ElGamal 密码

El Gamal Cryptosystem

刘卓

在密码学中，ElGamal 加密算法是一个基于迪菲-赫尔曼密钥交换的非对称加密算法。它在 1985 年由塔希尔 · 盖莫尔提出。GnuPG 和 PGP 等很多密码学系统中都应用到了 ElGamal 算法。

ElGamal 加密算法可以定义在任何循环群 G 上。它的安全性取决于 G 上的离散对数难题。

1 离散对数问题

有一个素数 p 和一个整数 α,α 不能被 p 整除。那么如果给一个整数 β，则:

$$\alpha^x \equiv \beta \pmod p$$

。

即代表 x 有多组解。因为 $3^x = 11 \Leftrightarrow x = log_3(11)$, 但 $3^x = 11 \pmod{18}$ 就代表有很多解。

定义 *1*: *如果对于所有的 i 满足 $a^i \equiv 1 \pmod p$，那么将 a 称之为primitive root mod*

定理 *1*: *如果 a 是primitive root mod p，则 $a^1, a^2, \cdots, a^{p-1} \pmod p$*

莫比斯公式:

$$\mu(d) = \begin{cases} (-1)^k & \text{如果 d 是 k 个不同质数的乘积} \\ 0 & \text{其他} \end{cases}$$

定义 *2*: *分圆多项式 (Cyclotomic polynomial):*

$$C_n(x) = \prod_{m|n} (1 - x^m)^{\mu(\frac{n}{m})}$$

定理 *2*: *对于每一个素数 p，a 是primitive root mod p 当且仅当 $C_{p-1}(a) \equiv 0 \pmod p$*

例 1:

$p = 11$

$$C_{p-1}(2) = C_{10}(2) = 2^4 - 2^3 + 2^2 - 2^1 + 2^0 \pmod{11} \equiv 11 \pmod{11} \equiv 0 \pmod{11}$$

$$C_{p-1}(3) = C_{10}(3) = 3^4 - 3^3 + 3^2 - 3^1 + 3^0 \pmod{11} \equiv 61 \pmod{11} \not\equiv 0 \pmod{11}$$

因此 2 是一个primitive root mod 11，3 不是一个primitive root mod 11。

□

例 2：

$C_{18}(x), n = 18$

m	1	2	3	6	9	18
$\frac{18}{m}$	18	9	6	3	2	1
$\mu\left(\frac{18}{m}\right)$	0	0	1	-1	-1	1

$$C_{18}(x) = \frac{(1-x^{18})(1-x^3)}{(1-x^9)(1-x^6)} = \frac{(1+x^9)}{(1+x^3)} = \frac{(1+x^3)(1-x^3+x^6)}{(1+x^3)} = 1 - x^3 + x^6$$

$C_{10}(x), n = 10$

m	1	2	5	10
$\frac{10}{m}$	10	5	2	1
$\mu\left(\frac{10}{m}\right)$	1	-1	-1	1

$$C_{10}(x) = \frac{(1-x^{10})(1-x^1)}{(1-x^5)(1-x^2)} = \frac{(1+x^5)}{(1+x)} = x^4 - x^3 + x^2 - x + 1$$

□

定理 3： *如果 a 是primitive root mod p，则primitive root mod p的集合是：*

$$a^r : 1 \le r \le p-1, gcd(r, p-1) = 1$$

即如果知道其中一个 primitive root，如何找到所有 primitive root。

例 3：

找到所有的 primitive root mod 11，已知 2 是其中一个 primitive root。

k	1	2	3	4	5	6	7	8	9	10
$2^k \pmod{11}$	2	4	8	5	10	9	7	3	6	1

因为 $1, 3, 7, 9$ 和 $p - 1 = 11 - 1 = 10$ 互质，所以他们的 primitive root 就是他们的幂余 $= 2, 8, 7, 6$

□

例 4：

找到所有的 primitive root mod 11 后

k	1	2	3	4	5	6	7	8	9	10
$2^k \pmod{11}$	2	4	8	5	10	9	7	3	6	1

找到 $9x \equiv 5 \pmod{11}$ 的解
因为 2 是其中一个 primitive root, 所以 $x = 2^y$

$$\begin{aligned} 9x &\equiv 5 \pmod{11} \\ 2^{6} \cdot 2^y &\equiv 2^{4} \pmod{11} \\ 2^{6+y} &\equiv 2^4 \pmod{11} \\ 2^{2+y} &\equiv 1 \pmod{11} \end{aligned}$$

所以 $2 + y \equiv 0 \pmod{\mu(11)} = 0 \pmod{10} \Rightarrow y = 8$

$$x = 2^8 \pmod{11} = 3$$

□

例 5:
找到所有的 primitive root mod 11 后

k	1	2	3	4	5	6	7	8	9	10
$2^k \pmod{11}$	2	4	8	5	10	9	7	3	6	1

找到 $7^x \equiv 5 \pmod{11}$ 的解
因为 7 是其中一个 primitive root, 所以 $x = 7^y$

$$\begin{aligned} 7^x &\equiv 5 \pmod{11} \\ 2^{7^x} &\equiv 2^4 \pmod{11} \\ 2^{7x} &\equiv 2^4 \pmod{11} \\ 2^{7x-4} &\equiv 1 \pmod{11} \end{aligned}$$

所以 $7x - 4 \equiv 0 \pmod{\mu(11)} \Rightarrow 7x \equiv 4 \pmod{10} \Rightarrow x = 7^{-1} \cdot 4 = 3 \cdot 4 = 12 = 2 \pmod{10}$

$$x = 2$$

注意 7^{-1} 不是幂倒数而是余倒数。

□

定义 *3*: *如果一个素数 p 和一个 primitive root α mod p，给定一个整数 β，求 x 的方法是:*

$$\alpha^x \equiv \beta \pmod{p}$$

$$x \equiv log_\alpha(\beta) \pmod{p}$$

所以给定一个素数 p，找到一个 primitive root 是相当容易的。对于小的 p，我们可以通过穷举搜索来计算。但一般来说计算离散对数是困难的 (没有已知的多项式时间算法)。

2 ElGamal 密码

加密步骤如下：

1. *Alice* 和 *Bob* 共同决定一个质数 p 和一个 primitive root $r \bmod p$
2. *Alice* 从 $\{1, 2, \cdots, p-1\}$ 中选择 a，并计算公钥 $\alpha = r^a \pmod p$
3. *Bob* 从 $\{1, 2, \cdots, p-1\}$ 中选择 b，并计算公钥 $\beta = r^b \pmod p$
4. 假设 *Bob* 想发送一段信息 X 给 *Alice*。则从 $\{2, \cdots, p-2\}$ 选择一个会话密钥 k
5. *Bob* 计算 $U = r^k \pmod p$ 和 $V = \alpha^k X = r^{ak} X \pmod p$，然后将 (U, V) 发送给 *Alice*
6. *Alice* 解密密文，计算 $U^{-a} V = r^{-ka} \alpha^k X = r^{-ka} r^{ak} X = X$

例 6：

1. 假设 *Alice* 和 *Bob* 共同决定一个质数 $p = 11881379, r = 23$
2. *Alice* 选择 $a = 55$ 计算公钥 $\alpha = r^a \pmod p = 23^{55} \pmod p = 1308503 \pmod{11881379}$
3. 要加密信息 $X = \textbf{LUNCH}$, 则需要把 **LUNCH** 转化为数字。从 0 开始，按照字母表顺序，$L = 11, U = 20, N = 13, C = 2, H = 7$,

$$X = 11 \cdot 26^4 + 20 \cdot 26^3 + 13 \cdot 26^2 + 2 \cdot 26^1 + 7 \cdot 26^0 = 5387103$$

4. *Bob* 选择会话密钥 $k = 123$，然后计算

$$U = r^k \pmod p = 23^{123} \pmod{11881379} = 1777907 \pmod{11881379}$$

和

$$V = \alpha^k X = r^{ak} X \pmod p = 1308503^{123} \cdot 5387103 = 4944577 \pmod{11881379}$$

发送 $(U, V) = (1777907, 4944577)$ 给 *Alice*

5. 还原密文，*Alice* 计算

$$U^{-a} = U^{-55} = U^{p-1-55} = 1777907^{11881323} = 7112147 \pmod{11881379}$$

$$X = U^{-a} V = 7112147 \cdot 4944577 = 5387103 \pmod{11881379}$$

6. $X = 5387103 = \textbf{LUNCH}$

```
#Python 求 幂次模余
pow(1777907, 11881323, 11881379)#a^n%p, pow(a,n,p)
```

$\square$

如果攻击方想截取信息，那么能做什么?

假设 *Eve* 截获了 *Bob* 发送的 (U, V)，为了破译消息，*Eve* 需要知道 *Alice* 选择的整数 a。Eve 有两个选择:

- 解开未知数为 a 的方程 $\alpha = r^a \pmod p$
- 解开未知数为 k 的方程 $U = r^k \pmod p$

如果可以解开方程，则可以计算 $\alpha^{-k}V = \alpha^{-k}\alpha^k X = X$ 即破译成功。

但因为离散对数的问题很难破解，所以很难解开方程组。

不过 *Eve* 不能找出 *Bob* 发送给 *Alice* 的是什么，以混淆她发送给 *Alice* 的问题。这也是 ElGamal 密码其中一个问题。

例 7:

继续例 6 的问题。

Eve 拦截了信息 (U, V), 然后自己发送了一个自己创造的 $(U, V) = (5387871, 7127763)$ 发送 *Alice*

$$U^{-a} = U^{-55} = U^{p-1-55} = 5387871^{11881323} = 3552158 \pmod{11881379}$$
$$X = U^{-a}V = 3552158 \cdot 7127763 = 6866650 \pmod{11881379}$$

$$\begin{aligned}
&6866650 = 15 \cdot 26^4 + 12010 \\
&= 15 \cdot 26^4 + 17 \cdot 26^2 + 518 \\
&= 15 \cdot 26^4 + 0 \cdot 26^3 + 17 \cdot 26^2 + 19 \cdot 26^1 + 24 \cdot 26^0 \\
&= PARTY
\end{aligned}$$

$\square$

3 安全性问题

公钥密码系统 (如 RSA、El Gamal) 的一个主要缺点是，与现代密码 (DES、AES) 相比，它们的速度相对较慢。如果你想加密一些数据，使用公钥密码系统建立密匙。或者使用具有已建立的密钥的快速密码加密实际数据。

那么 El Gamal 密码的安全性如何保证呢? 与 RSA 类似，也是通过对一个数难以分解得出的。在离散对数问题中，也和质数一样难以分解一个大整数 N。

例 8:

让 $p = 941, \alpha = 627, x = 347, y = 781$

$$A = \alpha^x \pmod p = 627^{347} \pmod{941} = 390 \pmod{941}$$

$$B = \alpha^y \pmod p = 627^{781} \pmod{941} = 691 \pmod{941}$$

$$K = \alpha^{xy} \pmod p = 627^{347 \cdot 781} \pmod{941} = 470 \pmod{941}$$

虽然 Eve 拦截了 A, B, 但想要求出 x, y 进而求出 K 是非常困难的。或者知道 K 也很难求出 A, B。