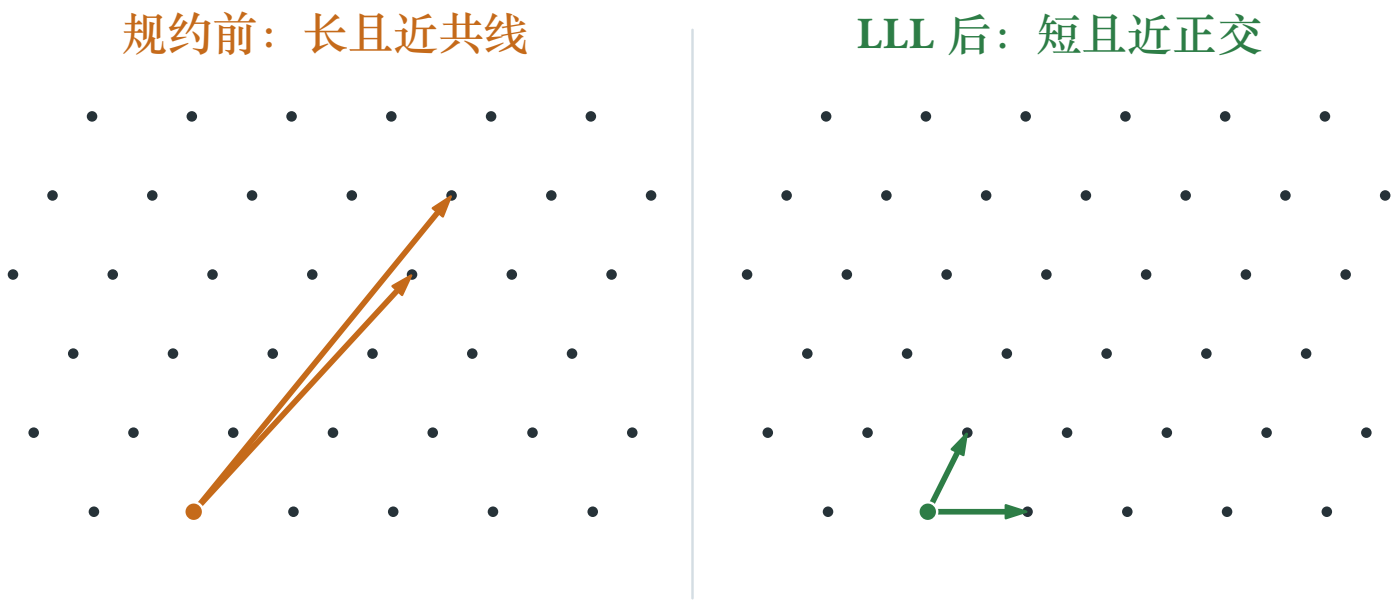

图 8　LLL 保持格不变，同时将长而近共线的基规约为更短、更接近正交的基。

LLL 格基规约算法

LLL 的输出（对满秩 n 维格）满足：

$$\|\boldsymbol{b}_1\| \leqslant 2^{\frac{n-1}{4}} \cdot \det(L)^{\frac{1}{n}},$$

且算法复杂度为 $O\left(n^5(\log B)^3\right)$，其中 B 为基向量长度的界。

BKZ（Block Korkine–Zolotarev）算法以更大的开销换取更强的规约效果，在维度不太高时（$n \leqslant 200$ 量级）实践中优于 LLL。

NTRU

NTRU（Hoffstein–Pipher–Silverman, 1998）是较早进入实用阶段的格密码方案。。

在多项式环 $R = \frac{\mathbb{Z}[x]}{x^N-1}$ 中，选取互素整数 p, q，且 q 远大于 p。密钥生成时选取小系数多项式 f, g，使 f 在模 p 和模 q 下均可逆。记这两个逆元为 f_p^{-1} 和 f_q^{-1}，公钥为

$$h = f_q^{-1} g \pmod{q}.$$

加密时，对小系数消息多项式 m 选取随机小系数多项式 r，计算

$$c = prh + m \pmod q.$$

解密时先计算

$$a = fc = prg + fm \pmod q.$$

将 a 的各个系数中心化到 $(-\frac{q}{2}, \frac{q}{2}]$。若参数使 $prg + fm$ 的系数不越过该区间，中心化后得到它的整数系数表示。随后模 p 约去 prg，再乘 f_p^{-1}，便恢复 $m \bmod p$。

NTRU 的格表示：考虑 $hf \equiv g \bmod q$，存在 k 使 $fh - kq = g$。写成矩阵形式

$$(f, -k)\begin{pmatrix} 1 & h \\ 0 & q \end{pmatrix} = (f, g).$$

向量 (f, g) 位于格 $L = \{(x, xh \bmod q) : x \in \mathbb{Z}\}$ 上。由于 f, g 的系数都很小，(f, g) 很可能是格中最短的向量之一。LLL 规约可望找到 (f, g) 从而恢复私钥。安全参数需确保 f, g 不够短，使得规约在计算上不可行。

LWE 加密

Regev（2005）基于 LWE 问题构建了公钥加密方案。私钥为向量 $\boldsymbol{s} \in \mathbb{Z}_q^n$；公钥为 $(A, \boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e})$。加密时选取随机二进制向量 $\boldsymbol{r}$，计算 $c_1 = A^T\boldsymbol{r}$，$c_2 = \boldsymbol{b}^T\boldsymbol{r} + m \cdot \lfloor \frac{q}{2} \rfloor$。解密时计算 $c_2 - \boldsymbol{s}^T c_1$，由噪声大小判定明文比特。

LWE 的格攻击与 NTRU 类似：构造格

$$L = \begin{pmatrix} A^T & O \\ \boldsymbol{b}^T & 1 \end{pmatrix}$$

使得目标小向量 $(\boldsymbol{e}, 1)$ 位于格中，用 LLL / BKZ 寻找。

NIST 标准 ML-KEM（源自 Kyber）和 ML-DSA（源自 Dilithium）使用 module-LWE / module-SIS 类问题。这些问题在多项式环 $R_q = \frac{\mathbb{Z}_{q[x]}}{x^n+1}$ 上引入模结构，可视为 LWE / SIS 的代数结构化变体。

Coppersmith 方法

Coppersmith 定理（1996）是格基规约在求解模方程小根问题中的重要应用。

Coppersmith 定理（简化单变元形式）

设 $f(x)$ 为模 N 下的 d 次首一多项式。对任意固定的 $\varepsilon > 0$，满足

$$f(x_0) \equiv 0 \pmod{N}, \quad |x_0| < N^{\frac{1}{d}-\varepsilon}$$

的整数根 x_0 可在多项式时间内求出。通过增大格的维数，根的搜索范围可渐近 $N^{\frac{1}{d}}$。

证明的核心工具是 Howgrave-Graham 引理：

Howgrave-Graham 引理

设 $f(x)=\sum_{i=0}^{d} a_i x^i$，定义其系数向量为 $\boldsymbol{b}_f=\left(a_0, a_1 X, \ldots, a_d X^d\right)$，其中 X 为 $|x_0|$ 的上界。若 $f(x_0) \equiv 0 \bmod M$ 且 $\|\boldsymbol{b}_f\|<\frac{M}{\sqrt{d+1}}$，则 $f(x_0)=0$（在整数上成立）。

证明：$|f(x_0)|=\left|\sum_{i=0}^{d} a_i x_0^i\right| \leqslant \sum_{i=0}^{d}|a_i| X^i$。由 Cauchy-Schwarz 不等式，

$$\sum_{i=0}^{d}|a_i| X^i \leqslant \sqrt{d+1} \cdot\|\boldsymbol{b}_f\|<M.$$

故 $-M<f(x_0)<M$。又 $f(x_0) \equiv 0 \bmod M$，因此 $f(x_0)=0$。 □

Coppersmith 方法

利用这一引理，可以先构造一组在 x_0 处模 N 为零的多项式，再用格基规约寻找系数较小的整系数线性组合。若组合后的多项式满足 Howgrave-Graham 的范数条件，模方程的根便同时是该多项式的整数根。

先看一个最小的示意构造。对

$$f(x) = x^d + a_{d-1}x^{d-1} + \ldots + a_0,$$

取 $N, Nx, \ldots, Nx^{d-1}$ 和 $f(x)$。它们在 x_0 处均模 N 为零。设 X 是 $|x_0|$ 的上界。为了使向量范数反映多项式在 $[-X, X]$ 上的大小，将 x^i 项的系数乘以 X^i。缩放后的系数向量构成如下 $(d+1) \times (d+1)$ 矩阵：

$$L = \begin{pmatrix} N & 0 & \ldots & 0 & 0 \\ 0 & NX & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & NX^{d-1} & 0 \\ a_0 & a_1X & \ldots & a_{d-1}X^{d-1} & X^d \end{pmatrix}.$$

$\det(L) = N^d X^{\frac{d(d+1)}{2}}$。

Coppersmith 方法

对这个 $(d+1)$ 维格，LLL 给出的第一条向量满足

$$\|\boldsymbol{b}_1\| \leqslant 2^{\frac{d}{4}} \det(L)^{\frac{1}{d+1}}.$$

只有当右端小于 $\frac{N}{\sqrt{d+1}}$ 时，Howgrave–Graham 引理才能保证对应多项式在 x_0 处的模等式提升为整数等式。因此，LLL 的输出是否足够短，还要由 N、X、d 和格的构造共同决定。

上述示意格给出的搜索界较弱。实际构造会加入 $xf(x), x^2 f(x), \ldots$ 等 x-shift 多项式，并使用 N^k 的倍数来扩大格。随着 shift 数量和幂次增加，可达到 $X < N^{\frac{1}{d}-\varepsilon}$，并渐近 $N^{\frac{1}{d}}$ 的界。

格攻击范式总结

将前述 NTRU、LWE、背包、HNP、Coppersmith 等攻击整理，可以抽取出一个统一的格攻击范式：

定理

1. 识别密码系统中的「小量」——短私钥、噪声项、高位缺失部分；
2. 围绕小量推导整数线性关系（恒等式）；
3. 将相关等式编码为格矩阵——已知量大数嵌入基，小量留在目标向量中；
4. 对格矩阵运行 LLL / BKZ 规约，从规约后的基中检索目标小量；
5. 调整平衡系数，使目标向量的范数落在 Hermite / Gaussian 界以内。

平衡系数的选取是实践中最关键的经验性步骤：目标向量的各分量需调整到大致相同的中值，通常引入对角缩放因子 C_i。若规约结果不理想，可以逐步增大格的维度（加入更多 x-shift 或方程）来改善效果。~~（调参😇）~~

CONTENT

目录

数字签名的引入

到目前为止，我们讨论的公钥密码全部聚焦于**加密**——用公钥加密、私钥解密，目标是隐蔽消息内容。但在真实世界中，仅靠加密远不够。收到一条加密消息，**你怎么知道它真的来自对方，而不是被他人伪造的**？

数字签名的引入

到目前为止，我们讨论的公钥密码全部聚焦于**加密**——用公钥加密、私钥解密，目标是隐蔽消息内容。但在真实世界中，仅靠加密远不够。收到一条加密消息，**你怎么知道它真的来自对方，而不是被他人伪造的**？

考虑现实中的签名：你在合同末尾签字，旁人可以通过比对笔迹来确认是你签的。签名有两个特点：

1. 只有你能签，别人不能；
2. 任何人看见了签名，都能验证是否是你签的。

数字签名的引入

把这个直觉翻译成密码学的语言：

$$\mathrm{Sign}(sk, m) \to \sigma, \quad \mathrm{Verify}(pk, m, \sigma) \to \{\textbf{接受}, \textbf{拒绝}\}.$$

用私钥对消息产生签名（只有签名者持有私钥），用公钥验证签名（公钥人人可获取）。

数字签名的引入

把这个直觉翻译成密码学的语言：

$$\mathrm{Sign}(sk, m) \to \sigma, \quad \mathrm{Verify}(pk, m, \sigma) \to \{\textbf{接受}, \textbf{拒绝}\}.$$

用私钥对消息产生签名（只有签名者持有私钥），用公钥验证签名（公钥人人可获取）。

这里的方向正好和加密相反：加密是公钥加密、私钥解密，签名是私钥签名、公钥验签。两者都属于陷门的应用，只是单向函数的运算方向互逆。

数字签名的引入

定义

数字签名方案由三元组 $(G, \text{Sign}, \text{Verify})$ 构成：

- G：密钥生成算法，输出公钥 pk 和私钥 sk；
- $\text{Sign}(sk, m)$：输出签名 σ；
- $\text{Verify}(pk, m, \sigma)$：输出接受或拒绝。

标准的安全目标为 EUF-CMA（Existential Unforgeability under Chosen Message Attacks）：即使攻击者可以自适应地获取任意选择消息的签名，也不能对一个新消息（未查询过的）伪造出有效签名。

数字签名实现了消息来源认证、内容完整性和不可否认性。TLS 证书、代码签名、操作系统更新验证、区块链交易等等，都建立在数字签名之上。

RSA 签名方案

先考虑 Textbook RSA 签名。将消息直接解释为代表元 $M \in \mathbb{Z}_n^*$，计算 $\sigma = M^d \bmod n$，验证时检验 $\sigma^e \equiv M \bmod n$。

RSA 签名方案

先考虑 Textbook RSA 签名。将消息直接解释为代表元 $M \in \mathbb{Z}_n^*$，计算 $\sigma = M^d \bmod n$，验证时检验 $\sigma^e \equiv M \bmod n$。

Textbook RSA 签名存在明显的安全问题：

1. 若 σ_1, σ_2 分别是 M_1, M_2 的签名，则 $\sigma_1\sigma_2$ 是 $M_1M_2 \bmod n$ 的签名（同态性）；
2. 攻击者可先选取任意 σ，再令 $M' = \sigma^e \bmod n$。此时 (M', σ) 能通过验证，构成一个有效的消息代表元与签名对（无消息攻击）。

Full Domain Hash（FDH）先用全域哈希将消息确定性地映射到 $\mathbb{Z}_n^*$，再应用 RSA 私钥运算。PSS（Probabilistic Signature Scheme）则使用带随机盐的编码。RSA-PSS 是广泛使用的标准 RSA 签名方案。

ElGamal 签名与 DSA

ElGamal 签名使用与 ElGamal 加密相同的公钥结构（$h = g^x$），但公私钥角色互换：

ElGamal 签名与 DSA

ElGamal 签名使用与 ElGamal 加密相同的公钥结构（$h = g^x$），但公私钥角色互换：

DSA（Digital Signature Algorithm）是 ElGamal 签名的变体，工作在 $\mathbb{Z}_p^*$ 的一个素数阶 q 子群中，签名较短且计算更高效。

ECDSA 是 DSA 在椭圆曲线群上的对应版本，同样是 NIST 标准方案，在区块链（如 Bitcoin、Ethereum）中被广泛使用。

随机数重用攻击

随机数重用攻击：

ElGamal / DSA / ECDSA 在签名时要求每次使用不同的随机数 k。若两次签名使用了相同的 k，攻击者可从两个签名对 (r, s_1)、(r, s_2) 中直接恢复私钥。

$$k = (H(m_1) - H(m_2))(s_1 - s_2)^{-1} \pmod{q}.$$

恢复 k 后，由 $s = k^{-1}(H(m) - xr)$ 立即得到 x。这个看似简单的漏洞曾导致 Sony PlayStation 3 签名密钥的泄露（2010 年 ECDSA k 重用事件）。

后量子签名概述

NIST 后量子标准化（2016–2024）最终选定了三类签名方案：

- **ML-DSA**（源自 CRYSTALS-Dilithium）：使用 module-LWE / module-SIS 类假设，是 NIST 给出的主要后量子签名标准。
- **FN-DSA**（源自 Falcon）：基于 NTRU 格，使用 GPV 框架和快速傅里叶采样。它的签名尺寸较小，实现中的采样与浮点计算需要谨慎处理。
- **SLH-DSA**（源自 SPHINCS+）：无状态哈希签名方案，签名较长。其安全分析涉及哈希函数的原像、第二原像、碰撞阻力以及伪随机性质。

后量子签名概述

RSA 和 ECC 分别依赖大整数分解与离散对数问题。Shor 算法使这两类问题在大规模容错量子计算机上失去安全性，因而后量子公钥密码逐步转向格、编码和哈希函数等困难性假设。NIST 于 2022 年公布首批遴选结果，并在 2024 年完成 ML-KEM、ML-DSA 和 SLH-DSA 的首批标准。

格密码之所以被认为抗量子，是因为 Shor 算法所依赖的隐藏子群问题（HSP）框架在非交换群乃至格空间中还未找到对应的高效实例，因此目前认为**格中的 SVP / CVP 在量子计算模型下同样是困难的**。