# 第 8 章：公钥密码
# Public-Key Encryption

赵俊舟

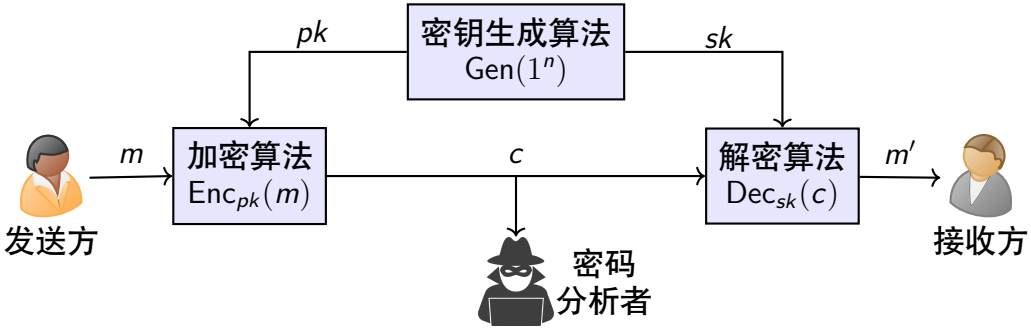
西安交通大学网安学院
junzhou.zhao@xjtu.edu.cn

2025 年 12 月 22 日

# 目录

# 目录

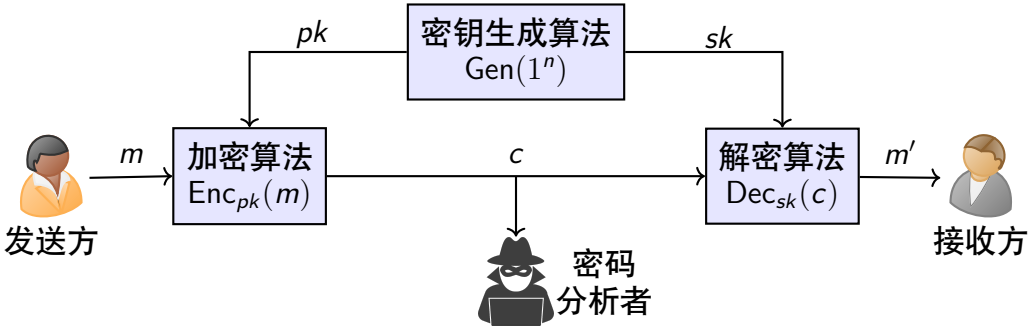
# 公钥密码（Public-Key Encryption）




**由定义在空间 $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ 上的运算 $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ 构成**

- $\mathcal{K}$ 为**密钥空间**，$\mathcal{M}$ 为**明文空间**，$\mathcal{C}$ 为**密文空间**
- $\mathrm{Gen}: \{0,1\}^n \mapsto \mathcal{K} \times \mathcal{K}$ 为**密钥生成函数**
- $\mathrm{Enc}: \mathcal{K} \times \mathcal{M} \mapsto \mathcal{C}$ 为**加密运算**，并且 $c = \mathrm{Enc}_{pk}(m)$
- $\mathrm{Dec}: \mathcal{K} \times \mathcal{C} \mapsto \mathcal{M}$ 为**解密运算**，并且 $m = \mathrm{Dec}_{sk}(c)$

# 对公钥密码的要求




- **正确性**：$\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m$
- **保密性**：由密文或明密文推测密钥和明文，在计算上不可行。
- **计算效率**：加解密算法的计算效率应足够高，便于系统实现。
- **Kerckhoffs 准则**：系统的安全性不依赖于对加解密算法的保密，而是私钥。

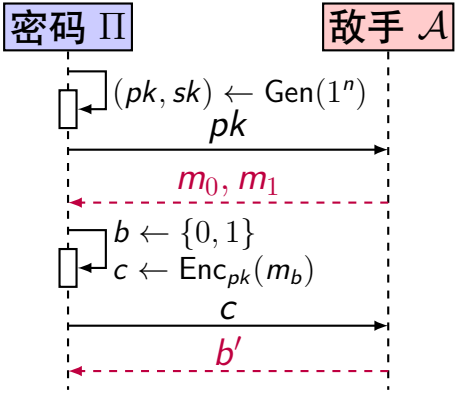
# 目录

# 窃听不可区分与 CPA 安全

- 概率多项式敌手
- 敌手只观察到一条密文
- 如果对于任意 PPT 敌手都存在可忽略函数 negl，使

$$\Pr[\mathsf{PubK}^{eav}_{\mathcal{A},\Pi}(n)=1] \le \frac{1}{2}+\mathsf{negl}(n)$$

则称密码体制 $\Pi$ 满足窃听不可区分或窃听安全（EAV-security）。



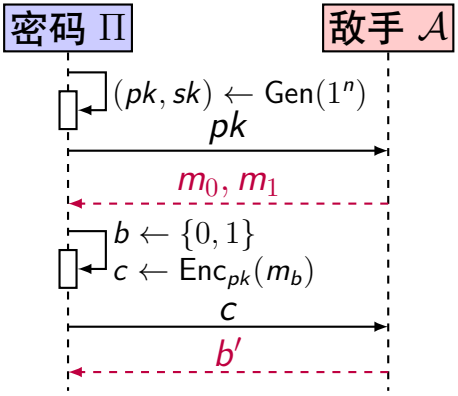

当 $b'=b$ 时，敌手成功，记为 $\mathsf{PubK}^{eav}_{\mathcal{A},\Pi}(n)=1$

# 窃听不可区分与 CPA 安全

- 与私钥密码相比，在公钥密码中，由于敌手知道公钥 $pk$，因此敌手自动具有 CPA 能力。
- 由于敌手具有 CPA 能力，**要求加密算法 $\mathsf{Enc}_{pk}$ 必须为随机函数**。

**例 (选择明文攻击)**

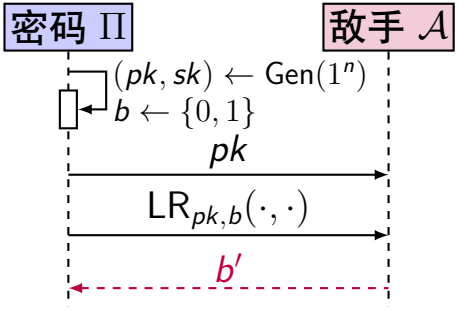
- 假设一门课考试成绩的可能取值为 $\{A, B, C, D, F\}$
- 如果使用确定性加密算法得到某成绩的密文为 $c$
- 敌手可以对所有可能成绩依次加密，然后和 $c$ 比对，进而得知成绩明文。




当 $b' = b$ 时，敌手成功，记为 $\mathsf{PubK}^{eav}_{\mathcal{A},\Pi}(n) = 1$

# 多密文 CPA 安全

- 定义一个 Left-or-Right Oracle $\mathsf{LR}_{pk,b}$：输入一对消息 $m_0, m_1$，计算 $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ 并返回 $c$。




- 当 $b' = b$ 时，敌手 $\mathcal{A}$ 成功，记
$$\mathsf{PubK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1$$
- 如果对于任意 PPT 敌手 $\mathcal{A}$，有
$$\Pr[\mathsf{PubK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$
称密码 $\Pi$ 是多密文 CPA 不可区分或多密文 CPA 安全。

**定理**

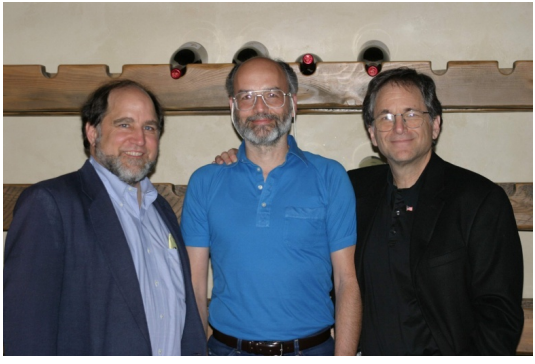
公钥密码 *CPA* 安全等价于多密文 *CPA* 安全。

# 目录

# 目录

# RSA 公钥密码体制

- 1977 年，Ron Rivest、Adi Shamir、Len Adleman 提出了公钥加密算法 RSA，基于 RSA 问题困难性假设。
- 尽管 RSA 算法目前仍被使用，但是由于要达到相同的安全要求，RSA 需要使用较长的密钥，计算开销大。
- RSA 算法正逐渐被基于 DH 困难性假设的椭圆曲线密码取代。



# RSA 公钥密码体制基本算法

- PPT 算法 GenRSA 通过调用 GenModulus，输出由两个 $n$ 比特素数乘积得到的整数 $N$，以及两个整数 $e, d$ 满足 $ed \equiv 1 \pmod{N}$。

**算法 1:** GenRSA$(1^n)$

**输入:** 安全参数 $1^n$

**输出:** $N, e, d$

1 $(N, p, q) \leftarrow \text{GenModulus}(1^n)$

2 $\phi(N) = (p-1)(q-1)$

3 选择 $e > 1$ 且 $\gcd(e, \phi(N)) = 1$

4 计算 $d = e^{-1} \bmod \phi(N)$

5 **return** $N, e, d$

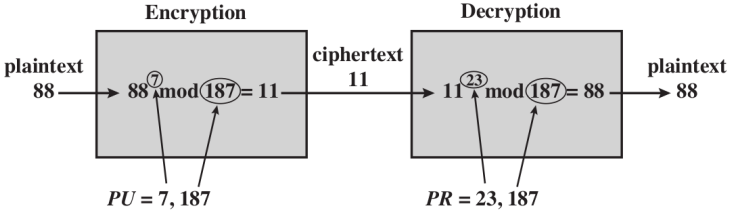
# RSA 公钥密码体制基本算法

## 设计 (RSA 基本算法)

- Gen：输入 $1^n$，调用 GenRSA 得到 $N, e, d$。公钥为 $\langle N, e\rangle$，私钥为 $\langle N, d\rangle$。
- Enc：输入公钥 $\langle N, e\rangle$ 和消息 $m \in \mathbb{Z}_N^*$，输出密文
$$c = m^e \bmod N$$
- Dec：输入私钥 $\langle N, d\rangle$ 和密文 $c$，输出明文
$$m = c^d \bmod N$$

- 正确性：
$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{ed \bmod \phi(N)} \equiv m \pmod N$$
- 安全性依赖于 RSA 问题的困难性假设。

# RSA 算法举例




- GenRSA 输出 $(N, e, d) = (187, 7, 23)$（选择 $p = 17, q = 11$，则 $N = pq = 187$，$\phi(n) = 160$，选择 $e = 7$ 满足 $\gcd(7, 160) = 1$，$d = 23$）；
- 公钥 $\langle 187, 7\rangle$，私钥 $\langle 187, 23\rangle$；
- 对明文 $m = 88$ 加密，得到 $c = 88^7 \bmod 187 = 11$；
- 对密文 $c = 11$ 解密，得到 $m = 11^{23} \bmod 187 = 88$。

# 目录

# RSA 安全性

- RSA 的安全性依赖于 RSA 问题困难性假设，敌手无法由公钥推测私钥，但这只是密码体制安全的必要条件。
- RSA 基本算法的加密运算是确定性的，显然不满足 CPA 安全的要求。
- 此外，RSA 基本算法还容易受到其他攻击。

# 一种低时间复杂度的攻击方法

- 假设消息 $m < 2^n$，则可能存在 $1 < r \leq s \leq 2^{n/2}$ 使 $m = rs$。
- 例如，$n = 64$，存在满足上述条件的 $r, s$ 的概率为 $0.34$。
- 下述算法找到 $r, s$ 使 $c \equiv (rs)^e \pmod N$，复杂度 $O(\sqrt{2^n})$。

---

**输入:** 公钥 $\langle N, e \rangle$，密文 $c$，消息范围 $2^n$
**输出:** $m$ 满足 $c \equiv m^e \pmod N$

```
1 set T = 2^{n/2}
2 for r = 1, ..., T do x_r = c / r^e mod N
3 sort the pairs {(r, x_r)}_{r=1}^{T} by their second component
4 for s = 1, ..., T do
5     if s^e mod N = x_r for some r then
6         return rs mod N
```

# 短消息小加密密钥时的攻击方法

- 公钥加密时习惯使用小的加密密钥以加速加密计算过程，例如 $e = 3$ 等。
- 如果加密的消息也比较小，例如 $m < N^{1/e}$，此时加密运算的模 $N$ 运算不起作用，因为 $m^e < N$。
- 此时密文 $c = m^e$，可以直接对密文 $c$ 求其 $e$ 次方根得到消息 $m$。
- 求一个数的 $e$ 次方根的计算复杂度为 $\text{poly}(\|N\|)$。
- 例如，$e = 3, \|N\| = 2048$，即使消息 $m$ 是任意 $256$ 长的比特串，该攻击方法也是奏效的。

# 已知部分消息时的攻击方法

**定理 (Coppersmith's Theorem)**

*Let $p(x)$ be a polynomial of degree $e$. Then in time* $\mathrm{poly}(\|N\|, e)$ *one can find all $m$ such that $p(m) = 0 \bmod N$ and $|m| \leq N^{1/e}$.*

- 假设发送的消息 $m = m_1 \| m_2$ 其中 $m_1$ 已知，$m_2$ 未知。
- 假设 $m_2$ 是一个 $k$ 长的比特串，则 $m = 2^k m_1 + m_2$。
- 假设 $e = 3$，当截获密文 $c = (m_1 \| m_2)^3 \bmod N$ 时，定义
$$p(x) = (2^k m_1 + x)^3 - c$$
- 从而，敌手可以利用上述定理解得 $p(x) = 0$ 的根。
- 当 $m_2$ 已知，$m_1$ 未知时，也可以用类似的方法攻击。

# 发送相关消息时的攻击方法

- 假设发送方发送了两个相关的消息给同一接收方，例如使用接收方相同的加密密钥加密消息 $m$ 和 $m+\delta$，其中 $m$ 未知，$\delta$ 已知。
- 敌手因此得到两个密文

$$c_1 = m^e \bmod N \qquad c_2 = (m+\delta)^e \bmod N$$

- 定义两个多项式

$$f_1(x) = x^e - c_1 \bmod N \qquad f_2(x) = (x+\delta)^e - c_2 \bmod N$$

- 因为这两个多项式存在相同的根 $x = m$，因此存在公因式 $x - m$。
- 可以利用类似于欧几里得算法的方法计算两个多项式的公因式，计算复杂度为 $\mathrm{poly}(\|N\|, e)$。

# 发送相同消息给不同接收方时的攻击方法

- 假设 $e = 3$，相同的消息 $m$ 发送给三个不同接收方，其公钥分别为 $\langle N_1, 3\rangle, \langle N_2, 3\rangle, \langle N_3, 3\rangle$。
- 假设 $\gcd(N_i, N_j) = 1, i \neq j$（否则，我们找到了因子，可以直接计算出某个接收方的私钥）。
- 敌手得到三条密文

$$c_1 = m^3 \bmod N_1 \quad c_2 = m^3 \bmod N_2 \quad c_3 = m^3 \bmod N_3$$

- 令 $x = m^3, N = N_1 N_2 N_3$，上式其实构成了一元线性同余方程组，可以利用中国余数定理高效求解。

# 共模攻击

- 假设发送方向两个不同接收方发送同一条消息 $m$，且两个接收方的模数相同，接收方公钥分别为 $\langle N, e_1 \rangle$ 和 $\langle N, e_2 \rangle$，且 $\gcd(e_1, e_2) = 1$。
- 敌手得到两个密文

$$c_1 = m^{e_1} \bmod N \qquad c_2 = m^{e_2} \bmod N$$

- 敌手利用扩展欧几里得算法可以得到 $xe_1 + ye_2 = 1$，于是

$$m \equiv m^1 \equiv m^{xe_1+ye_2} \equiv (m^{e_1})^x (m^{e_2})^y \equiv c_1^x c_2^y \pmod{N}$$

- 从而得到明文 $m$。

# 目录

# RSA 填充方案

- RSA 基本算法由于使用确定性加密算法，不满足 CPA 安全。
- 为了向加密运算中引入随机性，一种简单做法是在将消息 $m$ 编码为 $\mathbb{Z}_N^*$ 中的整数时采用随机化填充。
- 随机化填充：在加密消息 $m$ 前，随机选一个 $\ell$ 长的比特串 $r \in \{0,1\}^\ell$，并附在消息前面，得到 $\hat{m} = r \| m$，作为 $\mathbb{Z}_N^*$ 中的整数。
- 这种从消息到 $\mathbb{Z}_N^*$ 中整数的映射显然也是可逆的。

# 采用随机化填充的 RSA 密码体制

## 设计 (随机化填充 RSA 密码体制)

令 $\ell$ 为参数为 $n$ 的函数且 $\ell(n) < 2n$，定义密码体制 $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$：

- Gen：输入 $1^n$，调用 GenRSA 得到 $N, e, d$。公钥为 $\langle N, e\rangle$，私钥为 $\langle N, d\rangle$。
- Enc：输入公钥 $\langle N, e\rangle$ 和消息 $m \in \{0,1\}^{\|N\|-\ell(n)-1}$，选择随机串 $r \in \{0,1\}^{\ell}$ 并且得到 $\hat{m} = r\|m \in \mathbb{Z}_N^*$，输出密文
$$c = \hat{m}^e \bmod N$$
- Dec：输入私钥 $\langle N, d\rangle$ 和密文 $c$，计算
$$\hat{m} = c^d \bmod N$$
输出 $\hat{m}$ 的后 $\|N\| - \ell(n) - 1$ 比特。

# RSA PKCS #1 V1.5

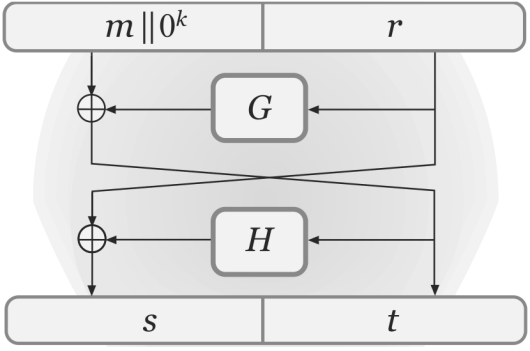
- RSA 实验室于 1993 年发布 RSA PKCS #1 V1.5，基于随机化填充方案。
- 用 $k$ 表示模数 $N$ 的字节数。
- 假设消息 $m$ 的最大字节数为 $k-11$，按如下方式加密一个 $D$ 字节的消息 $m$:

$$(0x00 \| 0x02 \| r \| 0x00 \| m)^e \bmod N$$

- 其中 $r$ 是一个 $k-D-3$ 字节的随机串，不含 $0x00$ 字节，最小长度为 $8$ 字节。
- RSA PKCS #1 V1.5 被发现不满足 CPA 安全，容易受到短消息攻击。

# RSA-OAEP 最优非对称加密填充

- RSA-OAEP 于 1994 年提出，并于 2002 年成为 RSA PKCS #1 v2.1 标准，证明满足 CCA 安全。
- 令 $G\colon \{0,1\}^k \mapsto \{0,1\}^{\ell+k}$ 和 $H\colon \{0,1\}^{\ell+k} \mapsto \{0,1\}^k$ 为两个哈希函数。
- 采用一个两轮的 Feistel 网络将消息映射为 $\mathbb{Z}_N^*$ 中的整数。




- $m' = m\|0^k, r \leftarrow \{0,1\}^k$
- $t = m' \oplus G(r), s = r \oplus H(t)$
- $\hat{m} = s\|t, c = \hat{m}^e \bmod N$

# 目录

# 目录

# 椭圆曲线算术（实数域）

- 椭圆曲线是由威尔斯特拉斯方程所确定的平面曲线：

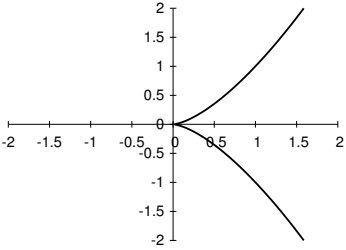
$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

考虑其标准形式：

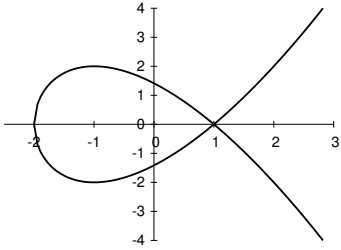
$$y^2 = x^3 + ax + b$$

称满足上述方程的序偶 $(x, y)$ 为椭圆曲线 $E(a, b)$ 上的点。

- 要求 $a, b$ 满足条件 $4a^3 + 27b^2 \neq 0$ 使 $x^3 + ax + b = 0$ 不含重根；否则椭圆曲线在重根处不存在切线。

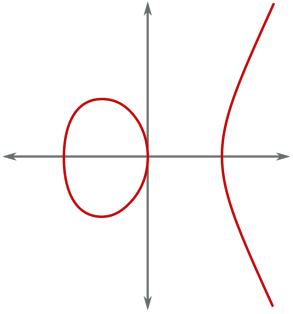



$E(0, 0)$ 在 $(0, 0)$ 点不光滑

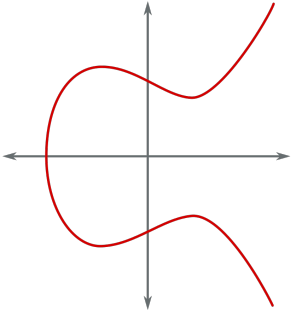



$E(-3, 2)$ 在 $(1, 0)$ 点有交叉

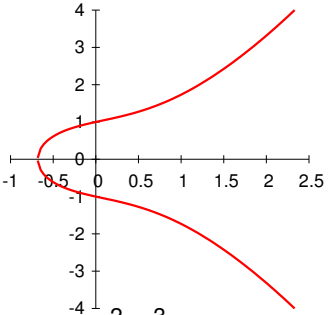
# 椭圆曲线举例



$y^2 = x^3 - x$



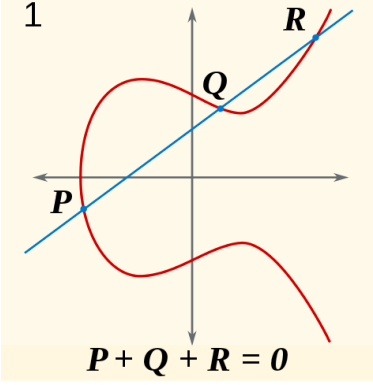
$y^2 = x^3 - x + 1$




$y^2=x^3+x+1$

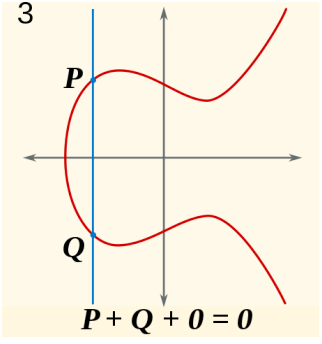
# 椭圆曲线上形式加法的定义

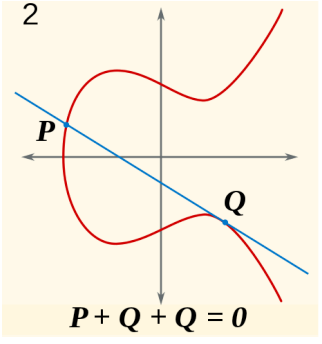
- 椭圆曲线包括一个称为无穷远点或零点的元素，记为 $O$；
- $O$ 是加法的零元，对于椭圆曲线上的任意点 $P$，满足 $P + O = O + P = P$；
- 如果椭圆曲线上三个点处于一条直线上，那么它们的和为 $O$；

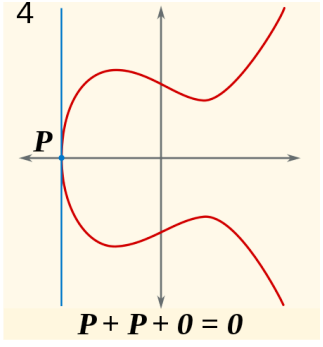


# 椭圆曲线上形式加法的定义

- 一条垂直线与曲线相交于 $P=(x,y)$ 和 $Q=(x,-y)$，也相交于无穷点 $O$，有 $P+Q+O=O$，称 $Q=-P$ 为 $P$ 的负元；
- 在点 $Q$ 处画一切线求出另一交点 $P$，则 $Q+Q+P=O$，即 $2Q=-P$；
- 椭圆曲线上的点及其上的形式加法构成一个 Abel 群。

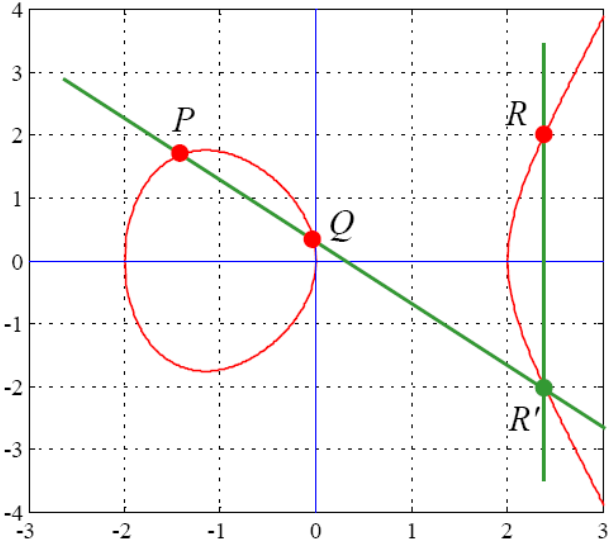


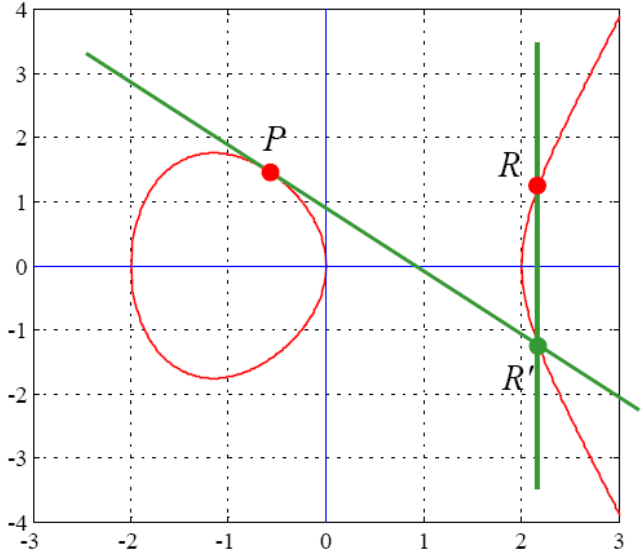
# 加法

$R = P + Q$ （或 $R = P \cdot Q$）



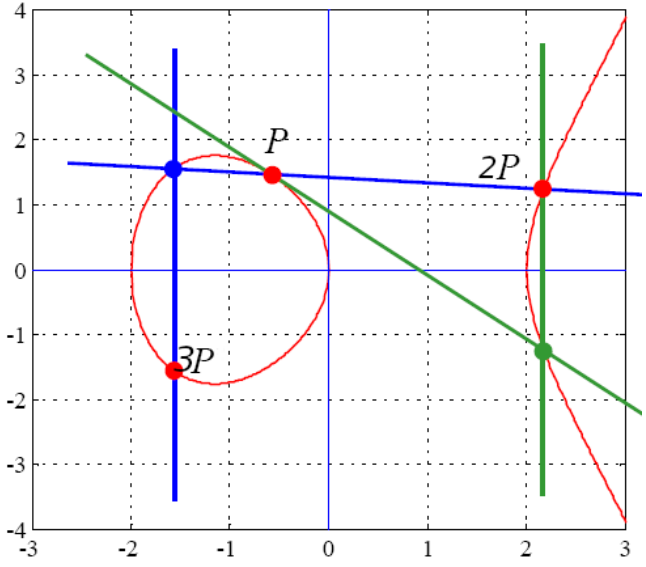
# 累加

$R = P + P = 2P$ （或 $R = P^2$）



# 累加

$R = P + P + P = 3P$ （或 $R = P^3$）



# 计算直线与椭圆曲线交点

- 经过点 $P$ 和 $Q$ 的直线：
  $y = sx + y_0$，其中

$$s = \frac{y_Q - y_P}{x_Q - x_P}$$

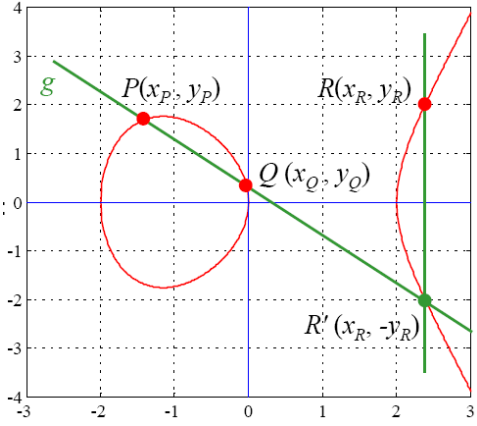
$$y_0 = y_P - sx_P$$

- 直线与曲线的另一个交点：

$$(sx + y_0)^2 = x^3 + ax + b$$

得到 $R$ 点坐标：

$$\begin{cases} x_R &= s^2 - x_P - x_Q \\ y_R &= -(sx_R + y_0) \end{cases}$$



# 计算切线与椭圆曲线交点

- $P$ 点的切线：$y = sx + y_0$，其中

$$s = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

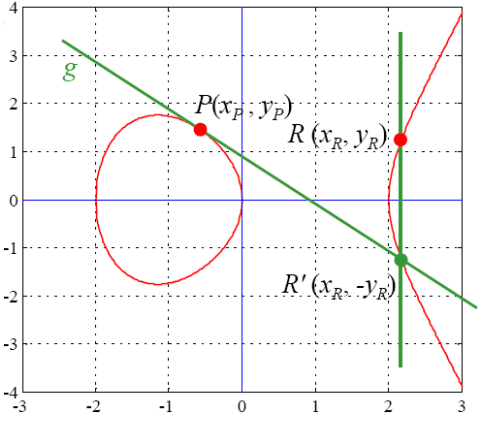
$$y_0 = y_P - sx_P$$

- 切线与曲线交点：

$$(sx + y_0)^2 = x^3 + ax + b$$

得到 $R$ 点坐标：

$$\begin{cases} x_R &= s^2 - 2x_P \\ y_R &= -(sx_R + y_0) \end{cases}$$



# 目录

# 有限域上的椭圆曲线

- 椭圆曲线密码体制使用的是变元和系数均为有限域中元素的椭圆曲线。
- 定义在 $\mathrm{GF}(p)$ 上的椭圆曲线 $E_p(a, b)$：

$$y^2 = (x^3 + ax + b) \bmod p$$

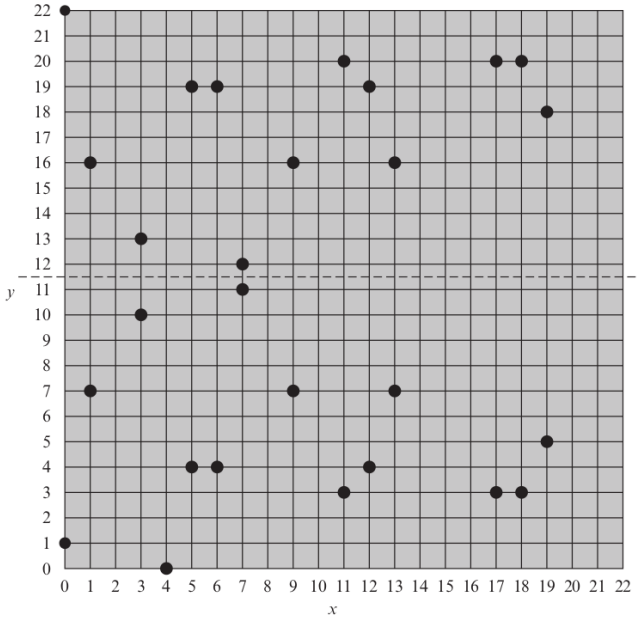
其中变元和系数均取自集合 $\mathbb{Z}_p$，模 $p$ 运算。

# 椭圆曲线 $E_{23}(1,1)$ 上的点

- 对于每个 $x \in \mathbb{Z}_p$，计算 $y^2 = x^3 + x + 1 \bmod p$；
- 对每个结果确定它是否有一个模 $p$ 的平方根；
- 如果没有，在 $E_{23}(1,1)$ 中就没有具有这个 $x$ 值的点；
- 如果有，就有两个满足平方根是 $y$ 的值（除非这个值是单个的 $y$ 值 $0$）。这些点就是 $E_{23}(1,1)$ 上的点（外加无穷远点）。

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

# 椭圆曲线 $E_{23}(1,1)$ 上的点



# 椭圆曲线点加运算

- 将椭圆曲线 $E_{11}(1,6)$ 上的点 $P=(2,4)$ 反复累加
- 计算 $2P=P+P$

$$\begin{cases} s &= \frac{dy}{dx}=\frac{3x_P^2+a}{2y_P} \\ y_0 &= y_P - sx_P \end{cases} \qquad \begin{cases} x_R &= s^2-2x_P \\ y_R &= -(sx_R+y_0) \end{cases}$$

- 计算 $3P=P+P+P=2P+P$

$$\begin{cases} s &= \frac{y_Q-y_P}{x_Q-x_P} \\ y_0 &= y_P - sx_P \end{cases} \qquad \begin{cases} x_R &= s^2-x_P-x_Q \\ y_R &= -(sx_R+y_0) \end{cases}$$

- 所有运算均在 $\mathrm{GF}(11)$ 上进行。

# 椭圆曲线点加运算

- **取** $P=(2,4)$, **计算** $2P=P+P$

$$s=\frac{3\times 4+1}{2\times 4}=\frac{13}{8}=7\times 2=3 \qquad x_R=9-2\times 2=5$$

$$y_0=4-3\times 2=-2=9 \qquad y_R=-(3\times 5+9)=9$$

**所以** $2P=(5,9)$

- **再计算** $3P=P+P+P=2P+P$

$$s=\frac{9-4}{5-2}=\frac{5}{3}=4\times 5=9 \qquad x_R=81-2-5=8$$

$$y_0=4-9\times 2=-3=8 \qquad y_R=-(9\times 8+8)=-3=8$$

**所以** $3P=(8,8)$

# 目录

# 椭圆曲线密码学

- 大多数公开密钥密码系统都使用非常大的整数或多项式，计算量大，密钥和报文存储量也极大。
- 椭圆曲线密码系统可以使用较短的密钥实现同样的安全强度。
- 椭圆曲线的加法类似于模乘，累加类似于模指数。
- 需要有对应于 DLP 的难解问题。

## 椭圆曲线对数问题

- $Q = kP$，其中 $Q$，$P$ 为 $E_p(a, b)$ 上的点，整数 $k < p$；
- 给定 $k$，$P$，容易计算 $Q = kP$；
- 但是给定 $Q$，$P$，求 $k$ 很困难。

# 椭圆曲线对数问题

## 例

- 考虑椭圆曲线 $E_{23}(9,17)$，即 $y^2 = x^3 + 9x + 7 \bmod 23$。$P = (16,5)$ 和 $Q = (4,5)$ 是椭圆曲线上的点，且 $Q = kP$，求 $k$ 为多少？
- 可以通过穷举攻击方法，多次计算 $P$ 的倍数直至找到 $Q$：$P = (16,5), 2P = (20,20), 3P = (14,14), 4P = (19,20), 5P = (13,10), 6P = (7,3), 7P = (8,7), 8P = (12,17), 9P = (4,5)$
- 所以 $k = 9$。
- 实际应用中，$k$ 的值非常大，穷举攻击不可行。

# 椭圆曲线密码

## 椭圆曲线密码系统

- **域标识**：定义椭圆曲线采用的有限域椭圆曲线参数，即系数 $a$ 和 $b$。
- **基准点**（Base Point）：指定的椭圆曲线上的点 $G$。
- **阶**（Order）：$G$ 点的阶 $n$，使得 $nG = O$，记作 $order(G) = n$。

## 椭圆曲线公钥系统

- 定义在有限域上的椭圆曲线，例如 $E_p(a, b)$
- 选择基准点 $G = (x, y)$
- 选择整数 $k < order(G)$ 作为私钥
- 公钥为 $P = kG$

# 椭圆曲线加密：EC ElGamal

## EC ElGamal 加密算法

- **加密**：发送方随机选择一个正整数 $r$，加密点 $P_m$ 产生密文 $c = \{c_1, c_2\} = \{rG, P_m + rP\}$
- **解密**：接收方解密 $c_2 - kc_1 = P_m + rP - krG = P_m$

## 例 $(E_{751}(-1, 188), G = (0, 376))$

- 发送消息 $P_m = (562, 201)$，接收方的公钥 $P = (201, 5)$。
- 发送方首先随机选择 $r = 386$，并计算 $rG = (676, 558)$
- $P_m + rP = (562, 201) + 386(201, 5) = (385, 328)$
- 这样，密文即为 $C = \{rG, P_m + rP\} = \{(676, 558), (385, 328)\}$
- 接收方用私钥 $k$ 解密 $c_2 - kc_1 = P_m$

# EC Diffie-Hellman (ECDH) 密钥交换

- Alice 和 Bob 选择合适的椭圆曲线，例如 $E_p(a, b)$；
- 选择基准点 $G$，要求 $order(G)$ 是一个大整数；
- Alice 和 Bob 之间的密钥交换如下：

**ECDH 密钥交换协议**

- Alice 和 Bob 各自选择自己的私钥 $k_A, k_B < n$；
- Alice 与 Bob 分别计算公钥 $P_A = k_A G$，$P_B = k_B G$，并交换；
- 计算密钥 $K = k_A P_B = k_B P_A$。

- 因为 $K = k_A k_B G$，所以这两个密钥是一样的。
- 由于椭圆曲线对数问题是单向函数，所以 Eve 无法获知 $K$。

# 举例：ECDH 密钥交换

**例**

- $E_p(0,-4)$，即 $y^2 = x^3 - 4$，$G = (2,2)$，$p = 211$，$n = 240$；
- 计算 $240G = O$；
- $k_A = 121$，$P_A = 121(2,2) = (115,48)$；
- $k_B = 203$，$P_B = 203(2,2) = (130,203)$；
- $K = 121(130,203) = 203(115,48) = (161,69)$。

# 椭圆曲线加密：明文嵌入

- 将明文消息 $m$ 编码为点 $P_m = (x, y)$，即明文嵌入。
- 选择整数 $\kappa$，通常 $30 \le \kappa \le 50$，对消息 $m$ 计算如下一系列 $x$

$$x = m\kappa + i \quad i = 0, 1, 2, \cdots$$

直到 $\exists y$ 满足 $y^2 = x^3 + ax + b \pmod p$。

- 因为 $0 \sim p$ 的整数中有一半是模 $p$ 平方剩余，所以尝试了 $r$ 次之后，找到一个平方剩余的概率不小于 $1 - 2^{-r}$。

**例 (明文嵌入)**

- $y^2 = x^3 + 3x \pmod{4177}$，消息 $m = 2174$，取 $k = 30$
- 当 $i = 15$ 时，$x = mk + i = 65235$，$x^3 + 3x \bmod p = 38^2$
- 所以得到椭圆曲线上的点 $(65235, 38)$
- 若已知点 $(65235, 38)$，则明文 $m = \lfloor \frac{65235}{k} \rfloor = \lfloor 2174.5 \rfloor = 2174$

# 椭圆曲线密码的安全性

- 椭圆曲线密码的安全性是建立在由 $kP$ 和 $P$ 确定 $k$ 的难度之上的，即椭圆曲线对数问题。
- 椭圆曲线密码可以使用比 RSA 短得多的密钥。
- 密钥长度相同时，椭圆曲线密码与 RSA 所执行的计算量也差不多。
- 与具有同等安全性的 RSA 相比，由于椭圆曲线密码使用的密钥更短，所以椭圆曲线密码所需的计算量比 RSA 少。

| Symmetric | 56 | 80 | 112 | 128 | 192 | 256 |
|---|---|---|---|---|---|---|
| RSA n | 512 | 1024 | 2048 | 3072 | 7680 | 15360 |
| ECC p | 112 | 161 | 224 | 256 | 384 | 512 |
| Key size ratio | 5:1 | 6:1 | 9:1 | 12:1 | 20:1 | 30:1 |

# 小结

1. 基本概念
2. 安全性定义
3. RSA 密码体制
4. 椭圆曲线密码