RSA 加密算法

RSA
刘卓

1 现代密码学

现代密码学的假设:

1. 对手知道正在使用的系统;
2. 对手可以访问任意数量的相应的明文-密文对;
3. 对手可以访问加密转换 $E_k(M) = C$ 中使用的密钥;
4. 安全性是对手不能构造解密变换 $D_k(C) = M$。

定义 *1*: *如果反映射 D_k 的构造在理论上非常复杂，以至于我们现在的计算工具无法实现，那么映射 E_k 就是一个 trapdoor 函数。*

例 1

设两个因子 p,q，其中

$$\begin{aligned}
p &= 163473364580925384844313388386509085984178367003300 \\
&\quad 92312181110852389333100104508151212118167511579 \\
q &= 1900871281664822113126851573935413975471896789968 \\
&\quad 5154936666385390880271038021044989571912614655710 \\
p \cdot q &= 310741824049004372135075003588856793003734602284272754572016194882 \\
&\quad 320644051808150455634682967172328678243791627283803341547107310850 \\
&\quad 1954852900733772482278352574238645401469173660247765234660
\end{aligned}$$

尝试因式分解。 □

2 数论

定义 *2*: *Euler φ (n) 函数计算在封区间 [1,n] 内与 n 没有公因数的整数的个数。也就是说,*

$$\varphi(n) = \#\{m \in \mathbb{Z} : 1 \leq m \leq n, gcd(m,n) = 1\}$$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\varphi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	8

引理 ***1***：*如果 m,n 为正整数，使得 gcd(m,n)= 1,* 则：

$$\varphi(mn) = \varphi(m)\varphi(n)$$

例 2

$$\varphi 55) = \varphi(5 \times 11) = \varphi(5)\varphi(11) = 4 \times 10$$

□

定义 ***3*****:** 莫比斯公式：

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

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\mu(d)$	1	-1	-1	0	-1	1	-1	0	0	1	-1	0	-1	1	1

$4 = 2 \cdot 2$，有相同的质数所得，所以是 0，$12 = 2 \cdot 2 \cdot 3$ 也有相同质数所得，所以是 0。$15 = 3 \cdot 5$ 由不同质数所得，所以是 $(-1)^2 = 1$

引理 ***2***：假设 m 和 n 没有公因数。进一步假设 m 是 k 个不同素数的乘积 n 是 r 个不同素数的乘积。那么 mn 就是 $k + r$ 不同质数的乘积, 即：

$$\mu(mn) = (-1)^{k+r} = (-1)^k(-1)^r = \mu(m)\mu(n)$$

定理 ***1****:* 如果 n 是一个正整数，则可以被质因数分解：

$$n = p_1^{a_1} p_2^{a_2} \cdots p_r^{a_r}$$

进一步表示为：

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_r}\right)$$

例 3

计算 $\varphi(360)$:

$$\begin{aligned} 360 &= 2 \cdot 180 \\ &= 2 \cdot 2 \cdot 90 \\ &= 2 \cdot 2 \cdot 2 \cdot 45 \\ &= 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 5 \\ &= 2^3 \cdot 3^2 \cdot 5 \end{aligned}$$

$$
\begin{aligned}
\varphi(360) &= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_r}\right) \\
&= 360\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{5}\right) \\
&= 96
\end{aligned}
$$

```python
# 计算欧拉公式值
def gcd(p,q):
    while q != 0:
        p, q = q, p%q
    return p

def is_coprime(x, y):
    return gcd(x, y) == 1

def phi(x):
    if x == 1:
        return 1
    else:
        n = [y for y in range(1,x) if is_coprime(x,y)]
        return len(n)
print(phi(360))
```

$\square$

那么 $\varphi(n)$ 和 $\mu(d)$ 有什么关系呢?

$$\varphi(n) = \sum_{d|n} \mu(d)\frac{n}{d}$$

例 4

让 $n = p_1^{a_1}p_2^{a_2}$

则 $p_1^i p_2^j$, $0 \le i \le a_1$,$0 \le j \le a_2$

- $d = p_1^0 p_2^0 = 1 \Rightarrow \mu(1) = (-1)^0 = 1$
- $d = p_1^1 p_2^0 = p_1 \Rightarrow \mu(p_1) = (-1)^1 = -1$
- $d = p_1^0 p_2^1 = p_2 \Rightarrow \mu(p_2) = (-1)^1 = -1$
- $d = p_1^0 p_2^1 = p_1p_2 \Rightarrow \mu(p_1p_2) = (-1)^2 = 1$

$$\varphi(n) = \mu(1)\cdot\frac{n}{1} + \mu(p_1)\cdot\frac{n}{p_1} + \mu(p_2)\cdot\frac{n}{p_2} + \mu(p_1p_2)\cdot\frac{n}{p_1p_2}$$

$$
\begin{aligned}
&= n - \frac{n}{p_1} - \frac{n}{p_2} + \frac{n}{p_1 p_2} \\
&= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)
\end{aligned}
$$

□

模运算的两个基本结果:

1. **定理** *2(***费马** *Fermat): 如果 p 是一个素数 a 是一个不能被 p 整除的整数，那么*

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

2. **定理** *3(***欧拉** *Euler): 如果 a 和 n 是相对素数的非零整数，那么*

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

例 5

$$115 = 5 \cdot 23 \Rightarrow \varphi(115) = \varphi(5)\varphi(23) = 4 \cdot 22 = 88$$

$$\underbrace{12659}_{a}{}^{\overbrace{88}^{\varphi(n)}} \equiv 1 \quad mod \quad \underbrace{115}_{n}$$

□

3 RSA

RSA 加密算法是一种非对称加密算法，在公开密钥加密和电子商业中被广泛使用。RSA 是由罗纳德 · 李维斯特（Ron Rivest）、阿迪 · 萨莫尔（Adi Shamir）和伦纳德 · 阿德曼（Leonard Adleman）在 1977 年一起提出的。当时他们三人都在麻省理工学院工作。RSA 就是他们三人姓氏开头字母拼在一起组成的。

它是基于这样一个假设: 两个素数相乘很容易，但将两个素数的乘积分解成质数分量却很难。到目前为止，世界上还没有任何可靠的攻击 RSA 算法的方式。只要其密钥的长度足够长，用 RSA 加密的信息实际上是不能被破解的。

3.1 RSA 使用步骤

1. (密钥创建，私钥) *Bob* 选择两个只有自己执导的质数 p 和 q，然后计算 $N = p \cdot q$;
2. *Bob* 选择一个正整数素数的加密指数 (encryption exponent)e，计算 $\varphi(N) = (p-1)(q-1)$;
3. *Bob* 公开 N 和 e;
4. (加密) *Alice* 将明文 m, 使用 *Bob* 公开的密钥加密。密文计算方法为 $c \equiv m^e \pmod{N}$;

5. *Alice* 发送密文 c 给 *Bob*;

6. (解密)*Bob* 解开密文 c，一共两步：

   (a) $d \equiv e^{-1}(\mathrm{mod} \quad \varphi(N)) \equiv e^{-1}(\mathrm{mod} \quad (p-1)(q-1))$

   (b) $m \equiv c^d(\mathrm{mod} \quad N)$

3.2 RSA 安全性

假设 *Eve* 已经知道公钥 N, e 和密文 c，想要得到 d。那么 *Eve* 需要尝试解方程

$$d \cdot e \equiv 1 \pmod{\varphi(N)}$$

但是仅从 N 得到 $\varphi(N)$ 等价于分解 N，这是非常困难的，因为：

$$\varphi(N) = (p-1)(q-1) = pq - p - q + = N - (p+q) + 1$$

为了得到到 $\varphi(N)$, 但有两件事困扰着你：

1. 将 N 分成两个质数 p, q 从目前计算机算力角度来说是非常困难的；

2. p, q 的和从未公开

虽然难以破解，但假如 N 的长度小于或等于 256 位，那么用一台个人电脑在几个小时内就可以暴力因数分解它的因子了。但当 N 很大时，破解难度和时间就会飞增。现在加密系统中，一般采取 2048 位长度的 N 来确保安全性

3.3 RSA 加密方法

为了发送消息，即一段字符串，我们需要将将消息转换为数字，反之亦然。

基本思想是使用字母数量等于 N 以 26 为基数的对数。为了得到的任何 k 位数字 (以 26 为底) 都必须小于 N，所以我们必须有

$$N \geq 26^k - 1 \Rightarrow k = \log_2 6(N)$$

k 为整数，向下取整。这里的 k 就是一次性能发送多少位字符的数量。比如说 $N = 131 \cdot 1873 = 245363 \Rightarrow k = \log_2 6(245363) = 3$，也就是说使用质数 $p = 131, q = 1873$ 时，一次可以加密 3 个字符的信息发送给对方。

例 6

令明文 $m = \mathbf{THE}$

那么 $m = \mathbf{T} \cdot 26^0 + \mathbf{H} \cdot 26^1 + \mathbf{E} \cdot 26^2 = 19 + 7 \cdot 26 + 4 \cdot 26^2 = 2905$

又已知 $e = 323, N = 245363$，所以 $2905^323 \equiv 13388 \pmod{245363}$

那么如果是以拉丁字母表 26 个字母为基底的话，加密过程如下：

$$\begin{aligned}
13388 &= 24 + 514 \cdot 26 \\
&= 24 + (20 + 19 \cdot 26) \cdot 26 \\
&= 24 \cdot 1 + 20 \cdot 26 + 19 \cdot 26^2 \\
&= \mathbf{Y} \cdot 26^0 + \mathbf{U} \cdot 26^1 + \mathbf{T} \cdot 26^2
\end{aligned}$$

所以明文 **THE** 通过 RSA 可以加密为 **YUT**

□

3.4 幂取模运算 Power in modulo arithmetic

定理 4： 如果 a,n,x,y 是非负整数并且满足 $gcd(a,n)=1$，则：

$$x \equiv y \pmod{\varphi(n) \Rightarrow a^x \equiv a^y \pmod{n}}$$

证明：

$$\begin{aligned} x &\equiv y \pmod{\varphi(n)} \\ &\equiv y + k \cdot \varphi(n) \\ &\Rightarrow a^x = a^{y+k\cdot\varphi(n)} \\ &\Rightarrow a^x = a^y + a^{k\cdot\varphi(n)} \\ &\Rightarrow a^x = a^y + a^{\varphi(n)k} \\ &\Rightarrow a^x = a^y \pmod{n} \end{aligned}$$

例 7

$$7^4 2 \pmod{11}$$

$$42 \equiv 2(\bmod \quad \varphi(11)) = 2(\bmod \quad 10)$$

$$7^4 2 \equiv 7^2(\bmod \quad 11) = 49(\bmod \quad 11) = 5$$

□

3.5 计算大数的余

例 8

如果计算 $1915793^{2641} \pmod{5678923}$ 这种大数的 mod 呢？我们不想处理大于 n^2 的数字。应用定理 4.

第一步：

a = 1915793,k = 5678923,n = 5678923

将 k 转化为二进制数，

$$k = 5678923 = 1 + 2^4 + 2^6 + 2^9 + 2^1 1 = 1 + 16 + 64 + 512 + 2048$$

```
#二进制数
number = 2641
number_bin = bin(number)[2:]

cov_number_bin = bin(number)[2:][::-1]

sum_number = ''

for i in range(len(cov_number_bin)):
    if cov_number_bin[i] == '1':
        # print(i)
        if sum_number == '':
            sum_number += str(2**(i))
        else:
            sum_number += ' + ' + str(2**(i))


print(sum_number)
```

第二步:

用重复的平方来得到上面的幂的余:

$a^2 \equiv 3278564 \pmod{n}$,得到 3278564,就可以继续使用作为 a^4 的基本单位,不用计算 a^4,只需要计算 $a^4 \pmod{n} = (a^2)^2 \pmod{n} = 3278564^2 \pmod{n} \equiv 1631541 \pmod{n}$。得到 1631541 后又可以继续应用在 a^8 上,使得 $a^8 \pmod{n} = (a^4)^2 \pmod{n} = 1631541^2 \pmod{n} \equiv 4704430 \pmod{n}$。以减少运算量,以此列推,得到所有 a 的 2 的幂次项的余。

$$
\begin{array}{lll}
a^2 \equiv 3278564 \pmod{n} & a^4 \equiv 1631541 \pmod{n} & a^8 \equiv 4704430 \pmod{n} \\
a^{16} \equiv 1424066 \pmod{n} & a^{32} \equiv 3532287 \pmod{n} & a^{64} \equiv 3305529 \pmod{n} \\
a^{128} \equiv 1529537 \pmod{n} & a^{256} \equiv 5673135 \pmod{n} & a^{512} \equiv 5106329 \pmod{n} \\
a^{1024} \equiv 2627277 \pmod{n} & a^{2048} \equiv 1180227 \pmod{n} & \cdots
\end{array}
$$

第三步:

$$
\begin{aligned}
a^k &= a \cdot a^{16} \cdot a^{64} \cdot a^{512} \cdot a^{2048} \\
&= 1915793 \cdot 1424066 \cdot 3305529 \cdot 5106329 \cdot 1180227 \pmod{n} \\
&= 1162684 \pmod{n}
\end{aligned}
$$

□

3.6 完整的一次 RSA 加密和解密过程

例 9

选择 $p=53, q=89$, 已知加密因子和解密因子 $e=119, d=424$
首先检查满足 $e \cdot d \equiv 1 \pmod{\varphi(4717)} = 1 \pmod{(53-1)(89-1)} = 1 \pmod{4576}$

1. 加密明文 75
第一步:

$$N = p \times q = 53 \cdot 89 = 4717$$

第二步:

$$c = m^e \pmod{N} = 75^{119} \pmod{4717} = 3500$$

第三步:

$$119 = 2^0 + 2^1 + 2^2 + 2^4 + 2^5 + 2^6 = 1 + 2 + 4 + 16 + 32 + 64$$

第四步:

k	1	2	4	8	16	32	64
$75^k \pmod{N}$	75	908	3706	3249	4072	929	4547

$$75^{119} = 75 \cdot 908 \cdot 3706 \cdot 3249 \cdot 4072 \cdot 929 \cdot 4547 \pmod{4717} = 3530$$

2. 解密密文 2

$$m = c^d \pmod{N} = 2^{423} \pmod{4717} = 3414$$

$\square$