

# 密码学实验实验报告八

18374480-黄翔

2021 年 5 月 23 日

## 1 实验目的

1. 掌握 RSA 算法原理及实现
2. 了解常见的 RSA 攻击方法

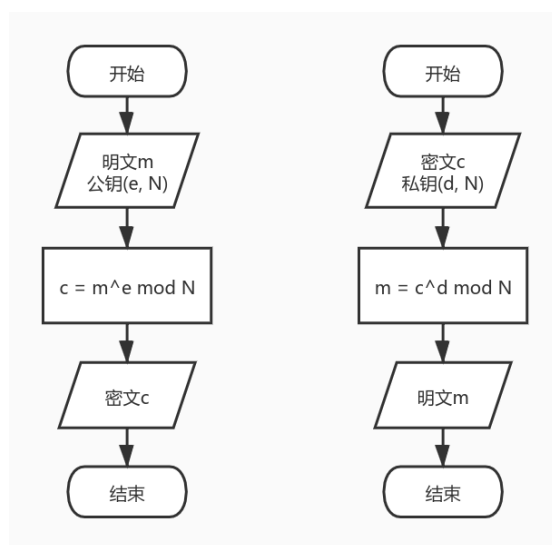
## 2 实验环境

1. python 3.9

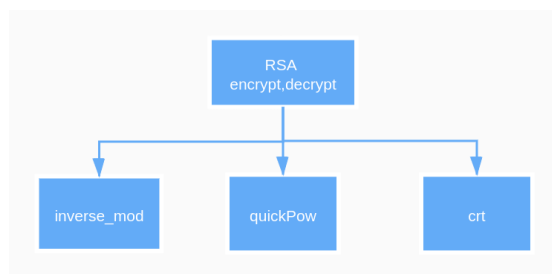
### 3 实验内容

#### 3.1 RSA 加解密

##### 3.1.1 算法流程图



##### 3.1.2 函数调用关系



##### 3.1.3 算法伪代码

---

**Algorithm 1** RSA 加密

---

**Input:** 明文  $m$ , 公钥对  $(e, N)$

**Output:** 密文  $c$

1:  $c \leftarrow m^e \bmod N$

2: **return**  $c$

---

---

**Algorithm 2** RSA 解密

---

**Input:** 密文  $c$ , 私钥对  $(d, N)$ **Output:** 明文  $m$ 1:  $m \leftarrow c^d \bmod N$ 2: **return**  $m$ 

---

---

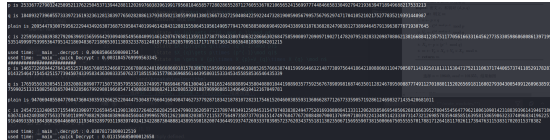
**Algorithm 3** RSA 快速解密

---

**Input:** 密文  $c$ , 私钥对  $(d, p, q)$ **Output:** 明文  $m$ 1:  $n \leftarrow p \times q$ 2:  $V_p \leftarrow C^{d \bmod p-1} \bmod p$ 3:  $V_q \leftarrow C^{d \bmod q-1} \bmod q$ 4:  $X_p \leftarrow q \times (q^{-1} \bmod p)$ 5:  $X_q \leftarrow p \times (p^{-1} \bmod q)$ 6:  $m \leftarrow V_p \times X_p + V_q \times X_q \bmod n$ 7: **return**  $m$ 

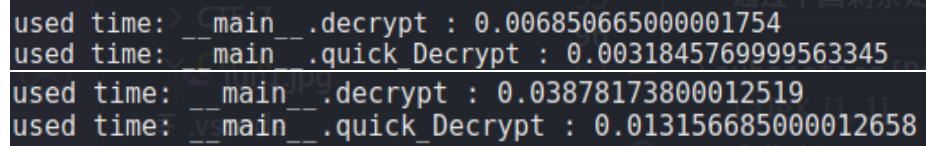
---

### 3.1.4 测试样例及结果截图



### 3.1.5 总结

**CRT 加速** 通过中国剩余定理, 将模  $N$  的运算降低到模  $p, q$ , 同时指数  $d$  降低到  $d \bmod p-1, d \bmod q-1$ 。理论上, 利用 CRT 解密速度能快上 4 倍

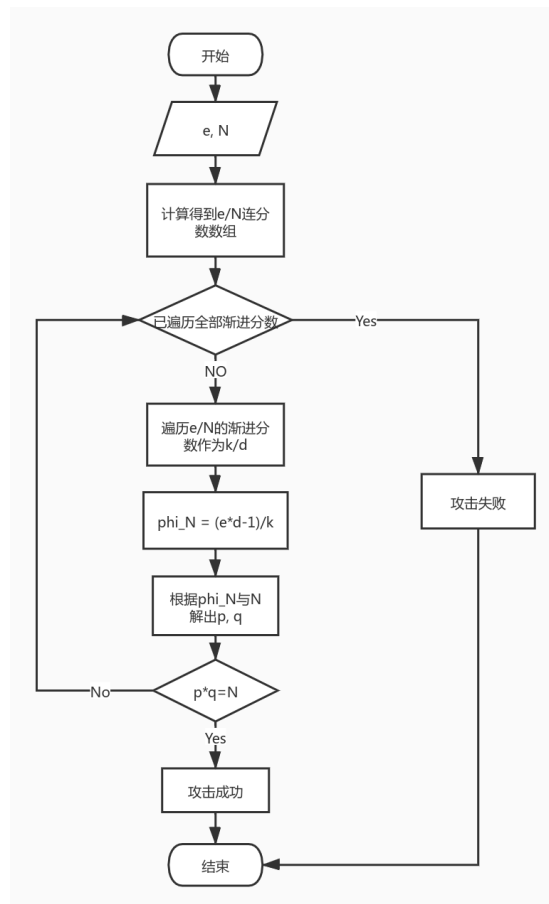


**参数选择** RSA 算法安全性与其选择的参数息息相关, 具体见思考题。

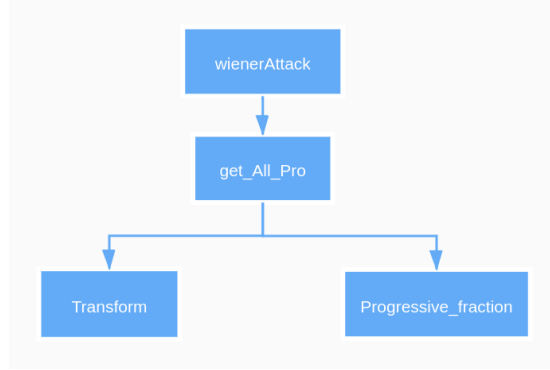
**RSA 安全性** RSA 是不可证明安全的算法，其安全性不高于大数分解的安全性，但是目前实际使用中其对攻击的良好抵抗效果体现了其安全性。

## 3.2 Wiener's Attack

### 3.2.1 算法流程图



### 3.2.2 函数调用关系



### 3.2.3 算法伪代码

---

**Algorithm 4** Wiener Attack

---

```
1: function Transform(up, down)
2:   //此函数用于获得  $\frac{up}{down}$  全部连分数
3:   list  $\leftarrow$  [ ]
4:   while down do
5:     list.append(up//down)
6:     up  $\leftarrow$  down
7:     down  $\leftarrow$  up mod down
8:   end while
9:   return list
10: end function
11: function Progressive_fraction(subList)
12:   //此函数用于对连分数列表求渐进分数
13:   up  $\leftarrow$  1, down  $\leftarrow$  0
14:   subList  $\leftarrow$  inverse(subList)
15:   for frac in subList do
16:     down  $\leftarrow$  up
17:     up  $\leftarrow$  frac  $\times$  up + down
18:   end for
19:   return down, up
20: end function
```

```

21: function getAllPro(up, down)
22:   //此函数用于获得  $\frac{up}{down}$  所有渐进分数
23:   list  $\leftarrow$  Transform(up, down)
24:   for i  $\leftarrow$  1 to len(list) do
25:     yield Progressive__fraction(list[0 : i])
26:   end for
27: end function
28: function Wiener__Attack(e, N)
29:   for (d, k) in getAllPro(e, N) do
30:     if k is 0 or (e  $\times$  d - 1)  $\not\equiv$  0 mod k then
31:       Continue
32:     end if
33:      $\varphi(N) \leftarrow \frac{e \times d - 1}{k}$ 
34:     p, q  $\leftarrow$  Solve( $\varphi(N) = N - (p + q) + 1$ )
35:     if p  $\times$  q is N then
36:       Attacked!
37:       return d, p, q
38:     end if
39:   end for
40:   Failed
41: end function

```

---

#### 3.2.4 测试样例及结果截图

```

Attacked!!!
used time: MyCrypto.RSA.rsa.quick.Decrypt : 0.0017832879993875395
flag{although_I_am_very_vegetable_but_I_have_delicious_or4nges}

```

#### 3.2.5 总结

攻击条件  $d$  比较小时 ( $d < \frac{1}{3}N^{\frac{1}{4}}$ )

攻击原理

$$\begin{aligned}
 ed &\equiv 1 \pmod{\varphi(N)} \\
 ed - 1 &= k \times \varphi(N) \\
 \frac{e}{\varphi(N)} - \frac{k}{d} &= \frac{1}{d \times \varphi(N)}
 \end{aligned}$$

$$\frac{e}{N} - \frac{k}{d} = \frac{1}{d \times \varphi(N)}$$

通过  $\frac{e}{N}$  的渐进分数既可逼近得到  $\frac{k}{d}$

## 4 总结

通过本次实验, 我掌握 RSA 算法原理及实现, 了解常见的 RSA 攻击方法。进一步体会到了 RSA 公钥密码体制与对称密码的区别, 体会到了非对称密码中陷门的作用。通过对攻击方法的学习, 体会到了密码算法在实际应用中的安全很大程度取决于使用者的参数选择。

## 5 思考题

1. n 的选择
  - (a) p, q 之差要大
  - (b) p, q 本身要大
  - (c) p-1, q-1 的最大公因子要尽可能小
  - (d) p, q 必须为强素数
2. e 的选择
  - (a) e 不能太小, 防止低指数攻击
  - (b) 应使 e 对  $\varphi(n)$  的阶为最大
3. d 的选择: d 不能太小, 一般大于  $N^{\frac{1}{4}}$