素数测试

Prime Testing

刘卓

1 素数

为了创建一个 RSA 密钥对, 鲍勃需要选择两个大素数 p 和 q。更准确地说, 他需要一种区分质数和合数, 因为如果他知道如何做到这一点, 那么他可以选择大随机数, 直到他达到一个质数。

1.1 欧拉定理

欧拉定理 (Euler's theorem) 是数论中是一个关于同余的性质。是费马小定理的推广。

定义 *1*: *给定一个整数 n 和整数 a, 如果满足*

$$a^n \not\equiv a \pmod{n}$$

*那么我们 a 是 n 的一个**见证** $Witness$。*

而有一个 Witness 就说明 n 是一个合数。

但这个算法有一定问题：$561 = 3 \cdot 11 \cdot 17$ 是一个合数，然而 561 没有 Witness 因为 $a^{561} \equiv a \pmod{561}$。

因此我们称没有 Witness 的合数被称为卡迈克尔数，也称伪素数。并且 1984 年由 Alford、Granville 和 Pomerance 证明伪素数有无限多个。

1.2 二次剩余

二次剩余 (Quadratic Residue) 是指 a 的平方 a^2 除以 p 得到的余数。

定义 *2*: *给定一个整数 a 是二次剩余的话，那么它必须满足*

$$x^2 - a \equiv 0 \pmod{p}$$

是有解的。

例 1:

$$\text{mod}5 \quad : \quad \begin{array}{c|cccc} x & 1 & 2 & 3 & 4 \\ \hline x^2 & 1 & 4 & 4 & 1 \end{array}$$

$$\text{mod}7 \quad : \quad \begin{array}{c|cccccc} x & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline x^2 & 1 & 4 & 2 & 2 & 4 & 1 \end{array}$$

$$\text{mod} 11 \quad : \quad \begin{array}{c|cccccccccc} x & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline x^2 & 1 & 4 & 9 & 5 & 3 & 3 & 5 & 9 & 4 & 1 \end{array}$$

□

定理 1: 一个数的二次剩余正好是 $p-1$ 的一半。即有 $(p-1)/2$ 个。

证明 57 页

定理 2: 对于任何一个大于 *2* 的素数 p，并且 $a \not\equiv 0 \pmod p$, 则:

$$a^{\frac{p-1}{2}}(\text{mod} p) = \begin{cases} 1 & ifa \in QR[p] \\ -1 & ifa \notin QR[p] \end{cases}$$

证明略。

定义 3: 勒让德符号 *(Legendre symbol)*:

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & ifa \in QR[p] \\ -1 & ifa \notin QR[p] \\ 0 & ifp \mid a \end{cases}$$

其中 p 是奇素数，a 是整数。

因为从定理 2 可知，

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod p \text{和} \left(\frac{a}{p}\right) = \left(\frac{a \pmod p}{p}\right)$$

所以

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$$

定理 3: 二次互反律 *(Law of Quadratic Reciprocity)*, 如果 $p\ q$ 都是奇素数，那么:

$$\left(\frac{p}{q}\right) = (-1)^{\frac{(p-1)(q-1)}{4}}\left(\frac{q}{p}\right)$$

例 2:

$$\left(\frac{5}{71}\right) = (-1)^{\frac{(5-1)(71-1)}{4}} \cdot \left(\frac{71}{5}\right) = \left(\frac{71}{5}\right) = \left(\frac{71 \pmod 5}{5}\right) = \left(\frac{1}{5}\right) = 1$$

$$\left(\frac{3}{79}\right) = (-1)^{\frac{(3-1)(79-1)}{4}} \cdot \left(\frac{79}{3}\right) = -\left(\frac{79}{3}\right) = -\left(\frac{79 \pmod 3}{3}\right) = -\left(\frac{1}{3}\right) = -1$$

□

因此如果 $p\ q$ 都是奇素数，那么 $\frac{(p-1)(q-1)}{4}$ 是偶数，于是就可以知道

$$x^2 \equiv p \pmod q\text{有一个解} \Leftrightarrow x^2 \equiv q \pmod p\text{有一个解}$$

定义 4: 雅可比符号 (Jacobi Symbol)):

$$J(a,n) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\cdots\left(\frac{a}{p_k}\right)$$

其中 $n = p_1p_2\cdots p_k$。它是勒让德符号的延伸, 不过当 n 很大且其质因数未知时，根据这个定义计算并不容易。但是我们仍然可以通过下面的递归来计算:

$$J(a,n) = \begin{cases} 0 & if\ gcd(a,n) \neq 1 \\ 1 & if\ a = 1 \\ (-1)^{\frac{n^2-1}{8}} J\left(\frac{a}{2}, n\right) & if\ a\ is\ even \\ (-1)^{\frac{(a-1)(n-1)}{4}} J(n \pmod a, a) & if\ a\ is\ odd\ and\ a > 1 \end{cases}$$

例 3:

计算 $J(24,601)$

$$\begin{aligned} J(24,601) &= ((-1)^{(60-1)(601+1)/8)} \cdot J(\color{blue}{12}, 601) \\ &= J(\color{blue}{12}, 601) \\ &= (-1)^{((60-1)(601+1)/8)} \cdot J(\color{blue}{6}, 601) \\ &= J(\color{blue}{6}, 601) \\ &= J(\color{red}{3}, 601) \\ &= ((-1)^{(3-1)(601-1)/4)} \cdot J(\color{red}{601}, 3) \\ &= J(\color{red}{601}, 3) \\ &= J(\color{red}{601} \pmod 3, 3) \\ &= J(1,3) \\ &= 1 \end{aligned}$$

注意红色是奇数，蓝色是偶数。

```python
#雅可比符号 Jacobi Symbol 计算
def jacobi(a, n):
    assert(n > a > 0 and n%2 == 1)
    t = 1
    while a != 0:
        while a % 2 == 0:
            a /= 2
            r = n % 8
            if r == 3 or r == 5:
                t = -t
        a, n = n, a
        if a % 4 == n % 4 == 3:
```

```
            t = -t
        a %= n
    if n == 1:
        return t
    else:
        return 0

j = jacobi(24, 601)

print(j)
```

□

2 Solovay–Strassen 素数判定法则

Solovay–Strassen 素数判定法 (Solovay–Strassen Primality Test) 是一种随机算法，步骤如下:

1. 随机在区间 $[1, 2, \cdots, n-1]$ 内选择 k 个 a，即 $a_1, a_2, \cdots, a_k \in 1, 2, \cdots, n-1$;
2. 对于每一个 ai，都需要确定以下两个等式:
   - $J(a, n) = a^{(n-1)/2} \pmod n$
   - $gcd(a, n) = 1$
3. 以上两个等式如果有一个不成立，那么 n 不是素数;
4. 如果以上两个等式对于所有的 a_i 都满足，那么 n 可能为素数;

例 4:
使用 Solovay–Strassen 素数判定法则检测 $n = 8911$ 是否为素数:
随机取 5 个数: $a_1 = 10, a_2 = 20, a_3 = 30, a_4 = 40, a_5 = 50$

i	a_i	$gcd(a_i, n)$	$J(a_i, n)$	$a^{(n-1)/2} \pmod n$
1	10	1	1	0
2	20	1	1	0
3	30	1	-1	0
4	40	1	1	0
5	50	1	1	0

因为 $J(a, n) \neq a^{(n-1)/2} \pmod n$
因此 $n = 8911$ 不是素数
其时间复杂度为 $O(k \cdot log^3(n))$，k 为测试数量。

算法有可能返回错误的答案。如果输入 n 确实是素数，那么输出判断就一定正确。然而，如果输入 n 是合数，那么输出可能是不正确的素数。n 被称为 Solovay-Strassen 伪素数。

□

3 使用二次因式攻击 RSA

二次因式 (Quadratic Factoring), 对于一个因子 n, 攻击者Eve想到找两个数 x,y, 并且 $(n-1)/2 \geq x > y$, 满足下列式子:

$$x^2 - y^2 \equiv 0 \pmod n$$

如果 $x+y$ 和 $x-y$ 都不能被 n 整除, 则 $gcd(x+y,n)$ 和 $gcd(x-y,n)$ 都是 n 的非凡因子 (nontrivial factors of n), 即:

$$0 < x{-}y\ x < n{-}1, 0 < x+y\ n{-}1$$

攻击方式如下:

1. 随机在区间 $[0,\cdots,(n-1)/2]$ 内选择 k 个 x, 即 $x_1, x_2, \cdots, x_k \in [0,\cdots,(n-1)/2]$;
2. 计算所有 $x_i \pmod n$
3. 如果 $i \neq j$, 但满足 $x_i^2 \equiv x_j^2 \pmod n$
4. 则 $gcd(x_i + x_j, n) = p$ 和 $gcd(x_i - x_j, n) = q$ 都是 n 的非凡因子
5. p, q 知道以后, 就容易求出 $N = p \cdot q$, RSA 即遭到破解

4 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 5: *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 4: *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 6: *分圆多项式 (Cyclotomic polynomial):*

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

定理 5: *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 5:

$p = 11$

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

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

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

□

例 6：

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

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

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

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

例 7：

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 8：

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

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

$\square$

例 9:
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。

$\square$

定义 7: *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)。