SHANDONG UNIVERSITY

密码学引论作业 5

谢子洋 202100460116

2023 年 4 月 16 日

1 正确性证明

设已知 RSA 的各项参数:

明文为 m, 加密密钥为 e, 解密密钥为 d, 选取两大素数为 p 和 q, 大素数乘积为 n.

$$\begin{aligned}
Dec_d(Enc_e(m)) &= m^{ed} \ mod\ n \\
&= m^{1+k\frac{(p-1)(q-1)}{gcd(p-1,q-1)}} \ mod\ n \\
&= m(m^{\frac{\phi(n)}{gcd(p-1,q-1)}})^k \ mod\ n \\
&= m(m^{lcm(p-1,q-1)})^k \ mod\ n
\end{aligned}$$

引理:

设 $m = 2^{\alpha_0} p_1^{\alpha_1} \dots p_r^{\alpha_r}$, 其中 p_i 为两两不同奇素数, 则

$$\delta_m(a) | \lambda(m) \text{ , 其中 } \lambda(m) = [2^{c_0}, \varphi(p_1^{\alpha_1}), \dots, \varphi(p_r^{\alpha_r})],\ c_0 = \begin{cases} 0 & \alpha_0 = 0 \ or \ 1 \\ 1 & \alpha_0 = 2 \\ \alpha_0 - 2 & \alpha_0 \geq 3 \end{cases}$$

因为 n=pq, 所以 $\delta_n(m)|\lambda(n)$ 其中 $\lambda(n) = [\varphi(p), \varphi(q)] = lcm(p-1, q-1)$

所以对任意 m, 都有

$$m^{lcm(p-1,q-1)} = 1 \ (mod\ n)$$

成立.

因此上式变为:

$$\begin{aligned}
Dec_d(Enc_e(m)) &= m(m^{lcm(p-1,q-1)})^k \ mod\ n \\
&= m \cdot 1 \ mod\ n \\
&= m
\end{aligned}$$

2 CRT 加速 RSA 解密过程

2.1 算法

密钥生成阶段:

$$dp = e^{-1} mod\,(p-1)$$

$$dq = e^{-1} mod\,(q-1)$$

$$qInv = q^{-1} mod\ p$$

解密阶段:

$$m1 = c^{dp} mod\ p$$

$$m2 = c^{dq} mod\ q$$

$$h = qInv * \ ((m1-m2) mod\ p)) mod p$$

$$m = m2 + hq$$

2.2 正确性

因为

$$m1 = m\ mod\ p = c^d\ mod\ p = c^{d\ mod \varphi(p)}\ mod\ p$$

$$m2 = m\ mod\ q = c^d\ mod\ q = c^{d\ mod \varphi(q)}\ mod\ q$$

因此有同余方程组

$$\begin{cases} m = m1\ mod\ p \\ m = m2\ mod\ q \end{cases}$$

利用中国剩余定理 CRT 计算该同余方程组, 得到结果:

$$\begin{aligned} m &= (q \cdot qInv \cdot m_1 + p \cdot pInv \cdot m_2)\ mod\ n \\ &= m_2 + q \cdot (qInv \cdot ((m_1 - m_2) mod\ p))\ mod p) \\ &= m \end{aligned}$$

2.3 实验过程与代码实现

利用 CRT 加速 RSA 解密需要在参数生成时计算额外参数, 且计算中要用到大素数 p 和 q. 而本题中仅给出 n, 难以直接使用 CRT 加速解密. 本文使用分解 n 的 Las Vegas 算法, 根据密钥 e、d 分解出 p 和 q.

```python
#python语言实现分解n的Las Vegas算法
def factorN(n,e,d):
    r=e*d-1
    s=0
```

```
    while(r%2==0):
        r=r>>1
        s+=1
    while(True):
        b=random.randint(2,n)
        if(gmpy2.gcd(b,n)>1):#随机选择成功
            print("sucess")
            x=gmpy2.gcd( b+1,n)
            break
        a=gmpy2.powmod(b,r,n)
        if(a==1):pass#失败
        tempStore=a
        while(a!=1):
            tempStore=a
            a=gmpy2.powmod(a,2,n)
        #此时a=1
        if(tempStore==n-1):pass
        else:
            print("sucess")
            x= gmpy2.gcd(tempStore+1,n)
            break
    p=x
    q=n//p
    return p,q
```

已知大素数 p 和 q, 可求出 CRT 加速 RSA 解密所需参数 dp、dq、qInv,

$$dp = e^{-1} mod\,(p-1)$$

$$dq = e^{-1} mod\,(q-1)$$

$$qInv = q^{-1} mod\ p$$

其后可对密文 c 进行 CRT 加速解密.

```
class Alice():
  def __init__(self):
      self.__q=0
      self.__p=0
      self.__n=0
      self.__phi_n=0
      self.__e=0
      self.__d=0
      self.__dp=0
      self.__dq=0
      self.__qInv=0
  def setPara(self,p,q,e,d):
      self.__q=q
      self.__p=p
      self.__n=p*q
      self.__phi_n=(p-1)*(q-1)
      self.__e=e
      self.__d=d
      self.__dp=gmpy2.invert(self.__e,(self.__p-1))
      self.__dq=gmpy2.invert(self.__e,(self.__q-1))
      self.__qInv=gmpy2.invert(self.__q,self.__p)
```

```
    #向外界发送公钥
    def getPublicKey(self):
        return [ self.__n , self.__e]

    #Alice CRT解密
    def decryptByCRT(self,c):
        m1=gmpy2.powmod(c,self.__dp,self.__p)
        m2=gmpy2.powmod(c,self.__dq,self.__q)
        temp=gmpy2.mod((m1-m2),self.__p)
        h=gmpy2.mod((self.__qInv*temp),self.__p)
        m=m2+h*self.__q
        return m
```

最后同时调用两部分代码实现 CRT 加速解密.

```
def decrypt():
  n=3026533
  e=3
  d=2015347
  c=152702
  #1. 计算pq
  p,q=factorN(n,e,d)
  #2. 已知pq进行CRT加速解密
  A=Alice()
  A.setPara(p,q,e,d)
  m=A.decryptByCRT(c)
  print(m)
decrypt()
```

2.4 实验结果

参数	n	p	q	c	m
数值	3026533	2003	1511	152702	1186745