第6章 公钥密码学和离散对数

杨礼珍

同济大学计算机科学与技术系, 2017

Outline

第6章作业

E6.1：对$\mathbb{Z}_p^*$上的ElGamal公钥加密体制做如下变形：
公钥α, β, p，私钥a如ElGamal体制所定义，加密如下定义：
选取随机数$x \in \mathbb{Z}_p^*$，

$$e_k(x, k) = (y_1, y_2)$$

其中

$$y_1 = \alpha^k \pmod p$$

且

$$y_2 = x + \beta^k \pmod p$$

要求：
（1）给出解密运算
（2）课本中已证明ElGamal体制具有如下结论：任何解CDH的算法，都可以用于解密密文，反之亦然。请证明该结论对以上所定义的加密体制同样成立。

第6章作业

课本习题：6.20
思考题：6.22

6.1 ElGamal密码体制

- ElGamal密码体制的提出：由Taher Elgamal 在1985 提出。
- 应用例子：GnuPG(GNU Privacy Guard)免费软件、PGP及其它加密系统。
- 所基于的困难问题：离散对数问题
- 学习难点：一般交换群上的ElGamal体制涉及到抽象代数的基础概念和有限域的基本运算。

对乘法群$(G,\cdot)$中的n阶元素α，定义：

$\alpha \in G$**的阶** 如果n是满足$\alpha^n=1$的最小正整数，则称n为α的阶。记$<\alpha>=\{\alpha^i : 0 \le i \le n-1\}$，则$<\alpha>$为循环群，$\alpha$为其本原元（或生成元）。

$\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$，其中$p$为素数

模p本原元(或原根) 如果$\alpha \in \mathbb{Z}_p^*$的阶为$p-1$，则称$\alpha$为模$p$本原元，这时有$\{\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{p-1} \bmod p\} = \mathbb{Z}_p^*$。

6.1 ElGamal密码体制

离散对数问题

设α为乘法群$(G,\cdot)$上的n阶元素，
且$\beta \in \langle\alpha\rangle = \{1,\alpha,\alpha^2,\ldots,\alpha^{n-1}\}$。求$a(0 \le a \le n-1)$满足

$$\alpha^a = \beta$$

a记为$a = \log_\alpha \beta$，称为β的离散对数。

Diffie-Hellman密钥协商方案：
公开参数：乘法群G，n阶元素$\alpha \in G$

Alice	Bob

Alice		Bob
		选择随机数$a \in \mathbb{Z}_n$
	$\xleftarrow{\beta=\alpha^a}$	
选择随机数$k \in \mathbb{Z}_n$		
	$\xrightarrow{y_1=\alpha^k}$	
计算共享密钥$K = \beta^k = (\alpha^a)^k$		计算$K = {y_1}^a = (\alpha^k)^a$

把Diffie-Hellman密钥协商方案修改成公钥密码体制：

公开参数：乘法群G，n阶元素$\alpha \in G$

Alice　　　　　　　　　　　　　　　　　　　　　　　　Bob

选择私钥$a \in \mathbb{Z}_n$

$$\xleftarrow{\text{公钥：} \beta=\alpha^a}$$

选择随机数$k \in \mathbb{Z}_n$

计算$y_1 = \alpha^k$

$$\text{计算共享密钥}K \begin{aligned}[t] &= \beta^k \\ &= (\alpha^a)^k \end{aligned}$$

加密消息x：$y_2 = xK = x\beta^k$

$$\xrightarrow{\text{密文：} y_1, y_2}$$

$$\text{计算}K \begin{aligned}[t] &= {y_1}^a \\ &= (\alpha^k)^a \end{aligned}$$

$$\text{解密：} \quad x \begin{aligned}[t] &= y_2 K^{-1} \\ &= y_2 (y_1^a)^{-1} \end{aligned}$$

$\mathbb{Z}_p^*$上的ElGamal密码体制

- 私钥a：$a \in \mathbb{Z}_p^*$
- 公钥(p, α, β)：素数p，模p本原元α，$\beta = \alpha^a \bmod p$
- 加密运算：选取随机数$k \in \mathbb{Z}_p^*$，定义：

$$e_K(x, k) = (y_1, y_2)$$

其中

$$y_1 = \alpha^k \bmod p$$

$$y_2 = x\beta^k \bmod p$$

- 解密运算：

$$d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p$$

- 解密运算有效性证明：

$$y_2(y_1^a)^{-1} \equiv x\beta^k(\alpha^{ka})^{-1} \equiv x\beta^k(\beta^k)^{-1} \equiv x \pmod p$$

另一证明（看成群$\mathbb{Z}_p^*$的元素）：

$$\frac{y_2}{y_1^a} = \frac{x\beta^k}{\alpha^{ka}} = \frac{x\beta^k}{\beta^k} = x$$

- 思考（本章作业E6.1(1)）：如果加密运算中y_2如下计算：

$$y_2 = x + \beta^k \pmod p$$

请求出对应的解密运算

- 由公钥p, β, α求解私钥a是$\mathbb{Z}_p^*$上的离散对数问题：$a = \log_\alpha \beta$，该问题认为是计算困难的，这样如果选取足够大的素数p（至少300个十进制位），敌手就无法从公钥计算出私钥。

Example

例6.1： 设ElGamal密码体制：$p = 2579, \alpha = 2, a = 765$，那么

$$\beta = \alpha^a \pmod p = 2^{765} \pmod{2579} = 949$$

(一)加密：Alice想给Bob发送消息$x = 1299$，她进行如下加密运算：
(1)选取随机数k，如$k = 853$。
(2)计算：

$$\begin{aligned} y_1 &= 2^{853} \pmod{2579} \\ &= 435 \\ y_2 &= 1299 \times 949^{853} \pmod{2579} \\ &= 2396 \end{aligned}$$

(二)解密：Bob收到Alice发来的密文$y = (435, 2396)$后，如下计算明文：

$$\begin{aligned} x &= 2396 \times (435^{765})^{-1} \pmod{2579} \\ &= 1299 \end{aligned}$$

ElGamal密码体制的**密钥生成过程**：

1. 产生素数$p = rq_0 + 1$，q_0为大素数，r是小的整数：素性测试算法
2. 产生模p本原元α：产生随机数$\alpha \in \mathbb{Z}_p^*$，根据以下定理判定$\alpha$是否为本原元

Theorem

(定理5.8)如果$p > 2$是素数，且$\alpha \in \mathbb{Z}_p^*$。那么$\alpha$是模$p$的本原元当且仅当

$$\alpha^{\frac{p-1}{q}} \not\equiv 1 \pmod{p}, \text{ 对所有素数} q | p-1$$

成功概率估算：$\mathbb{Z}_p^*$中本原元的数量为$\phi(p-1)$，那么随机数α为本原元的概率$= \frac{\phi(p-1)}{p} \geq \frac{q_0-1}{q_0 r+1} > \frac{1}{r+1}$。

引理1：若α是群G中的m阶元素。对$n > 0$，$\alpha^n = 1$当且仅当$m|n$。
证明：设$n = mq + r(0 \le r < m)$。那么$1 = \alpha^n = \alpha^{mq+r} = \alpha^r$，因此$r = 0$（否则与$0 \le r < m$矛盾），即$m|n$。
引理2：设p是素数且α是模p本原元，那么$\beta = \alpha^i$的阶为：

$$\frac{p-1}{\gcd(p-1, i)}$$

证明：设β的阶为d。那么$1 = \beta^d = \alpha^{id}$，由引理1得到$p-1|id$，即$\frac{p-1}{\gcd(p-1,i)}|d$。另一方面$\beta^{\frac{p-1}{\gcd(p-1,i)}} = \alpha^{i\cdot\frac{p-1}{\gcd(p-1,i)}} = 1$，由引理1得到$d|\frac{p-1}{\gcd(p-1,i)}$。从而命题成立。
定理5.8证明：令g是模p本原元。设$\alpha \in \mathbb{Z}_p^*$模$p$的阶为$d$，且$\alpha = g^i$ $(1 \le i \le p-2)$。由引理2知道，α的阶为$\frac{p-1}{\gcd(p-1,i)}$。那么α为本原元当且仅当$\gcd(p-1, i) = 1$，即对所有$q|p-1$，都有$\gcd(q, i) = 1 \Longleftrightarrow \alpha^{\frac{p-1}{q}} = g^{i\cdot\frac{p-1}{q}} \not\equiv 1 \pmod p$

3. 产生随机数$a \in \mathbb{Z}_p^*$
4. 计算$\beta = \alpha^a \pmod{p}$：平方-乘算法

加密过程：

1. 产生随机数$k \in \mathbb{Z}_p^*$
2. 用平方-乘算法计算：

$$y_1 = \alpha^k \bmod p$$

$$y_2 = x\beta^k \bmod p$$

解密过程：

1. 用平方-乘算法计算：

$$x = y_2 \cdot y_1^{p-1-a} \bmod p$$

注意：$y_1^{-a} \bmod p = y_1^{p-1-a} \bmod p$，比先计算$y^a$再求逆省去了求逆运算。

交换群/乘法群G上的ElGamal公钥加密体制

- 私钥a：$a \in \{1, \ldots, n-1\}$
- 公钥(G, α, β)：交换群G，G上的n阶元素α，$\beta = \alpha^a$
- 加密运算：选取随机数$k \in \{1, \ldots, n-1\}$，定义：

$$e_K(x, k) = (y_1, y_2)$$

其中

$$y_1 = \alpha^k$$

$$y_2 = x\beta^k$$

- 解密运算：

$$d_k(y_1, y_2) = y_2(y_1^a)^{-1}$$

群的定义

Definition

群(Group)$(G,\cdot)$，其中G为元素集合，$\cdot$是定义在G上的二元运算，满足以下性质：

1. 封闭性：对任意元素$a,b\in G$有$a\cdot b\in G$
2. 结合性：对任意元素$a,b,c\in G$有$(a\cdot b)\cdot c=a\cdot(b\cdot c)$
3. 存在单位元素：存在元素$e\in G$，使得对于任意$a\in G$有$e\cdot a=a\cdot e=a$，该元素称为单位元素，经常用1表示。
4. 可逆性：对任意$a\in G$，存在元素$b\in G$有$a\cdot b=b\cdot a=1$。元素b称为a的逆元(可以证明逆元是唯一的)。

注：为了简化表示，$a\cdot b$经常写成ab。

乘法群（或称为交换群、阿贝尔群(Abel群)）：群$(G,\cdot)$称为乘法群，若满足交换性，即对任意$a,b\in G$有$ab=ba$。

群的例子

Example

整数集合$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$，以及定义在其上的加法运算$+$，我们可以验证满足：

- 封闭性：对任意整数a, b，$a + b$也是整数。
- 结合性：对任意整数a, b, c有$(a + b) + c = a + (b + c)$。
- 存在单位元素：0是单位元，因为对任意整数a有$0 + a = a + 0 = a$。
- 可逆性：对任意整数a，逆元为$-a$，因为$a - a = -a + a = 0$。
- 交换性

因此$(\mathbb{Z}, +)$是乘法群。

群的例子

Example

设p为素数，$\cdot$定义为mod$p$的乘法运算，$\mathbb{Z}_p^* = \mathbb{Z}_p/\{0\}$。那么$(\mathbb{Z}_p^*, \cdot)$是群。可以验证有：

- 封闭性：对任意$a, b \in \mathbb{Z}_p^*$，$ab \in \mathbb{Z}_p^*$。
- 结合性：对任意$a, b, c \in \mathbb{Z}_p^*$有$(ab)c = a(bc)$。
- 存在单位元素：1是单位元，因为对任意$a \in \mathbb{Z}_p^*$有$1a = a1 = a$。
- 可逆性：对任意$a \in \mathbb{Z}_p^*$，逆元为a^{p-2}，因为由费马小定理可得$aa^{p-2} = a^{p-2}a = a^{p-1} = 1$。
- 交换性

群的例子

Example

设n为正素数，$+$定义为$\bmod n$的加法运算。那么$(\mathbb{Z}_n, +)$是乘法群。证明作为练习。

交换群

不是所有群上的离散对数问题都是难解的：

Example

$(\mathbb{Z}_n, +)$上的离散对数问题是容易解的：

- $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$
- 运算：$x + y \bmod n$
- x的k次幂运算x^k为$x + x + \ldots + x \bmod n = kx \bmod n$，由$y = xk \bmod n$得到：

$$k = \log_x y = \begin{cases} yx^{-1} \bmod n & \text{当} d = 1 \\ y'x'^{-1} \bmod n' + id & \text{当} 1 < d < n, d|y \\ \text{任意值} & \text{当} d = n \\ \text{无解} & \text{当} d > 1, d \nmid y \end{cases}$$

其中$d = (x, n), x' = \frac{x}{d}, y' = \frac{y}{d}, n' = \frac{n}{d}$。

交换群

以下群中的离散对数问题认为是难解的，在密码应用中最为重要：

1. $G=(\mathbb{Z}_p^*,\cdot)$，p是素数，α是模p的一个本原元。
2. $G=(\mathbb{Z}_p^*,\cdot)$，p,q是素数，$q|p-1$，α是$\mathbb{Z}_p$的一个q阶元素。
3. $G=(\mathbb{F}_{2^n}^*,\cdot)$，$\mathbb{F}_{2^n}^*$表示有限域$\mathbb{F}_{2^n}$的乘法群，$\alpha$是$\mathbb{F}_{2^n}^*$的一个本原元。
4. $G=(E,+)$，其中E是模素数p的一个椭圆曲线，$\alpha\in E$是一个具有素数$q=\#E/h$阶的点，这里（典型的）$h=1,2$或4。
5. $G=(E,+)$，其中E是有限域$\mathbb{F}_{2^n}^*$上的椭圆曲线，$\alpha\in E$是一个具有素数$q=\#E/h$阶的点，这里（典型的）$h=2$或4。

$G=(\mathbb{Z}_p^*,\cdot)$上的Elgamal密码体制

为了提高计算效果，又不失安全性，α可取为q阶元素，其中$q|p-1$且q为素数

- 如何计算q阶元素：如果α是本原元，即α为$p-1$阶元素，那么$\alpha^{\frac{p-1}{q}}$为q阶元素。
- 改进指数运算效率：如α是q阶元素，那么$\alpha^a=\alpha^{a \bmod q}$，降低了指数运算次数。
- 不降低安全性：一般认为$(\mathbb{Z}_p,\cdot)$的q阶子群上的离散对数问题(即取α为q阶元素)和$(\mathbb{Z}_p,\cdot)$上的离散对数问题(即取α为本原元) 难度一样。

有限域（不考试）

有限域的应用：

- 构造Elgamal公钥体制
- AES
- 数字签名、认证等密码协议应用
- 纠错码。。。

我们学习过的域(Field)：

- 无限域（元素个数无限）：
  1. 实数域$(R, +, \cdot)$
  2. 有理数域$(Q, +, \cdot)$
  3. 复数域$(C, +, \cdot)$
- 有限域（元素个数有限个）：
  1. 模素数p剩余系：$(\mathbb{Z}_p, +, \cdot)$

有限域

域$(\mathbb{F}, +, \cdot)$的定义：

- $\mathbb{F}$为元素集合，如果数量有限则称为有限域。
- 有两个F上的二元运算：$+, \cdot$
- 关于运算+构成交换群，即$(\mathbb{F}, +)$是交换群，其中单位元记为0。
- 记$\mathbb{F}^* = \mathbb{F}/\{0\}$，那么$(\mathbb{F}^*, \cdot)$是交换群，其中单位元记为1。
- 满足分配律：对任意$x, y, z \in \mathbb{F}$有：$(x + y)z = xz + yz$

有限域

有限域的基本性质：

- 性质1：有限域的元素个数为p^n，其中p为素数
- 性质2：元素个数相同的有限域同构，即实质上元素个数为p^n的有限域是唯一的，仅是符号表示不同。
- 因此把元素个数为p^n的有限域记为$\mathbb{F}_{p^n}$，或者$GF(p^n)$，而$\mathbb{F}_p$通常写成$\mathbb{Z}_p$

有限域

有限域$\mathbb{F}_{p^n}$的构造：

1. $\mathbb{Z}_p$上的多项式集合记为$\mathbb{Z}_p[x]$，即

$$\mathbb{Z}_p[x] = \{a_n x^n + \ldots + a_0, a_i \in \mathbb{Z}_p, i = 0, \ldots, n, n = 0, 1, \ldots\}$$

2. 找到$\mathbb{Z}_p$上一个次数为n的不可约多项式f(x)：“不可约多项式”类似于整数中的“素数”，即不存在次数不为0的多项式$f_1(x), f_2(x) \in \mathbb{Z}_p[x]$，满足

$$f(x) = f_1(x) f_2(x)$$

3. $\mathbb{F}_{p^n} = \{a_{n-1}x^{n-1} + \ldots + a_1 x + a_0, a_i \in \mathbb{Z}_p, i = 0, \ldots, n-1\}$
4. “加法”运算+定义为：对元素$\alpha = g_1(x), \beta = g_2(x)$，定义

$$\alpha + \beta = (g_1(x) + g_2(x)) \bmod f(x)$$

5. “乘法”运算·定义为：对元素$\alpha = g_1(x), \beta = g_2(x)$，定义

$$\alpha \cdot \beta = (g_1(x) g_2(x)) \bmod f(x)$$

有限域$\mathbb{F}_{p^n}$的构造的补充说明：

- 多项式$f(x)$的最高次数记为$\deg(f(x))$
- $g(x) \bmod f(x)$定义：如果

$$g(x) = q(x)f(x) + r(x), q(x), r(x) \in \mathbb{Z}_p[x]$$

且$\deg(r(x)) < \deg(f(x))$，那么定义

$$g(x) \bmod f(x) = r(x)$$

- 幂乘运算（指数运算）：$g(x)^k \pmod{f(x)}$使用平方-乘算法提高效率，注意：对应的运算改成$\mathbb{F}_{p^n}$上的运算。

有限域$\mathbb{F}_{p^n}$的构造的补充说明：
设$g(x) = g_{n-1}x^{n-1} + \ldots + g_1 x + g_0$

- $g(x)$关于加法+运算的逆元为：$-g(x)$，即

  $$-g(x) = (-g_{n-1} \bmod p)x^{n-1} + \ldots + (-g_1 \bmod p)x + (-g_0 \bmod p)$$

  如果$p = 2$，$-g(x) = g(x)$。
- 计算$g(x)$关于乘法运算的逆元：即求$g^{-1}(x)$满足$g^{-1}g(x) \equiv 1 \bmod f(x)$，和$\mathbb{Z}_p^*$一样，$g(x)$使用扩展Euclidean算法求逆元，注意：对应的运算改成$\mathbb{F}_{p^n}$上的运算。

Example

例6.8：构造$\mathbb{F}_{2^3}$

1. 找到$\mathbb{Z}_2$上一个次数为3的不可约多项式$f(x) = x^3 + x + 1$
2. $\mathbb{F}_{2^3}$的$2^3 = 8$个元素定义为8个次数小于3的多项式：

$$\{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$$

3. 加法运算：元素$\alpha = a_2x^2 + a_1x + a_0$，$\beta = b_2x^2 + b_1x + b_0$，因为$\alpha, \beta$的次数小于$f(x)$，那么

$$\alpha + \beta = (a_2 \oplus b_2)x^2 + (a_1 \oplus b_1)x + (a_0 \oplus b_0)$$

注意：mod2加法运算其实是异或$\oplus$运算。例子：

$$\begin{aligned}(x^2+x+1) + (x+1) \bmod f(x) &= x^2 \\ (x^2+x+1) + (x^2+1) \bmod f(x) &= x\end{aligned}$$

Example

例6.6：构造$\mathbb{F}_{2^3}$（续）

1. 乘法运算：首先在$\mathbb{Z}_2[x]$中计算乘积，
如$\alpha = x^2+1, \beta = x^2+x+1$，那么

$$(x^2+1)(x^2+x+1) = x^4+x^3+2\cdot x^2+x+1 = x^4+x^3+x+1$$

然后乘积结果mod$(f(x) = x^3+x+1)$：

$$x^4+x^3+x+1 = (x+1)(x^3+x+1)+(x^2+x)$$

$$\begin{pmatrix} g(x) & x^4 & +x^3 & & +x & +1 \\ xf(x) & x^4 & & +x^2 & +x & \\ g(x)-xf(x) & & x^3 & +x^2 & & +1 \\ f(x) & & x^3 & & +x & 1 \\ g(x)-xf(x)-f(x) & & & x^2 & +x & \end{pmatrix}$$

因此$(x^2+1)(x^2+x+1) = x^2+x$

Example

例6.8：构造$\mathbb{F}_{2^3}$（续）

1. 如果把多项式$a_2x^2+a_1x+a_0$表示成三元组$a_2a_1a_0$，那么课本中的例6.6的表格给出了所有元素的乘法运算结果。
2. 关于乘法运算的逆元计算：应用扩展Euclidean算法计算。
   如求x^2的逆：

i	r_j	q_j	s_j	t_j
0	x^3+x+1		1	0
1	x^2	x	0	1
2	$x+1$	x	1	x
3	x	1	x	x^2+1
4	1	x	$x+1$	x^2+x+1

因此$(x+1)(x^3+x+1)+(x^2+x+1)(x^2)=1$，
则x^2的逆元为x^2+x+1

6.5 椭圆曲线
6.5.2 模素数的椭圆曲线（不考试）

定义6.4 $\mathbb{Z}_p$上的椭圆曲线E

$p > 3$是素数，则E包含了无穷远点$\mathcal{O}$，以及$\mathbb{Z}_p$上的同余方程

$$y^2 \equiv x^2 + ax + b \pmod{p}$$

的所有解$(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$。其中$a, b \in \mathbb{Z}_p$是满足$4a^3 + 27b^2 \not\equiv 0$的常量。

6.5 椭圆曲线
6.5.2 模素数的椭圆曲线

E上的加法运算定义如下：

- 对$P=(x_1,y_1), Q=(x_2,y_2)\in E$，定义

$$P+Q=\begin{cases} \mathcal{O} & \text{如果} x_1=x_2, y_2=-y_1 \\ (x_3,y_3) & \text{否则} \end{cases}$$

其中

$$\begin{aligned} x_3 &= \lambda^2-x_1-x_2 \\ y_3 &= \lambda(x_1-x_2)-y_1 \end{aligned} \qquad \lambda=\begin{cases} (y_2-y_1)(x_2-x_1)^{-1} & P\neq Q \\ (3x_1^2+a)(2y_1)^{-1} & P=Q \end{cases}$$

- 对$P\in E$，定义

$$P+\mathcal{O}=\mathcal{O}+P=P$$

可以证明，$(E,+)$是阿贝尔群，$\mathcal{O}$为单位元。

已知：α是乘法群G上的n阶元素，$\beta \in \langle\alpha\rangle$。求：$a = \log_\alpha \beta$
算法思路：$m = \sqrt{n}$。假设$a = mj + i, 0 \le i < m$。那么

$$\beta = \alpha^a = \alpha^{mj+i} \Longleftrightarrow \alpha^{mj} = \beta\alpha^{-i}$$

1. 计算表格：

$$\begin{array}{|c|c|} 0 & \alpha^0 \\ 1 & \alpha^m \\ 2 & \alpha^{2m} \\ \dots & \dots \\ j & \alpha^{jm} \\ \dots & \dots \\ m-1 & \alpha^{(m-1)m} \\ \hline \end{array} \quad (1)$$

2. 对表格(1)的第2列坐标排序
3. 对$i(i \le 0 < m)$，查找$\beta\alpha^{-i}$是否在表(1)的第2列中，如果有$\alpha^{mj} = \beta\alpha^{-i}$，那么$a = mj + i$。

问题6.2（离散对数第i比特问题）

已知：素数p，$\alpha \in \mathbb{Z}_p^*$是本原元，$\beta \in \mathbb{Z}_p^*$，整数$i(1 \le i \le \lceil lb(p-1) \rceil)$。
求：$L_i(\beta) = \log_\alpha \beta$的二进制表示的第$i$个最低比特，即若

$$\log_\alpha \beta = x_{\lceil lb(p-1) \rceil} 2^{\lceil lb(p-1) \rceil} + \ldots + x_2 x^2 + x_1 2 + x_0, x_i \in \{0, 1\}$$

那么$L_i(\beta) = x_{i-1}$

结论：求$L_1(\beta)$容易。
证明：称β为模p二次剩余当且仅当存在x有$\beta \equiv x^2 \pmod p$，可证β为二次剩余当且仅当$\log_\alpha \beta$为偶数。

$$L_1(\beta) = x_0 = \begin{cases} 0 & \text{当且仅当} \log_\alpha \beta \text{为奇数} & \Leftrightarrow \beta^{\frac{p-1}{2}} \equiv 1 \pmod p \\ 1 & \text{当且仅当} \log_\alpha \beta \text{为偶数} & \Leftrightarrow \beta^{\frac{p-1}{2}} \not\equiv 1 \pmod p \end{cases}$$

假设$p-1=2^s t$，t是奇数。有以下结论：

1. 可以证明对$i \le s$，容易计算$L_i(\beta)$
2. 任何计算$L_{s+1}(\beta)$的假设算法都可以用于计算$\log_\alpha \beta$。

现对$p \equiv 3 \pmod 4$（此时$s=1$）证明结论2。
假设β是二次剩余，那么存在偶数a，有$\beta \equiv \alpha^a \pmod p$，现在证明如果知道$L_2(\beta)$，则容易计算$\alpha^{a/2}$。
因为$p \equiv 3 \pmod 4$，整数$\frac{p-1}{2}$是奇数，那么$\alpha^{\frac{p-1}{2}} = -1$不是二次剩余。有

$$\beta^{\frac{p-1}{2}} \equiv 1 \pmod p \Leftrightarrow \beta^{\frac{p+1}{2}} \equiv \beta \pmod p \Rightarrow \beta\text{的平方根} = \pm\beta^{\frac{p+1}{4}} \bmod p$$

那么

$$\alpha^{a/2} = \begin{cases} \beta^{\frac{p+1}{4}} \bmod p & \text{若} L_2(\beta) = L_1(\beta^{\frac{p+1}{4}} \bmod p) \\ -\beta^{\frac{p+1}{4}} \bmod p & \text{否则} \end{cases}$$

上式计算理由是：$L_2(\beta) = L_1(\alpha^{a/2})$，并且因为$-1$不是二次剩余(而1是二次剩余)，$L_1(\pm\beta^{\frac{p+1}{4}} \bmod p)$为不同值。

（续）

设$a = \log_\alpha \beta = \sum_{i \geq 0} x_i 2^i$为二进制表示，则

$$\beta = \alpha^a = \alpha^{x_0+2x_1+2^2x_2+x^3x_3+x_42^4 \ldots}$$

那么：

$$x_0 = L_1(\beta), x_1 = L_2(\beta/\alpha^{x_0}), x_2 = L_2(\alpha^{\frac{a-x_0}{2}}/\alpha^{x_1})$$

$$x_3 = L_2(\alpha^{\frac{a-x_0-x_12}{2^2}}/\alpha^{x_2}), x_4 = L_2(\alpha^{\frac{a-x_0-x_12-x_22^2}{2^3}}/\alpha^{x_3}), \ldots$$

我们只讨论$\mathbb{Z}_p^*$上的ElGamal密码体制。
下面比较攻击ElGamal密码体制和以下2个问题的难度比较：

问题6.3 Computational Diffie-Hellman(CDH)：

设α是乘法群$(G,\cdot)$上的n阶元素，选择随机数(a,b)，给定

$$(\alpha, \alpha^a, \alpha^b)$$

求：α^{ab}

问题6.4 Decision Diffie-Hellman(DDH)：

设α是乘法群$(G,\cdot)$上的n阶元素，给定

$$(\alpha^b, \alpha^c, \alpha^d)$$

判断是否有：$d = bc \bmod n$，或者$\alpha^{bc} = \alpha^d$?

计算难度比较：$DDH \le CDH \le$ 离散对数问题

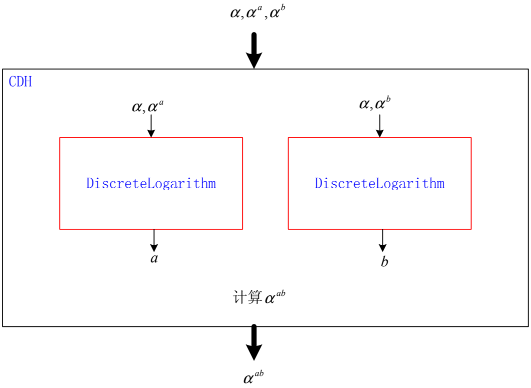
2. 证明$CDH \propto_T DiscreteLogarithm$：即证明存在多项式时间算法把$CDH$问题规约为离散对数问题，证明如下：

已知：$(\alpha, \alpha^a, \alpha^b)$

1 利用解决离散对数的算法计算：a, b

2 计算α^{ab}

现在我们证明了离散对数问题至少和CDH一样困难。



6.7 ElGamal密码体制的安全性

计算难度比较：$DDH \leq CDH \leq$ 离散对数问题

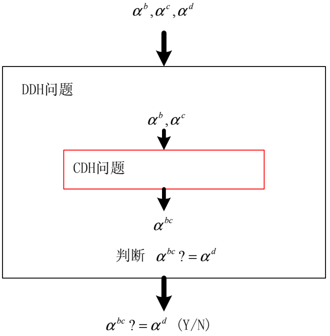
1. 证明$DDH \propto_T CDH$：即证明存在多项式时间算法把DDH问题规约为CDH问题，证明如下：

   已知：$(\alpha^b, \alpha^c, \alpha^d)$

   1 利用解决CDH的算法计算α^{bc}

   2 判定$\alpha^{bc} = \alpha^d$是否成立

现在我们证明了CDH至少和DDH一样困难。



任何解CDH的算法，都可以用于解密ElGamal密文，反之亦然，即未知私钥情况下，解密ElGamal密文$\Longleftrightarrow$CDH。

证明：

1.证明任何解CDH的算法，都可以用于解密ElGamal密文：

1. 假设OracleCDH是解CDH的一个算法
2. 假设：ElGamal的公钥为α, β, p，私钥a
3. 设$y_1 = \alpha^k, y_2 = x\beta^k$是ElGamal密码的密文，如下计算明文$x$：

$$\delta = OracleCDH(\alpha, \beta, y_1) = OracleCDH(\alpha, \alpha^a, \alpha^k) = \alpha^{ak} = \beta^k$$

计算

$$x = \delta^{-1} \cdot y_2 \equiv \beta^{-k} \cdot x\beta^k$$

（证明续）2.证明任何解密ElGamal密文的算法，都可以用于解CDH：

1. 假设Oracle-Elgamal-Decrypt是解密ElGamal密文的一个算法
2. 假设：CDH的输入为：$\alpha, \beta = \alpha^a, \gamma = \alpha^b$
3. 可如下计算CDH的输出α^{ab}：令ElGamal的参数如下：公钥α, β，密文$y_1 = \gamma, y_2 \in \langle\alpha\rangle$为随机选定，计算：

$$
\begin{aligned}
x &= Oracle - Elgamal - Decrypt(\alpha, \beta = \alpha^a, (y_1 = \alpha^b, y_2)) \\
&= y_2 y_1^{-a} \quad \text{(解密运算)} \\
&= y_2 \cdot \alpha^{-ab}
\end{aligned}
$$

然后计算

$$\delta = y_2 x^{-1} = y_2 \cdot (y_2 \cdot \alpha^{-ab})^{-1} = \alpha^{ab}$$