



第2章 公开密钥密码学

南京大学计算机系黄皓教授

2011年10月10日



参考文献

1. Wenbo Mao(毛文波), 现代密码学理论与实践, 电子工业出版社, 2004年7月。
2. 周玉洁, 冯登国, 公开密钥密码算法及其快速实现, 国防工业出版社, 2002年9月。
3. D. Hankerson, A. Menezes, S. Vanstone, 椭圆曲线密码学导论, 电子工业出版社, 2005年8月第1版。
4. Bruce Schneier, 应用密码学, 机械工业出版社, 2000年1月。



内容

1. 公开密钥密码体制的基本概念
2. **RSA**密码算法
3. 离散对数的基本概念
4. **ElGamal**密码算法
5. 椭圆曲线密码算法



1. 公开密钥密码体制的基本概念



公开密码体制的提出

- 1976年W. Diffie, M. Hellman, Multiuser cryptographic techniques. In Proceedings of AFIPS 1976 NCC, pages 109-112. 提出了解密密钥和加密密钥不同、加密密钥可以公开的密码体制的可能性和思路。
- R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of ACM, 21(2):120126, 1978. 提出了利用整数分解问题的困难性设计的一个公开密码算法，称为**RSA**算法。



公开密钥算法的特点

■ 加密功能

- $\langle K_U, K_R \rangle$ 是一对密钥; $\langle pUbl\text{ic}, pR\text{ivate} \rangle$
- $E(K_U, M) = Y$
- $E(K_R, Y) = M$,
- $E(K_U, Y) \neq M$

加密者不必保密 K_U , 接受者可以将自己的公钥公布在目录服务器上。

■ 签名功能

- $E(K_R, M) = Y$,
- $E(K_U, Y) = M$,
- 没有 K_R 寻找 Y , 使得 $E(K_U, Y) = M$ 是计算不可能的。



单向陷门函数

- 对于每一个给定的 k ， $f: x \rightarrow f(k, x)$ 是一一对应函数。
- 给定 x 和 k ，计算 $y=f(k, x)$ 是容易的；
反之，给定 y 和 k ，计算 x 是困难的问题。
- 存在陷门信息 $d(k)=k'$ 及函数 $g(k', y)$ ，当 $y=f(k, x)$ 时， $x=g(d(k), y)$ 。
- 陷门信息 $d(k)$ 使得计算 $f(k, x)$ 的逆变得容易起来。
- 加密密钥 k 就不用保密了，甚至像电话本那样公布：
 - 给某人发一封密件，只要“密钥本”里面找到他的“加密密钥”，用它来加密文件就不用担心其他人窃取机密了。



2. 背包密码算法

M.E.Hellman, An Mathematics of Public-key Cryptography,
Scientific American, v241, n8, Aug. 1979, pp146-157.



背包密码算法

- 给定 $\{M_1, M_2, \dots, M_n\}$ 和 S ,
- 求 $b_i, i=1, 2, \dots, n$, 满足:

$$S = b_1 * M_1 + b_2 * M_2 + \dots + b_n * M_n$$

$$b_i = 0 \text{ 或 } 1, \quad i = 1, 2, \dots, n$$

- 例: $\{1, 5, 6, 11, 14, 20\}$
 - 明文: 1 1 1 0 0 1
 - 背包: 1 5 6 11 14 20
 - 密文: $1+5+6+20=32$



超递增背包

- { 1, 3, 6, 13, 27, 52}
 - 每一个数都比前面的数的总和大。超递增背包问题是容易问题
- 超递增背包问题是一个容易解的问题。



背包密码算法

- $\{m_1, m_2, \dots, m_k\}$: 超递增
= $\{1, 3, 6, 13, 27, 52\}$
 - N 与 m_1, m_2, \dots, m_k 互素。
 - $M > m_1 + m_2 + \dots + m_k$
 - N 与 M 互素
 - $m_1 * N \bmod M = M_1$
 - $m_2 * N \bmod M = M_2$
 -
 - $m_k * N \bmod M = M_k$
 - $\{M_1, M_2, \dots, M_k\}$: 是一般背包问题（难解的）。



背包密码算法

- 明文: $b_1 b_2 \dots b_k$
- 密文: $b_1 * M_1 + \dots + b_k * M_k$
- 解密
 - 求 N' 使的 $N * N' = 1 \pmod M$ (有解, 因为 N 与 M 互素。)
 - $(b_1 * M_1 + \dots + b_k * M_k \pmod M) * N'$
 $= b_1 * m_1 * N * N' + \dots + b_k * m_k * N * N' \pmod M$
 $= b_1 * m_1 + \dots + b_k * m_k \pmod M$
是易解的背包问题。
- 关键的陷门信息: N 。
- 密码学家 Shamir 和 Zippel 发现了变换中的缺陷, 可以从普通的背包问题中重构出超递增背包问题。 [W. Patarin, Mathematical Cryptography for Computer Scientists and Mathematicians, Totowa, N.J., Rowman & Littlefield, 1987.]



3. RSA公开密钥算法



素数

- 素数：只能被1和它本身整除的自然数；否则为合数。
- 每个合数都可以唯一地分解出素数因子
 - $6 = 2 \cdot 3$
 - $999999 = 3 \cdot 3 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 37$
 - $27641 = 131 \cdot 121$

从2 开始试验每一个小于等于 $\sqrt{27641}$ 的素数。



素因子分解的速度

- 整数 n 的十进制位数因子分解的运算次数所需计算时间（每微秒一次）

50	1.4×10^{10}	3.9小时
75	9.0×10^{12}	104天
100	2.3×10^{15}	74年
200	1.2×10^{23}	3.8×10^9 年
300	1.5×10^{29}	4.0×10^{15} 年
500	1.3×10^{39}	4.2×10^{25} 年



费马小定理

- 如果 p 是素数, a 与 p 互素, 则

$$a^{p-1} = 1 \pmod{p}$$

如果 $1 \leq i < j \leq p-1$, $i \cdot a \pmod{p} = j \cdot a \pmod{p}$ 则

存在 $1 \leq k \leq p-1$, 满足 $k \cdot a = 0 \pmod{p} \Rightarrow a \mid p$, 得出矛盾。

$\{a \pmod{p}, 2a \pmod{p}, \dots, (p-1)a \pmod{p}\}$ 是 $\{1, 2, \dots, p-1\}$ 的一个排列。

$$(a \pmod{p}) (2a \pmod{p}) \dots [(p-1)a \pmod{p}] = (p-1)! \pmod{p}$$

$$a^{p-1} (p-1)! = (p-1)! \pmod{p}$$

$$a^{p-1} = 1 \pmod{p}$$



欧拉函数

- $\varphi(n)$: 小于 n , 但与 n 互素的正整数的个数。
 - p 是素数, 则 $\varphi(p)=p-1$ 。
 - p, q 是素数, 则 $\varphi(pq)=(p-1)(q-1)$ 。



欧拉定理

欧拉定理： 如果 a 与 n 是互素的，则 $a^{\phi(n)} \equiv 1 \pmod{n}$

- 如果 a 与 n 是互素，则 $a \cdot b \equiv a \cdot c \pmod{n}$ 推出： $b \equiv c \pmod{n}$
- 设 $S = \{x_1, x_2, \dots, x_{\phi(n)}\}$ 是所有小于 n 且与 n 是互素的整数的集合。
- 因为 $a \cdot x_i \equiv a \cdot x_j \pmod{n}$ ，则有 $x_i \equiv x_j \pmod{n}$ ，
- $\{a \cdot x_1 \pmod{n}, a \cdot x_2 \pmod{n}, \dots, a \cdot x_{\phi(n)} \pmod{n}\} = S$

$$\prod_{i=1}^{\phi(n)} a \cdot x_i \pmod{n} = \prod_{i=1}^{\phi(n)} x_i \pmod{n}$$

$$a^{\phi(n)} \cdot \prod_{i=1}^{\phi(n)} x_i \pmod{n} = \prod_{i=1}^{\phi(n)} x_i \pmod{n}$$

$$a^{\phi(n)} \equiv 1 \pmod{n}$$



RSA 公开密钥密码算法

- M是明文，n是一个大数
 - 加密： $C = M^e \bmod n$
 - 解密： $M = C^d \bmod n = M^{ed} \bmod n$

- 条件：
 - 能找到e,d,n使得对所有的M，当M<n时，
$$M^{ed} = M \bmod n$$
 - 对所有的M，计算 M^e 和 C^d 的容易的。
 - 给定e、n，推导d是困难的。



RSA 公开密钥密码算法（续）

- p 、 q 是素数 秘密地选择
- $n = p q$, 公开 n
- 选择 e : e 与 $\phi(n)$ 是互素的; 公开选择
- 计算 d , 使得 $e \cdot d = 1 \pmod{\phi(n)}$; 秘密地计算

即: $e \cdot d = 1 + s \cdot \phi(n)$;

根据欧拉定理的推广

对于任意的 m , $0 < m < n$,

$$m^{ed} = m^{s \phi(n) + 1} \equiv m \pmod{n}$$



欧拉定理的推广

- 如果 p, q 是素数, $n=pq$, 则
对于任意的 $m, 0 < m < n$,

$$m^{\phi(n)} \equiv 1 \pmod{n}$$



欧拉定理的推广

若 $n=pq$ 是两个素数因子的乘积， X 限制在加密与解密所要求的集合 $\{0,1,\dots,n-1\}$ 之中，则对于任何明文 X 有 $X^{m\phi(n)+1} \equiv X \pmod{n}$

证明：

- $X=0$ 的情况显然是成立的。
- 下面要证明 $X>0$ 的情况也是成立的：
- 若 X 不是与 $n=pq$ 互素的，则 X 必须包括 p 或者 q 作为一个因子。设 p 为 X 的因子，对于某些正整数 c ，有关系式 $X=cp$ 。
- 由于 X 限于集合 $\{0,1,\dots,n-1\}$ 之中，且 $n=pq$ ，从而可知 X 必定是与 q 互素的，否则 X 将包含 q 作为一个因子，在这种情况下 X 将越出 $n-1$ 。
- 由欧拉定理有 $X^{\phi(q)} \equiv 1 \pmod{q}$ ，其中 $\phi(q)=q-1$ ，
- 但 $X^{m(p-1)\phi(q)} \equiv 1^{m(p-1)} \equiv 1 \pmod{q}$ ，对于任何整数 m 都成立，并且 $(p-1)\phi(q)=(p-1)(q-1)=\phi(n)$ ，因此有 $X^{m\phi(n)} \equiv 1 \pmod{q}$ ，或对于某些整数 t ，有 $1 = X^{m\phi(n)} + tq$
- 用 $X=cp$ 分别乘以等式的两端，得
- $$X = X^{m\phi(q)+1} + (tq)(cp)$$
- $$= X^{m\phi(q)+1} + tcn$$
- 因此
$$X^{m\phi(q)+1} \equiv X \pmod{n}$$
- 对于 q 为 X 的一个因子的情况，同理可证。



例

例:

- $n=15, p=3, q=5, \phi(n)=8$
- 生成密钥对: 令 $e=3$, 则 $d=3, (d e = 1 \bmod \phi(n))$
- 加密: 设 $M=7, C=7^e \bmod n = 7^3 \bmod 15 = 13$
- 解密: $M= C^d \bmod n = 13^3 \bmod 15 = 7$



问题

- 如何计算 $m^e \bmod n$
- 如何判定一个给定的整数是素数？
- 如何找到足够大的素数 p 和 q ？



如何计算 $m^e \bmod n$

- $e = \sum_{b_i \neq 0} 2^i$

- $m^e = m^{\sum_{b_i \neq 0} 2^i} \bmod n$

$$= \prod_{b_i \neq 0} m^{2^i} \bmod n$$

$$= \prod_{b_i \neq 0} (m^{2^i} \bmod n)$$

d:=1;

for i=k downto 0

 d:=d*d mod n

 if $b_i = 1$ then

 d:=d*a mod n

return d



大素数生成—素数的分布

- 存在无穷多个素数
 - 假设已知有 k 个素数 p_1, p_2, \dots, p_k ;
 - 考虑 $n = p_1 \cdot p_2 \cdot \dots \cdot p_k + 1$
 - 要么存在 $p \mid n$, 这时 $p \neq p_1, \dots, p \neq p_k$
 - 要么 n 也是素数



大素数生成—素数的分布

- 设 $x > 0$ ，记 $\pi(x)$ 为不大于 x 的素数的个数，则：

$$\lim_{x \rightarrow \infty} \frac{\pi(x) \ln x}{x} = 1$$

$$\pi(x) \approx \frac{x}{\ln x}$$

长度为 t 位的数共有 $2^t - 2^{t-1}$ 个，所以一个长度为 t 位的随机数为素数的概率为

$$\left(\frac{2^t}{\ln 2^t} - \frac{2^{t-1}}{\ln 2^{t-1}} \right) / (2^t - 2^{t-1}) = \frac{t-2}{(t-1)t \ln 2}$$

例如：一个长度为256位的随机数为素数的概率为

$$\frac{254}{255 \times 256 \times \ln 2} \approx 0.011$$



基于费马小定理的素性检测

- I. 随机选取一个奇数 n
- II. 随机选取一个整数 a , $a < n$
- III. $a^{n-1} \neq 1 \pmod n$, 则 n 不是素数, 转I
- IV. 如果多次通过检测就接受 n 是素数, 否则转II。



其它素性检测算法

- Solovay-Strassen算法
 - Miller-Rabin算法
 - Mesenne算法
 - 基于椭圆曲线的素性检测
-
- 裴定一，祝跃飞，算法数论，科学出版社，2002年9月第一版。
 - 周玉洁，冯登国，公开密钥密码算法及其快速实现，国防工业出版社，2002年9月



大素数生成—Solovay-Strassen素性检测

设 $n > 2$ 是一个奇数

(1) 随机均匀地选取 $a \in \{1, 2, \dots, n-1\}$

(2) 计算 $\gcd(a, n)$

(3) 如果 $\gcd(a, n) \neq 1$, 则 n 不是素数

(4) 计算 $\left(\frac{a}{n}\right)$ 和 $a^{(n-1)/2} \bmod n$ 其中 $\left(\frac{a}{n}\right)$ 是 *Jacobi* 符号

(5) 如果 $\left(\frac{a}{n}\right) = a^{(n-1)/2} \bmod n$, 则 n 可能是素数, 否则 n 不是素数。

- 随机生成一个奇数 $n > 2$
- 随机均匀地选取序列 $a_1, a_2, \dots, a_k \in \{1, 2, \dots, n-1\}$, 对 n 进行 k 次素性测试,
- 如果每次输出都是 “ n 可能是素数”, 则 n 是合数的概率小于 $1/2^k$ 。



RSA 参数的选择

■ 模数 n 的选择

□ p, q 的差 $|p-q|$ 必须很大。

■ $n=p \cdot q = [(p+q)/2]^2 - [(p-q)/2]^2$

■ 如果 $p-q$ 很小, 则 $n \approx [(p+q)/2]^2$

■ 逐个检查大于 $n^{1/2}$ 的 t , 看是否有 t^2-n 是一个平方数 s^2 。如果存在则 $n=t^2-s^2=(t+s)(t-s)$

□ $p-1$ 与 $q-1$ 的最大公因子应很小

□ p, q 必须为强素数

■ 赖溪松, 韩亮, 张真诚, 计算机密码学及其应用, 国防工业出版社, 2001年7月。



对RSA的攻击方法

1. 强力攻击（穷举法）：尝试所有可能的私有密钥
2. 因子分解方法
3. 时间性攻击：取决于解密算法的运算时间



Pollard p-1 算法（分解因子算法）

```
a=2;
for j=2 to B do a = aj;
d = gcd( a-1, n)
if d>1 and d<n then
    return d
else
    return false;
```

- Douglas R. Stinson, 密码学原理与实践, 电子工业出版社, 2003年2月。
- 周玉洁, 冯登国, 公开密钥密码算法及其快速实现, 国防工业出版社, 2002年9月。



其它因子分解算法

- 二次筛法(quadric sieve)

$$O(e^{(1+o(1))\sqrt{\ln n \ln \ln n}})$$

- 椭圆曲线分解算法(elliptic curve factoring)

$$O(e^{(1+o(1))\sqrt{2 \ln n \ln \ln n}})$$

- 数域筛法(number field sieve)

$$O(e^{(1.92+o(1))(\ln n)^{1/3} (\ln \ln n)^{2/3}})$$



因子分解的一些事例

- 1985年用二次筛法分解了69位的十进制数 $2^{251}-1$.
- 1989年Lenstra、Manasse利用二次筛法再大量的工作站上并行运算，分解了一个106位的十进制数
- 1994年Atkins等利用二次筛法分解了一个RSA-129的十进制数。
- RSA-130在1996年被利用数域筛法分解。
- RSA-140在1999年2月被利用数域筛法分解。
- RSA-155在1999年8月被利用数域筛法分解。
- 155位的十进制数大约十512位，因此512位的RSA不应该被认为是安全的。
- 推测：768位的模式将在2010年被分解，1024位的模数将在2018年被分解。



其它对RSA的攻击

- 选择密文攻击
 - 对RSA公共模数的攻击
 - 对RSA低加密指数的攻击
 - 对RSA低解密密指数的攻击
 - 对RSA的加密和签名的攻击
-
- 周玉洁，冯登国，公开密钥密码算法及其快速实现，国防工业出版社，2002年9月。



选择密文攻击

情况：Eve 在Alice的通信过程中进行窃听，设法成功选取了一个用Alice公钥加密的密文 c 。Eve想读出消息。

- e : 加密密钥, d : 解密密钥。
- $c = m^e \bmod n$, $m = c^d \bmod n$
- 已知 c , 没有 d , 求 m , 即求 $m = c^d \bmod n$
- 随机选取一个数 r , $r < n$ 。
- $x = r^e \bmod n$ $r = x^d \bmod n$

- $y = x^c \bmod n$ 选择的密文
 - $t * r = 1 \bmod n$
 - $u = y^d \bmod n$: 请Alice对 y 签名
 - $t u \bmod n$ 解密服务
- $$\begin{aligned} &= t y^d \bmod n \\ &= t x^d c^d \bmod n \\ &= c^d \bmod n \\ &= m \end{aligned}$$

不能天真地对消息签名。



公共模数攻击

- m 是明文消息，两个加密密钥是 e_1 、 e_2 ， n 是公共的模数，两个密文是
 - $c_1 = m^{e_1} \bmod n$
 - $c_2 = m^{e_2} \bmod n$
- 密码分析者知道 n 、 e_1 、 e_2 、 c_1 、 c_2
- 如果 e_1 、 e_2 互素(一般情况下会如此)，则存在 r 、 s 满足 $r \cdot e_1 + s \cdot e_2 = 1$ 。
- 不妨假定 $r < 0$ ， $c_1 \cdot c_1' = 1 \bmod n$ ，则
 - $(c_1')^r \cdot c_2^s \bmod n$
 - $= m^{r \cdot e_1 + s \cdot e_2} \bmod n$
 - $= m \bmod n$
- 不要在一组用户之间共享 n 。



RSA 算法的原理

欧拉定理的推广：

若 $n=pq$ 是两个素数因子的乘积，

M 限制在加密与解密所要求的集合 $\{0,1,\dots,n-1\}$ 之中，
则对于任何明文 M 和 c ，满足 $M^{c\phi(n)} \equiv 1 \pmod{n}$

互素定理：

如果 e 与 $\phi(n)$ 互素，则 c,d ，使得 $e d + c \phi(n) = 1$ ，即

$$e d = 1 - c \phi(n)$$

$$e d = 1 \pmod{\phi(n)}$$

RSA 算法：

加密： $C = M^e \pmod{n}$

解密： $C^d = (M^e)^d \pmod{n} = M^{ed} \pmod{n}$
 $= M M^{c \phi(n)} \pmod{n} = M \pmod{n} = M$



4. 离散对数的基本概念



离散对数

■ 离散对数问题

- 欧拉定理： 如果 a 与 n 是互素的则， $a^{\phi(n)} \equiv 1 \pmod{n}$
- 如果 a 与 n 是互素的则必存在 $1 \leq m \leq n-1$ 使得 $a^m \equiv 1 \pmod{n}$
- 满足式子 $a^m \equiv 1 \pmod{n}$ 的最小的整数 m (可能 $< \phi(n)$) 称为 $a \pmod{n}$ 的阶
- 例如
 - $7^1 \equiv 7 \pmod{19}$
 - $7^2 \equiv 11 \pmod{19}$
 - $7^3 \equiv 1 \pmod{19}$

 - **$7 \pmod{19}$ 阶为3 ($< \phi(19)=18$)**



素根

- 如果 $a \pmod{n}$ 的阶为 $n-1$, 则
 - $a^1 \pmod{n}$, $a^2 \pmod{n}$, ..., $a^{n-1} \pmod{n}$ 互不相同, 是 $1, 2, n-1$ 的一个排列。
 - 任意给定 $1 \leq x \leq n-1$, 存在 y , 使得 $a^y = x \pmod{n}$
 - 称 a 为 (\pmod{n}) 的素根
- 对于任意整数 b 和 n 的素根 a , 必存在 i , $1 \leq i \leq n-1$, $a^i = b \pmod{n}$ 指数 i 称为 b 以 $a \pmod{n}$ 为底数的离散对数。
- 求离散对数的问题是一个非常困难的问题。



循环群的基本概念

- G 是一个群， $a \in G$ ， e 是 G 的单位元，则满足 $a^i=e$ 的最小正整数 i 成为元素 a 的阶。如果这样的整数不存在，则称 a 为无限阶元素。
- 如果存在一个元素 $a \in G$ ，对与任意一个元素 b ，都存在一个整数 i ，使得 $b=a^i$ ，则 G 称为循环群，元素 a 称为 G 的一个生成元； G 也称为由 a 生成的群，记为 $G=\langle a \rangle$ ， a 称为群 G 的单位元的本原根。
- $Z_n = \{0, 1, \dots, n-1\}$ 按照整数模 n 加法构成加法群，
 Z_n 中与 n 互素的非0整数全体记为 Z_n^* ，
 Z_n 中与 n 互素的元素构成的子集按照模 n 乘法构成群， $|Z_n^*| = \phi(n)$
- p 为素数时，乘法群 Z_p^* 记为 F_p ， $|F_p| = p-1$ ， F_p 一定存在本原根（有限域的乘法群一定是循环群）。所以 F_p 一定循环群。按照模加法和模乘法构成一个域
- 假设 a 是 F_p 的本原根，则 F_p 上的任何元素 h ，存在 i ，使得 $h=a^i$ 。即：给定 a 、 h ，计算 i ， $i = \log_a h$ ，称为 F_p 上的离散对数问题 $\log_a h$ 。



循环群上的离散对数

- 离散对数问题：
 - 假设 $\alpha \in G$ 是循环群 G 的一个生成元，元素 $\beta \in \langle \alpha \rangle$
 - 找一个唯一的整数 s 使得 $\beta = \alpha^s$
- 离散对数的蛮力算法
 - 给定循环群 G 、生成元 α 、任意元素 β ;
 - 分别对 $i=1, \dots, |G|$, 检测 $\alpha^i = \beta$ 是否成立;
- 蛮力法最多需要 $|G|$ 次群运算。



离散对数算法— Shanks算法

- 设循环群G的阶为n。
- m是正整数，满足 $n > m > 1$ 。
- 对于任意的t, $0 \leq t < n$, $h = \alpha^t$, 我们有
- $t = qm + r$, $0 \leq q \leq [n/m]$, $0 \leq r < m$
- $h = \alpha^t = \alpha^{qm+r} = \alpha^{qm} \alpha^r = (\alpha^m)^q \alpha^r$
- $h \cdot \alpha^{-r} = (\alpha^m)^q$

- 计算
 $L = \{ (\alpha^m)^q, q=0, 1, \dots, [n/m] \}$
- 然后计算
 $h \cdot \alpha^{-r}, r=0, 1, \dots, m-1$
 - 找出哪一个出现在L表中,
 - 记录相应的q, r,
- $t = q \cdot m + r$ 就是需要的离散对数。
- 算法复杂度: $O([n/m] + m)$

Shanks算法是第一个达到求解离散对数一般算法的复杂度下界的确定性一般算法。



Diffie-Hellman问题

- 给定一个素数 p , $\text{mod } p$ 的一个素根 a , 已知 $a^x \pmod{p}$, $a^y \pmod{p}$ 求 $a^{xy} \pmod{p}$ 。
 - 求离散对数问题: 由 p , a^x , a 求 x 。
 - 计算 $a^y \pmod{p}^x = a^{xy} \pmod{p}$ 。
- Diffie-Hellman密钥交换
- Diffie-Hellman问题的难度不超过离散对数的难度。



Diffie-Hellman 密钥交换

- Alice 选取一个大的随机数 x ，发送给 Bob

$$X = \alpha^x \bmod n$$

- Bob 选取一个大的随机数 y ，发送给 Alice

$$Y = \alpha^y \bmod n$$

- Alice 计算 $s = Y^x \bmod n$

- Bob 计算 $t = X^y \bmod n$

- $s = t = \alpha^{xy} \bmod n$



5. ElGamal密码算法



ElGamal 加密算法

■ 生成密钥对

- 生成一个大的随机素数 p 和整数 $\text{mod } p$ 的乘法群 Z_p^* 的生成元 α ;
- 选取一个随机整数 s ($1 \leq s \leq p-2$), 计算 $\beta = \alpha^s \pmod{p}$;
- 公钥= (p, α, β) , 私钥= s 。

■ 加密明文信息 m

- 选取一个随机整数 k ($1 \leq k \leq p-2$),
- 计算 $X = \alpha^k \pmod{p}$, $Y = m (\beta)^k \pmod{p}$
- (X, Y) 就是密文

■ 解密

- 有私钥 s , 密文 (X, Y) 。注意: $Y = m \cdot \alpha^{sk} \pmod{p}$
- 计算 X^{p-1-s} , $X^{p-1-s} \pmod{p} = X^{-s} \pmod{p} = \alpha^{-sk} \pmod{p}$
- 计算 $Y \cdot X^{-s} \pmod{p}$ 恢复 m ,
 $Y \cdot X^{-s} \pmod{p} = m (\alpha^s)^k \cdot \alpha^{-sk} \pmod{p} = m$

- 加密者用 $(\alpha^s)^k$ 将明文隐藏起来, 因为解密者知道 s , 所以可以从密文的一部分 $X = \alpha^k$ 恢复 $(\alpha^s)^k$, 因而可以解密。



ElGamal 加密算法

- 不能用 β 作隐藏因子，因为 β 的逆是可以通过辗转相除法计算：因为 β 与 p 是互素的，所以存在 a, b ，使得： $a \cdot \beta + b \cdot p = 1$ 成立。
- 因为用户不知道 k ，所以无法计算 $(\beta)^k$ 的逆，而因为接收者需要知道 k 的信息，所以用 $X = \alpha^k$ 传递过去，因此密文变成了 (X, Y) 。



ElGamal 加密算法

- 低熵明文空间
 - a) 给定密文 c
 - b) 猜测明文，用已知的公钥加密明文，与 c 比较；
 - c) 如果相等，则猜到了明文，否则转a)。
- 1982年美国加州大学伯克利分校的S. Golwasser, S. Micali提出了一种**概率加密**方法，同一个明文加密后可以得到不同的密文。
- ElGamal算法具有这样的性质，密文除了与公钥相关外，还与加密时选择的随机数 k 有关，猜测的攻击者，不知道 k ，即使猜中了明文，加密后还不能发现。



ElGamal 签名算法

- 生成密钥对
 - 生成一个大的随机素数 p 和整数 $\text{mod } p$ 的乘法群 Z_p^* 的生成元 α ;
 - 选取一个随机整数 s ($1 \leq s \leq p-2$), 计算 $\beta = \alpha^s \pmod{p}$;
 - 公钥 (p, α, β) , 私钥 s 。
- 对信息 m 签名
 - 选取一个随机整数 k ($1 \leq k \leq p-2$), 计算 $X = \alpha^k \pmod{p}$
 - 从方程 $m = (s \cdot X + k \cdot Y) \pmod{p-1}$ 中求解 Y ;
 - 签名为 (X, Y) ;
- 验证签名
 - 验证等式: $\beta^X \cdot X^Y = \alpha^m \pmod{p}$
 - $(\alpha^s)^X \cdot (\alpha^k)^Y = \alpha^{s \cdot X + k \cdot Y} \pmod{p}$
 $= \alpha^{m + t \cdot (p-1)} \pmod{p}$
 $= \alpha^m \cdot \alpha^{t \cdot (p-1)} \pmod{p}$
 $= \alpha^m \pmod{p}$

■ 签名的算法不是用私钥加密明文。



6. 椭圆曲线密码算法



椭圆曲线密码体制

- 1985年, N. Koblitz和V. Miller分别独立提出了椭圆曲线密码体制(ECC), 其依据就是定义在椭圆曲线点群上的离散对数问题的难解性。



椭圆曲线

- 公元250年，古希腊出版了Diophantus的巨作：《Arithmetic》，共13卷，现保存的只有6卷。
- 书中主要考虑的问题是：有理多项式方程是否有有理解。

$f(x,y) \in \mathbb{Q}[x, y]$, 存在 $x, y \in \mathbb{Q}$, 使得 $f(x, y) = 0$



椭圆曲线

- Weierstrass方程:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad a_i \in K$$

光滑的Weierstrass曲线加上一个无穷远点 θ ，称为 K 上的椭圆曲线。

- 同样可以定义 R 、 C 、 F_q 上的椭圆曲线，如

$$E(K) = \{ (x, y) \in K^2 \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \} \cup \{\theta\}$$



有关的基本概念

1. 无穷远元素（无穷远点，无穷远直线）

- a) 平面上任意两相异直线的位置关系有相交和平行两种。引入无穷远点，是两种不同关系统一。
- b) 平面上一组相互平行的直线，有公共的无穷远点。
- c) 平面上任何相交的两直线 L_1, L_2 有不同的无穷远点。
- d) 全体无穷远点构成一条无穷远直线。

2. 射影几何学-齐次坐标

- a) 设在平面上已经建立了以 O 为原点的直角坐标系， (x, y) 为一点 p 的坐标。令 $x=X/Z, y=Y/Z (Z \neq 0)$ ，则比值 $X:Y:Z$ 完全确定 p 的位置， (X, Y, Z) 就叫做 p 的齐次坐标，当 $Z=1$ ，则点 (x, y) 的齐次坐标为 $(x, y, 0)$ 。
- b) 原点的齐次坐标显然可以写成 $(0, 0, 1)$ 。
- c) 在 $x=X/Z, y=Y/Z (Z \neq 0)$ 中，令 $Z \rightarrow 0$ ，则 $x \rightarrow \infty, y \rightarrow \infty$ 。因此可以用 $(x, y, 0)$ 表示无穷远点。



变量的相容性变换

- 若椭圆曲线的基础域**K**的特征不等于**2**或**3**，作相容性变换

$$(x, y) \rightarrow \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216} - \frac{a_1^3 + 4a_1a_2 - 12a_3}{24} \right)$$

把E变为曲线 $y^2 = x^3 + ax + b$ 。判别式 $\Delta = 16(4a^3 + 27b^2)b$ 。

- 若椭圆曲线的基础域**K**的特征等于**2**

- $a_1 \neq 0$, 作相容性变换

$$(x, y) \rightarrow \left(a_1^2 x + \frac{a_3}{a_1}, a_1^3 y + \frac{a_1^2 a_4 + a_3^2}{a_1^3} \right)$$

把E变为曲线 $y^2 + xy = x^3 + ax^2 + b$ 。这样的曲线称为非超奇异的，判别式 $\Delta = b$ 。

- $a_1 = 0$, 作相容性变换

$$(x, y) \rightarrow (x + a_2, y)$$

把E变为曲线 $y^2 + cy = x^3 + ax + b$ 。这样的曲线称为超奇异的，判别式 $\Delta = c^4$ 。



变量的相容性变换

- 若椭圆曲线的基础域**K**的特征等于**3**,
- $a_1^2 \neq a_2$, 作相容性变换

$$(x, y) \rightarrow \left(x + \frac{d_4}{d_2}, y + a_1 x + a_1 \frac{d_4}{d_2} + a_3\right)$$

$$\text{其中 } d_2 = a_1^2 + a_2, \quad d_4 = a_4 - a_1 a_3$$

把E变为曲线 $y^2 = x^3 + ax^2 + b$, 这样的曲线称为非超奇异的, 判别式 $\Delta = -a^3b$ 。

□ $a_1^2 = a_2$, 作相容性变换

$$(x, y) \rightarrow (x, y + a_1 x + a_3)$$

把E变为曲线 $y^2 = x^3 + ax + b$, 这样的曲线称为超奇异的, 判别式 $\Delta = -a^3$ 。

D. Hankerson, A. Menezes, S. Vanstone, 椭圆曲线密码学导论, 电子工业出版社, 2005年8月。



- 为什么把这样的曲线称为椭圆曲线？

椭圆曲线方程为：

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$



椭圆曲线与椭圆积分

- 形如 $\int R(x,y)dx$ 的不定积分，其中 y 是 x 的代数函数，即满足代数方程 $P(x,y)=0$ ， P 是关于 x 和 y 的整多项式。这类积分成为阿贝尔（Abel）积分。

为了纪念挪威天才数学家阿贝尔（N. H. Abel, 1802—1829）诞辰200周年，挪威政府于2003年设立了一项数学奖——阿贝尔奖。这项每年颁发一次的奖项的奖金高达80万美元，相当于诺贝尔奖的奖金，是世界上奖金最高的数学奖。

- 不定积分 $\int R(x, \sqrt{ax^2 + bx + c})dx$ 可以通过变量替换，变成有理函数的积分从而可以得到有限形式表达的原函数。
- 然而以下两类积分在一般情况下不能用初等函数表示原函数：

$$\int R(x, \sqrt{ax^3 + bx^2 + cx + d})dx$$

$$\int R(x, \sqrt{ax^4 + bx^3 + cx^2 + dx + e})dx$$



椭圆曲线与椭圆积分

$$\int R(x, \sqrt{ax^3 + bx^2 + cx + d}) dx$$

$$\int R(x, \sqrt{ax^4 + bx^3 + cx^2 + dx + e}) dx$$

- 上述的积分包含了4-5个参数，制作5个参数的积分表很困难，因此人们希望把这些积分化成几种参数尽可能少的形式。

$$\int \frac{dz}{\sqrt{(1-z^2)(1-k^2z^2)}}$$

$$\int \frac{z^2 dz}{\sqrt{(1-z^2)(1-k^2z^2)}}$$

$$\int \frac{dz}{(1+hz^2)\sqrt{(1-z^2)(1-k^2z^2)}}$$

$$(0 < k < 1)$$

$$\int \frac{d\varphi}{\sqrt{1-k^2\sin^2\varphi}}$$

$$\int \sqrt{1-k^2\sin^2\varphi} d\varphi$$

$$\int \frac{d\varphi}{(1+h\sin^2\varphi)\sqrt{1-k^2\sin^2\varphi}}$$

- 上述的积分分别成为第一、二、三类椭圆积分。



椭圆曲线与椭圆积分

椭圆 $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ 的曲线长度可以表示成 积分:

$$\int_0^{2\pi} a\sqrt{1-\varepsilon^2\sin^2 t} dt \quad (*)$$

其中 $\varepsilon = \frac{\sqrt{a^2 - b^2}}{a}$,

椭圆 $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ 可以表示成参数方程 :

$$x = \varphi(t) = a \cdot \sin t, \quad y = \psi(t) = b \cos t$$

曲线的长度可以用定积分 $\int_{\alpha}^{\beta} \sqrt{\varphi'^2(t) + \psi'^2(t)} dt$ 计算.

- 上述的椭圆曲线的长度公式与前述的椭圆积分的标准形式相同, 因此把椭圆积分中被积函数对应的有理函数方程表示的曲线称为椭圆曲线。
- Г. М. Фихтенгольц 菲赫金哥尔兹, 微积分学教程(二卷一分册), 人民教育出版社, 1978年3月。第8章第5节。



实域R上椭圆曲线

- 考虑椭圆曲线:

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

其中 $a, b \in \mathbb{R}$, $4a^2 + 27b^3 \neq 0$ 。

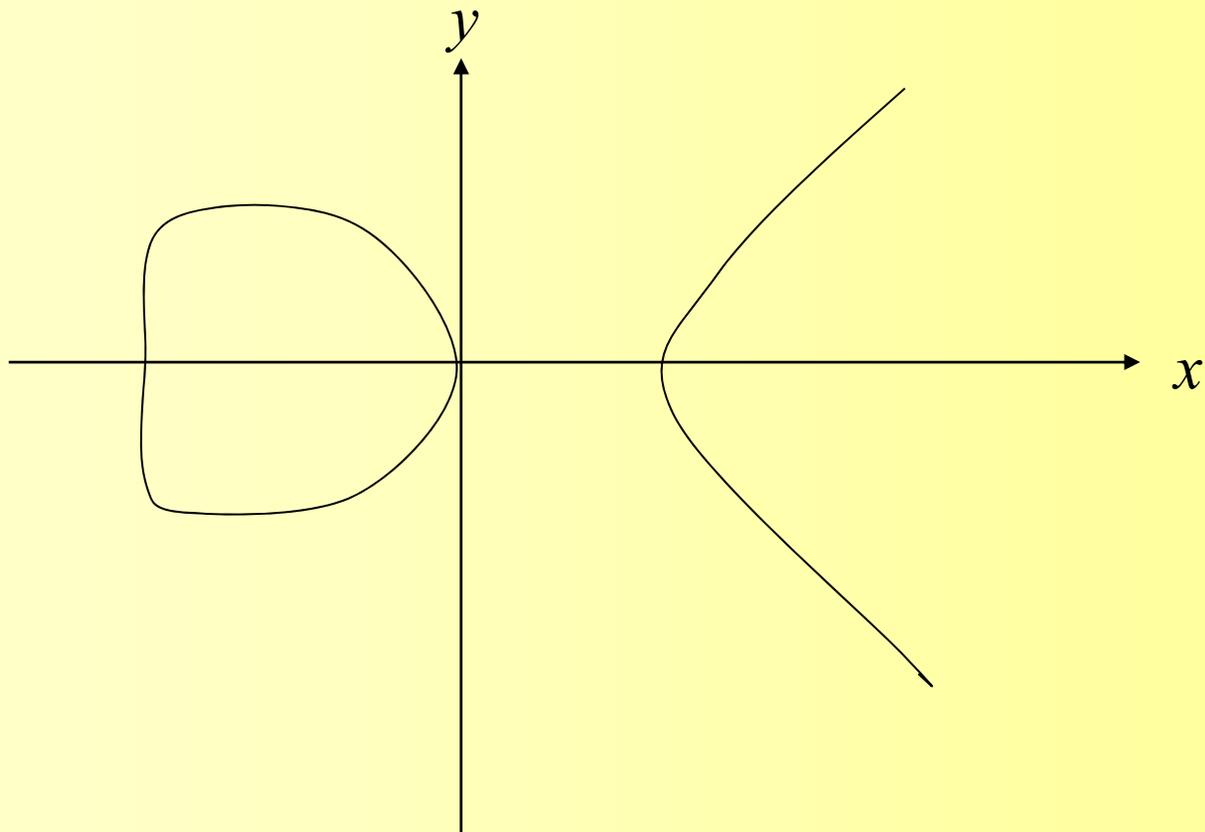
- 设 $f(x) = x^3 + ax + b$,
方程 $f(x) = 0$ 的判别式为: $\Delta = 4a^2 + 27b^3$
- 当判别式 $\Delta = 0$ 时, 方程 $f(x) = 0$ 有一个二重零点 x 。显然 $(x, 0)$ 在 E 上。
对于 $E(x, y) = y^2 - x^3 + ax + b = 0$, 有:

$$\frac{\partial E}{\partial y} = 2y \Big|_{(x,0)} = \frac{\partial E}{\partial x} \Big|_{(x,0)} = 0$$

曲线在这点是个奇点, 没有切线。 $(x-\alpha)^2(x-\beta)$ 在 α 点的导数为 0。



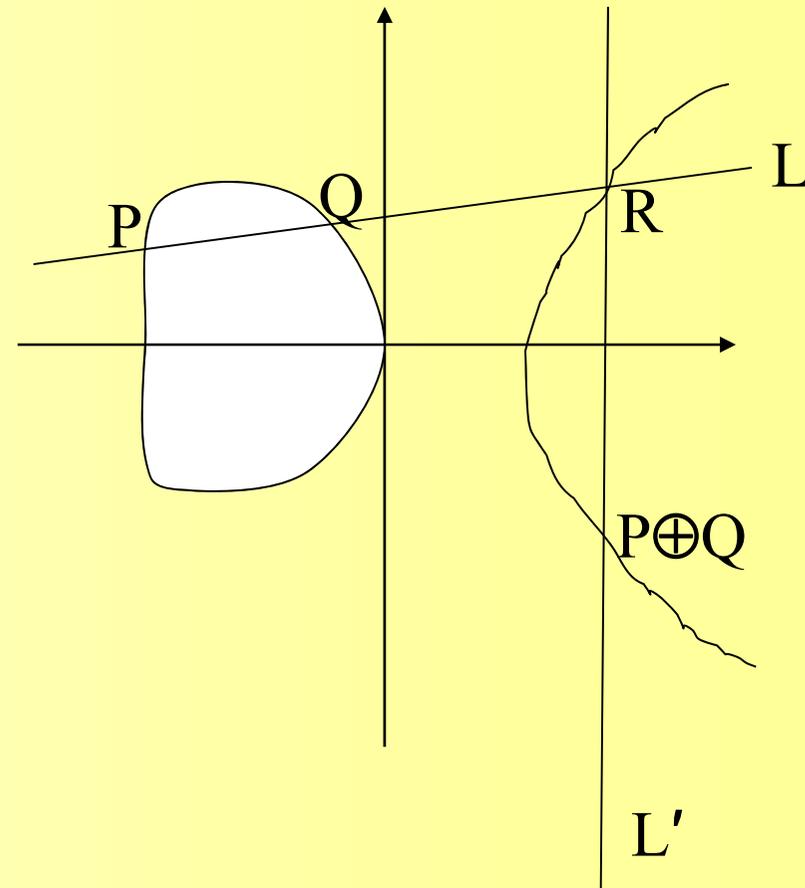
实域 \mathbf{R} 上椭圆曲线

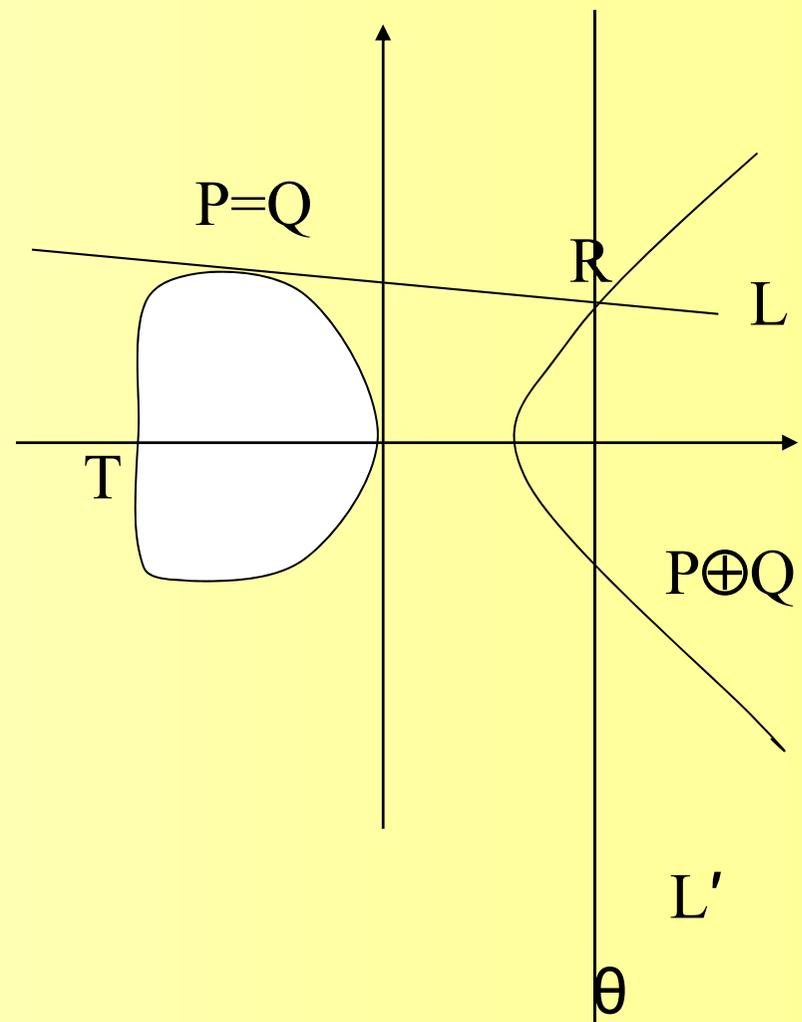
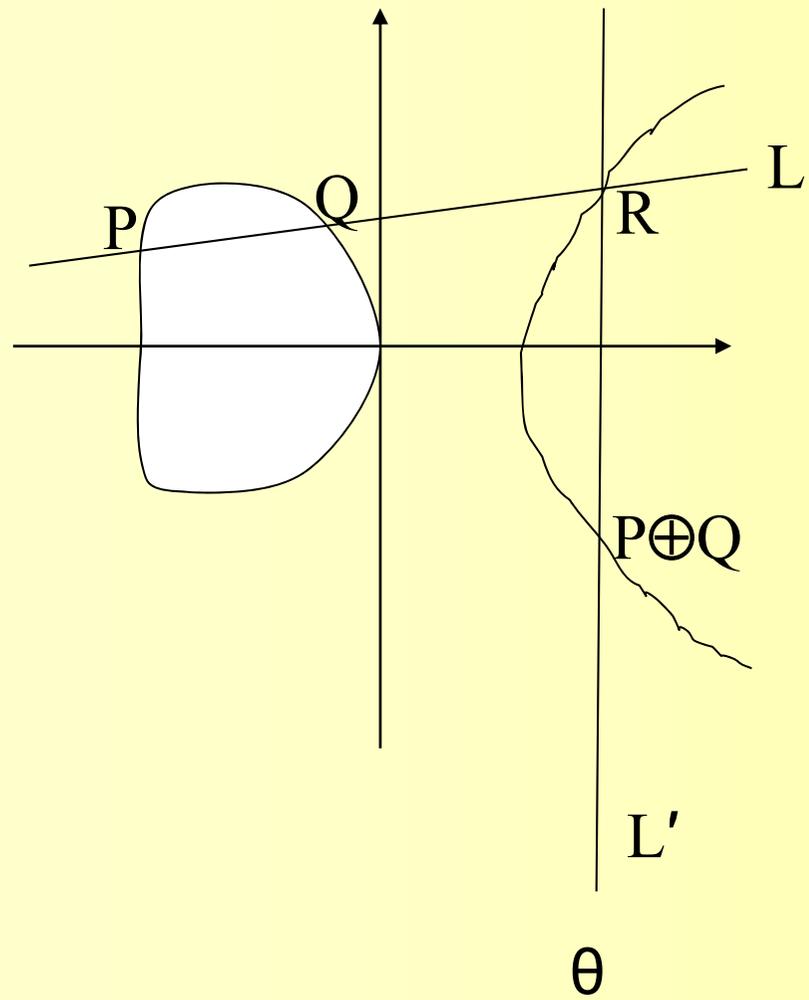




实域 \mathbf{R} 上椭圆曲线

- 实域 \mathbf{R} 上椭圆曲线的点的加法运算法则
 - 设 L 为一条直线。
 - 因为 E 的方程是三次的，所以 L 可与 E 恰有三个交点，记为 P, Q, R 。
(如果 L 与 E 相切，那么 P, Q, R 可以不是相异的)
 - 按下述方式定义 E 上运算 \oplus
 - 设 $P, Q \in E$ ， L 为联接 P, Q 的直线
(若 $P=Q$ ，则 L 取过 P 点的切线)；
 - 设 R 为 L 与 E 的另一个交点；
 - 再取连接 R 与无穷远点的直线 L' ，
则 L' 与 E 的另一个交点定义为 $P \oplus Q$ 。







椭圆曲线点集E上的 \oplus 法运算

- 设 $P(x_1, y_1), Q(x_2, y_2) \in E$, $R(x_3, y_3) = P \oplus Q$, 则

$$\lambda = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}, & \text{当 } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1}, & \text{当 } x_1 = x_2 \text{ 且 } y_1 = y_2 \neq 0 \end{cases}$$

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$



- Poincaré 定理: $E(F_q, \oplus)$ 构成交换群。
- Hasse定理: $|E(F_q)| = 1+q-t$, 其中 $|t| \leq 2 \cdot q^{1/2}$



有限域上椭圆曲线

- F_p (素域, p 为素数)上椭圆曲线

令 $p > 3$, $a, b \in F_p$, 满足 $4a^3 + 27b^2 \neq 0$, 由参数 a 和 b 定义的 F_p 上的一个椭圆曲线方程为:

$$y^2 = x^3 + a \cdot x + b$$

它的所有解 (x, y) , ($x \in F_p, y \in F_p$), 连同称为“无穷远点” (记为 θ) 的元素组成的集合记为 $E(F_p)$ 称为 F_p 上椭圆曲线。



$E(F_p)$ 的加法规则

- I. $\theta \oplus \theta = \theta$ (单位元素)
- II. $(x, y) \oplus \theta = (x, y)$, 任给 $(x, y) \in E(F_p)$
- III. $(x, y) \oplus (x, -y) = \theta$, 任给 $(x, y) \in E(F_p)$, 即点 (x, y) 的逆元为 $(x, -y)$.
- IV. 令 $(x_1, y_1), (x_2, y_2)$ 为 $E(F_p)$ 中非互逆元, 则

$$(x_1, y_1) \oplus (x_2, y_2) = (x_3, y_3), \text{ 其中}$$

$$\alpha = (y_2 - y_1) / (x_2 - x_1)$$

$$x_3 = \alpha^2 - 2x_1, \quad y_3 = \alpha(x_1 - x_3) - y_1$$

- V. (倍点运算规则) 设 $(x_1, y_1) \in E(F_p), y_1 \neq 0$, 则

$$2(x_1, y_1) = (x_3, y_3), \text{ 其中}$$

$$x_3 = \alpha^2 - 2x_1, \quad y_3 = \alpha(x_1 - x_3) - y_1 \quad \alpha = (3x_1^2 + a) / (2y_1)$$



椭圆曲线密码体制 — 密钥的生成

- I. 选取一个基域 F_q ，一个定义在 F_q 上的椭圆曲线 $E(F_q)$ ，和 E 上一个为素数阶 n 的点 P 。
- II. 在区间 $[1, n-1]$ 中随机选取一个整数 d 。
- III. 计算点 $Q := d \cdot P$ 。（ d 个 P 相 \oplus ）
- IV. 公开密钥 —— $(E(F_q), P, n, Q)$ 。
- V. 私钥为整数 d 。



椭圆曲线密码体制 — 加密与解密ECES

加密明文m

- I. 查找公钥($E(F_q), P, n, Q$),
- II. 将m表示成一个域元素 $m \in F_q$,
- III. 在区间 $[1, n-1]$ 内选取一个随机数k,
- IV. 依据公钥计算点 $(x_1, y_1) := k \cdot P$
(k个P相 \oplus)
- V. 计算点 $(x_2, y_2) := k \cdot Q$,如果 $x_2=0$,
则回到第iii)步
- VI. 计算 $C := m \cdot x_2$
- VII. 密文 = (x_1, y_1, C)

隐藏因子的信息

隐藏因子

解密密文 (x_1, y_1, C)

通过隐藏因子的信息计算隐藏因子

- I. 使用私钥d, 计算

$$d \cdot (x_1, y_1) = d \cdot k \cdot P$$

$$= k \cdot d \cdot P$$

$$= k \cdot Q$$

$$= (x_2, y_2)$$
- II. 计算 F_q 中 x_2^{-1}
- III. 计算 $m = C \cdot x_2^{-1}$, 恢复出明文数据m。

可以与ElGamel算法的步骤建立对应关系



椭圆曲线密码体制 — 签名与验证ECSA

对信息m签名

- I. 将m表示成一个二进制串
- II. 计算hash值 $e = H(m)$;
- III. 在区间 $[1, n-1]$ 内选取一个随机数k,
- IV. 计算点 $(x_1, y_1) := k \cdot P$ (k个P相 \oplus)
- V. 计算 $r = (x_1 + e) \bmod q$;
- VI. 利用私钥d计算
 $s = (k - d \cdot r) \bmod n$
- VII. 签名 (r, s) 。

验证签名(r, s)

- I. 查找公钥 $(E(F_q), P, n, Q)$,
- II. 计算点 $(x_1, y_1) = sP + rQ$
- III. 计算hash值 $e = H(m)$;
- IV. 计算 $R = (x_1 + e) \bmod q$;
- V. 如果 $R = r$, 则接受签名。

$$\begin{aligned} sP + rQ &= (k - dr)P + rdP \\ &= kP - drP + rdP \\ &= kP \\ &= (x_1, y_1) \end{aligned}$$



椭圆曲线离散对数问题 ECDLP

- 椭圆曲线上的离散对数问题：已知 P , $[m]P$, 求 m ,
- 椭圆曲线公钥算法的安全性基于椭圆曲线离散对数的难解性
- Hasse定理： $|E(F_q)| = 1+q-t$, 其中 $|t| \leq 2 \cdot q^{1/2}$
- (椭圆曲线可能的阶) 对于任意的素数 p 、满足 $|t| \leq 2p^{1/2}$ 的整数 t , 存在阶为 $p+1-t$ 的域 F_p 上的椭圆曲线 E 。
- F_p 上的椭圆曲线 E 的阶与 p 大体相当。



椭圆曲线离散对数问题 ECDLP

- 求解ECDLP的最好的算法的时间复杂度 $O(q^{1/2})$
- 在有限域中的离散对数算法的时间复杂度

$$\text{sub_exp}(q) = \exp(c \cdot (\log q)^{1/3} (\log \log q)^{2/3})$$

- 显然， $O(q^{1/2})$ 要比 $\text{sub_exp}(q)$ 增长快得多。
- 在ECDLP中，令 $q \approx 2^{160}$ ，抗强力搜索的难度是 2^{80} 。
- 为使有限域上离散对数问题获得的相似的难度， $\text{sub_exp}(q)$ 就需要 q 达到 2^{1000} 的数量级。
- 椭圆曲线上的离散对数问题比有限域上的离散对数问题更难处理。
- 可以在椭圆曲线公钥密码中以较小的密钥长度达到在有限域中更大的密钥长度同样的安全性。



实践题

- 实现RSA 算法
- 实现ElGamel算法
- 实现一个椭圆曲线算法

要求详细阐述算法中的数学运算实现原理。

例如在**RSA**算法中详细阐述幂模运算的方法，比较不同方法的性能。

- 讨论公开密钥算法的安全性
- <http://cs.nju.edu.cn/huanghao/security>