

# 密码学实验实验报告三

18374480-黄翔

2021 年 3 月 26 日

## 1 实验目的

1. 通过本次实验，了解古典加密算法思想，掌握常见的古典密码
2. 学会应用古典密码以及针对部分古典密码的破译

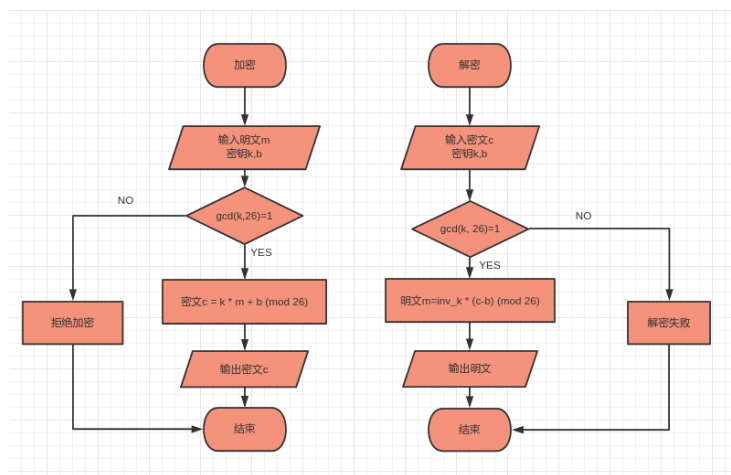
## 2 实验环境

- python 3.9.1+
- python 2.7.18+ for Varnam Cipher

## 3 实验内容

### 3.1 仿射密码

#### 3.1.1 算法流程



#### 3.1.2 算法伪代码

---

##### Algorithm 1 仿射密码加密

---

**Input:**  $m, k, b$

**Output:**  $c$

- 1: **if**  $\gcd(k, 26)$  is 1 **then**
  - 2:      $c \leftarrow k \times m + b \pmod{26}$
  - 3:     **return**  $c$
  - 4: **else**
  - 5:     拒绝加密
  - 6: **end if**
- 

---

##### Algorithm 2 仿射密码解密

---

**Input:**  $c, k, b, \text{Function } \text{inverse\_mod}$

**Output:**  $m$

- 1: **if**  $\gcd(k, 26)$  is 1 **then**
- 2:      $\text{inv}_k = \text{inverse\_mod}(k, 26)$
- 3:      $m \leftarrow \text{inv}_k \times (c - b) \pmod{26}$

```

4:   return  $m$ 
5: else
6:   解密失败,  $k$  不符合要求
7: end if

```

---

### 3.1.3 测试样例及结果截图

```

the cypher of cryptography is yzwlneazklhw      密文: yzwlneazklhw
the cypher of seeyoutomorrow is txxvjlcjrjkkjd
the plaintexe of thisisciphertext is tnumurnsftsvt b=
Warning, check you k
the cypher of abcdef is bdfhjl                    明文: bdfhjl

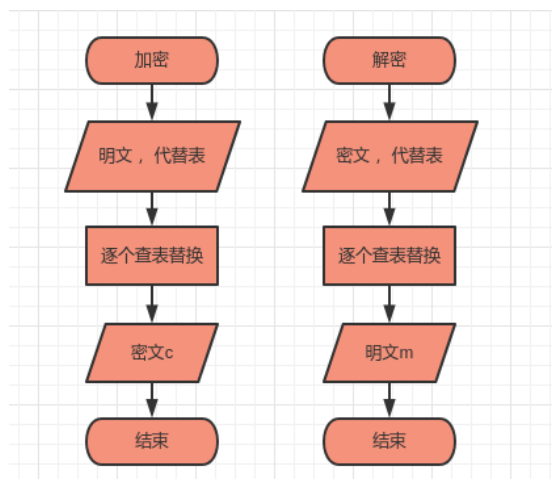
```

### 3.1.4 总结

只有当  $\gcd(k, 26) = 1$  时, 才能使加密过程为一一映射, 从而确保能够成功解密

## 3.2 单表代替密码

### 3.2.1 算法流程



### 3.2.2 算法伪代码

---

**Algorithm 3** 单表代替密码加密

---

**Input:**  $m, table$

**Output:**  $c$

---

```

1: for letterchinm do
2:   ch = table.cypher[ch]
3:   c+ = ch
4: end for
5: return c

```

---



---

#### Algorithm 4 单表代替密码解密

---

**Input:** *c, table*

**Output:** *m*

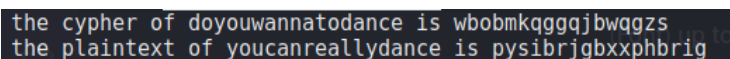
```

1: for letterchinc do
2:   ch = table.plain[ch]
3:   m+ = ch
4: end for
5: return m

```

---

#### 3.2.3 测试样例及结果截图



```

the cypher of doyouwannatodance is wbobmkqggqjbbwqgz
the plaintext of youcanreallydance is pysibrjgbxxphbrig

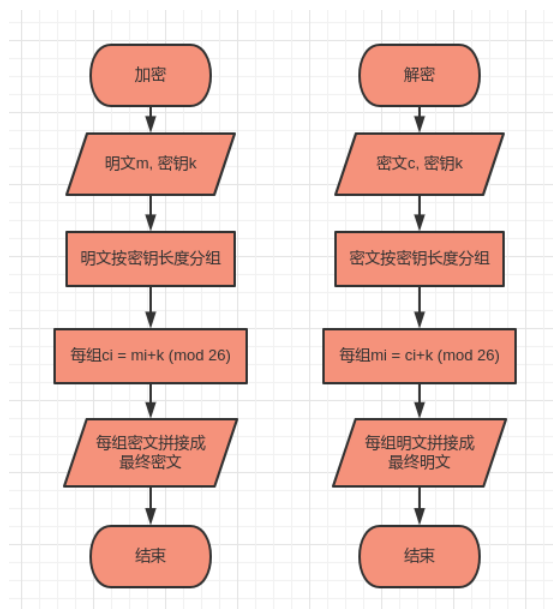
```

#### 3.2.4 总结

**密码安全性** 与仿射密码相比，单表代替密码的可能情况有 26 种，足够抵抗穷举攻击，但是由于字母一一对应，仍然无法抵抗词频攻击

### 3.3 维吉尼亚密码

#### 3.3.1 算法流程



#### 3.3.2 算法伪代码

---

**Algorithm 5** 维吉尼亚密码加密

---

**Input:**  $m, k$

**Output:**  $c$

```
1: for  $i \leftarrow 0$  to  $length(m) - 1$  do
2:    $c += (m[i] + k[i \% length(k)])$ 
3: end for
4: return  $c$ 
```

---

---

**Algorithm 6** 维吉尼亚密码解密解密

---

**Input:**  $c, k$

**Output:**  $m$

```
1: for  $i \leftarrow 0$  to  $length(c) - 1$  do
2:    $m += (c[i] - k[i \% length(k)])$ 
3: end for
4: return  $m$ 
```

---

### 3.3.3 测试样例及结果截图

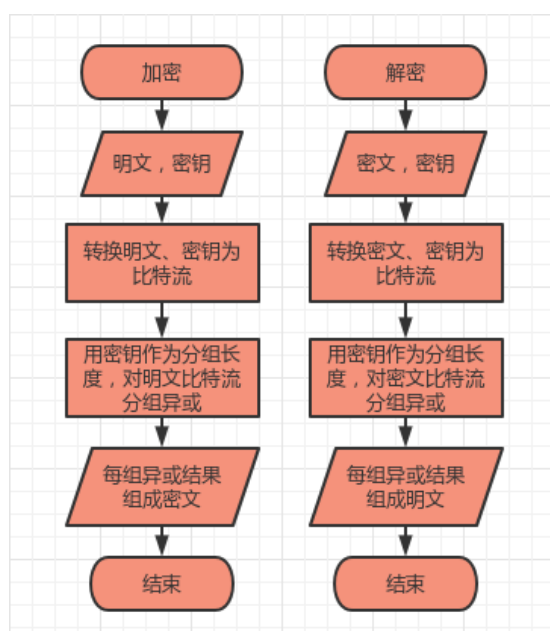
```
the cypher of zhonghuaminzuweidafuxing is huhrxltuvthhpizhscopyvt
the plaintext of kjyhyrnruwnadzm is jcsqsmddoauclyc
```

### 3.3.4 总结

**密码安全性** 维吉尼亚密码强度在于每个明文字母对应着多个密文字母，且每个密文字母使用唯一的密钥字母，因此字母出现的频率信息被隐藏了，不过并非所有的明文结构都隐藏了，依然可以使用频率分析攻击

## 3.4 弗纳姆密码

### 3.4.1 算法流程



### 3.4.2 算法伪代码

---

**Algorithm 7** 弗纳姆密码加密

---

**Input:**  $m, k$

**Output:**  $c$

- 1:  $m \rightarrow$  binary bit stream  $bit\_m$
- 2:  $k \rightarrow$  binary bit stream  $bit\_k$

```

3: for  $i \leftarrow 0$  to  $length(bit\_m)$  do
4:    $c+ = bit\_m[i] \oplus bit\_k[i \% length(bit\_k)]$ 
5: end for
6: return  $c$ 

```

---

#### Algorithm 8 弗纳姆密码解密

---

**Input:**  $c, k$

**Output:**  $m$

```

1:  $c \rightarrow$  binary bit stream  $bit\_c$ 
2:  $k \rightarrow$  binary bit stream  $bit\_k$ 
3: for  $i \leftarrow 0$  to  $length(bit\_m)$  do
4:    $m+ = bit\_c[i] \oplus bit\_k[i \% length(bit\_k)]$ 
5: end for
6: return  $m$ 

```

---

#### 3.4.3 测试样例及结果截图

```

$ cat -v output1.txt && echo '\n' && cat -v output2.txt
% ^X^A^S^M^P^F[ _BQCWV ^
abcdefghijklmnopqrstuvwxyz

```

#### 3.4.4 总结

**编程相关** 注意到给定文件形式为 binary file, python 在读文件时需要使用 rb 形式, 否则输出会出现错误 (linux 系统会出现此问题, windows 下则不会出现)

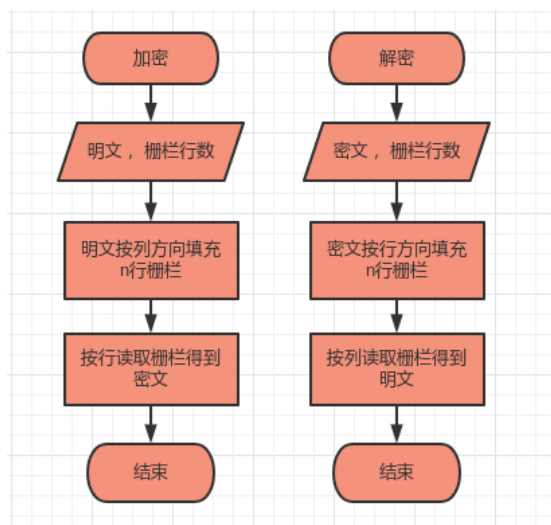
读取方式	结果 (python2)
'r'	abcdefghijkl
'rb'	abcdefghijkl

表 1: 不同读文件方式结果

**一次一密** 使用与明文一样长且无重复的随机密钥加密, 且密钥  $i$  只对一个明文加解密, 之后丢弃不用。每个新消息需要一个与其等长的新密钥

## 3.5 栅栏密码

### 3.5.1 算法流程



### 3.5.2 测试样例及结果截图

```
the cypher of whateverisworthdoingisworthdoingwell is wtesrdnsrdneherwtogwtoglaviohihiwl  
the plaintext of hatimriprathnelhelsoemotntawat is healthismoreimportantthanwealth
```

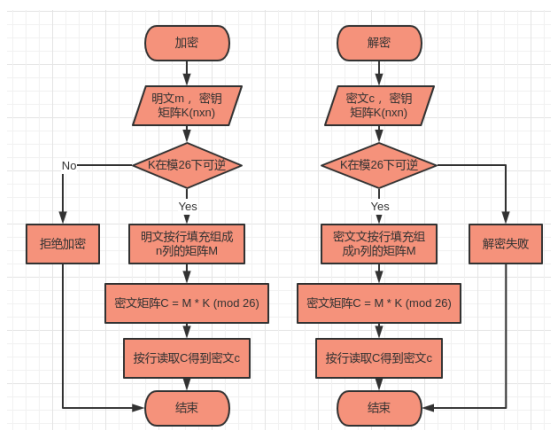
### 3.5.3 总结

栅栏密码当明文无法完全填充栅栏时，可以用一些标志字符补全。

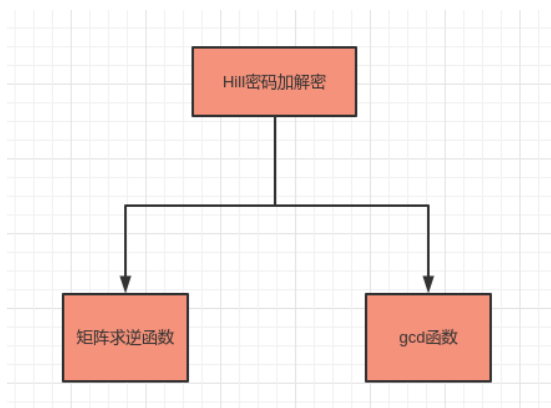


## 3.6 Hill 密码

### 3.6.1 算法流程



### 3.6.2 函数调用图



### 3.6.3 算法伪代码

---

#### Algorithm 9 Hill 密码加密

---

**Input:**  $m, K$ , Function  $gcd$

**Output:**  $c$

- 1: **if**  $gcd(det(K), 26) \neq 1$  **then**
- 2:     refuse to encrypt
- 3: **else**
- 4:      $m \rightarrow \text{Matrix } M$

```

5:    $C \leftarrow M \times K$ 
6:    $C \rightarrow c$ 
7:   return  $c$ 
8: end if

```

---



---

**Algorithm 10** Hill 密码解密

---

**Input:**  $c, K$ , Function  $gcd, Matrix\_inv$

**Output:**  $m$

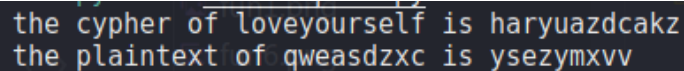
```

1: if  $gcd(det(K), 26) \neq 1$  then
2:   for  $k \leftarrow 0$  to 25 do
3:      $m = D(c, k)$ 
4:     caculate Frequence Table  $DT$ 
5:     for  $chin \{a, b, \dots, z\}$  do
6:        $\chi + = \frac{(DT[ch] - LT[ch])^2}{LT[ch]}$ 
7:     end for
8:      $Table[k] = \chi$ 
9:   end for
10:  return  $Table$ 
11:  fail to decrypt
12: else
13:   $c \rightarrow$ Matrix  $C$ 
14:   $M \leftarrow C \times Matrix\_inv(K)$ 
15:   $M \rightarrow m$ 
16:  return  $m$ 
17: end if

```

---

### 3.6.4 测试样例及结果截图



```

the cypher of loveyourself is haryuazdcakz
the plaintext of qweasdzxc is ysezymxvv

```

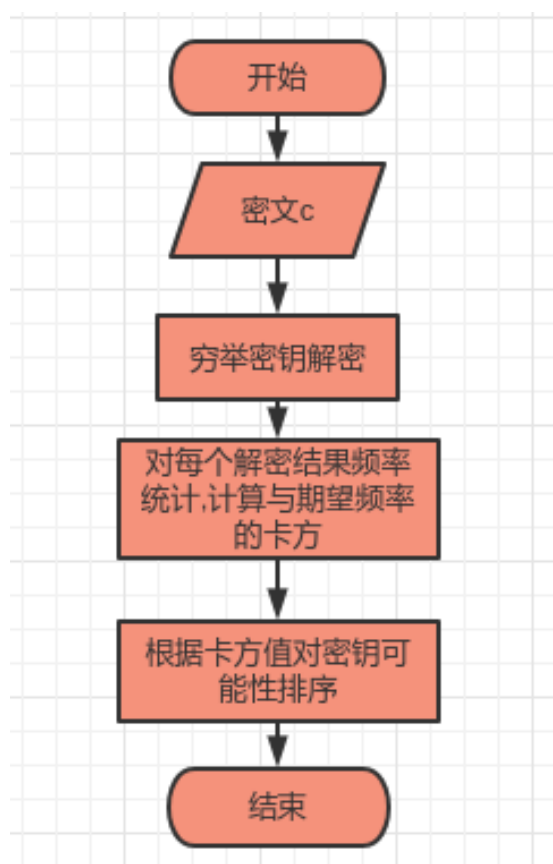
### 3.6.5 总结

**密钥矩阵要求** 密钥矩阵  $K$  需要在模 26 下可逆, 即  $gcd(det(K), 26) = 1$ , 从而保证密钥矩阵  $K$  可逆

**密码安全性** Hill 密码完全隐蔽了单字母频率特性。Hill 用的矩阵越大，隐藏的频率信息越多， $3 \times 3$  的 Hill 密码不仅隐藏了单字母的频率特性，还隐藏了双字母的频率特性。Hill 密码足以抵抗唯密文攻击，但是较易被已知明文攻击破解。

### 3.7 加法密码字母频率攻击

#### 3.7.1 算法流程



#### 3.7.2 算法伪代码

---

**Algorithm 11** 加法密码字母频率攻击

---

**Input:**  $c$ , Decrypt Function  $D$ , Letter Frequency Table  $LT$

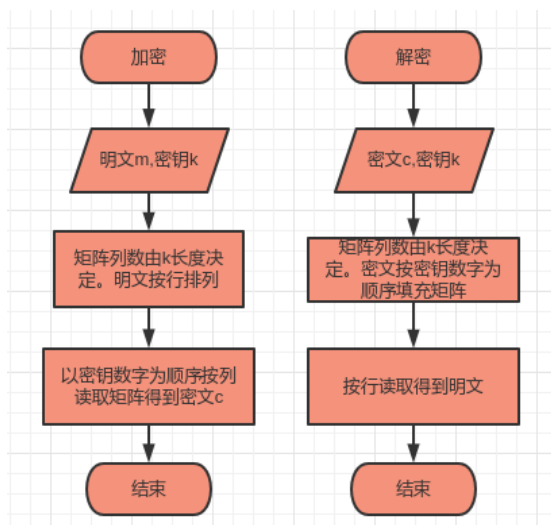
**Output:**  $Table$ : the  $k$  and its  $\chi$

1: **for**  $k \leftarrow 0$  to 25 **do**



## 3.8 矩阵密码

### 3.8.1 算法流程



### 3.8.2 测试样例及结果截图

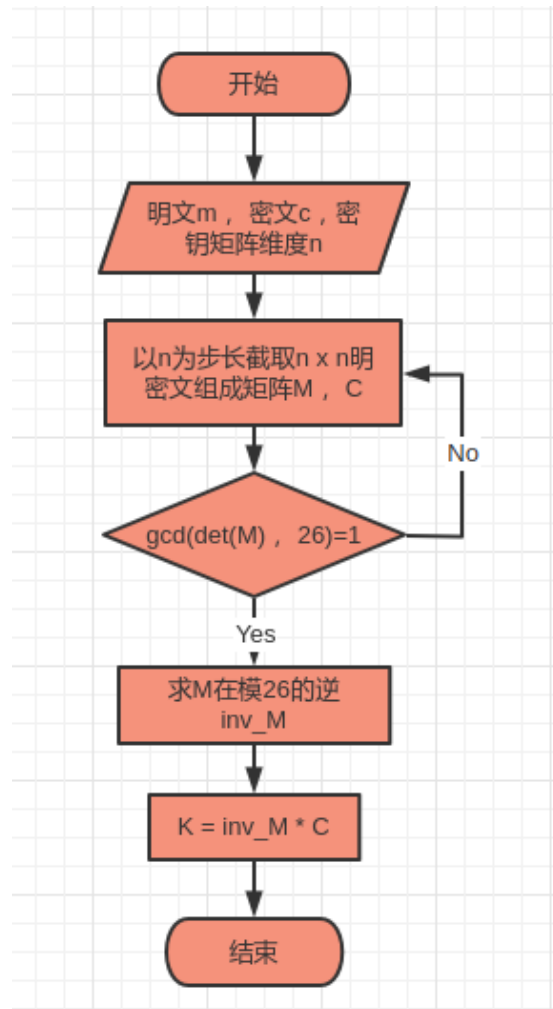
```
the cypher of tobeornottobethatisaquestion is reaootaeanttseoitobsitohuntqn
the plaintext of obestdnfhhmoeaaohleywsdloreb is ohehtworhaldsbemaendebooyfls
```

### 3.8.3 总结

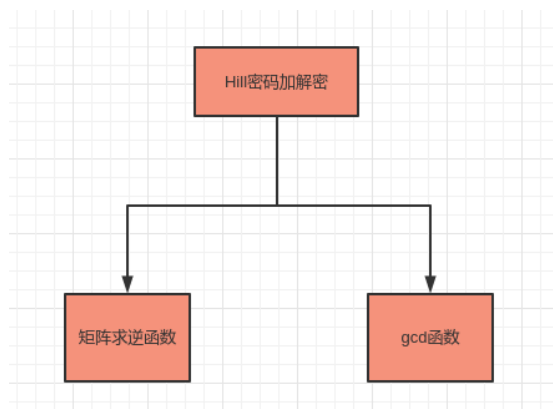
与普通的栅栏密码相比，矩阵密码的破解更加复杂。可以通过多次矩阵密码加密提高安全性

### 3.9 Hill 密码已知明文攻击

#### 3.9.1 算法流程



### 3.9.2 函数调用



### 3.9.3 算法伪代码

---

**Algorithm 12** Hill 密码已知明文攻击

---

**Input:**  $m, c, n$ , Function  $gcd$ ,  $Matrix\_inv$

**Output:**  $k$

```

1: for  $m[x \times n, (x + 1) \times n]$  in  $m$  do
2:    $M \leftarrow m[x \times n, (x + 1) \times n]$ 
3:    $C \leftarrow c[x \times n, (x + 1) \times n]$ 
4:   if then  $gcd(det(M), 26) \neq 1$ 
5:     Continue
6:   else
7:     return  $K \leftarrow Matrix\_inv(M) \times C$ 
8:   end if
9: end for
  
```

---

### 3.9.4 测试样例及结果截图

```

from plaintext youarepretty and cypher kqoimjvdbokn we crack the key:
[
2, 3      3.9.5 总结      318      \EndIf
1, 22     SNIPPET VIEW    319      \EndFor
]
from plaintext youaresocute and cypher ywwpcwsogfuk we crack the key:
Oops!!Fail to crack with plain youaresocute and cypher ywwpcwsogfuk
  
```

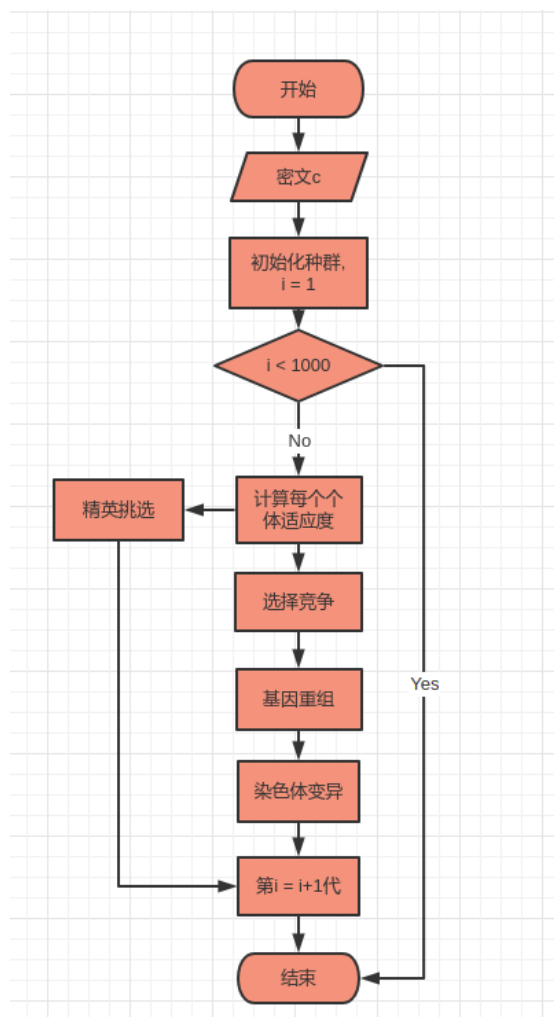
### 3.9.5 总结

1. Hill 密码不能抵抗已知明文攻击
2. 不是所有明密文对都可以破解出密钥，需要可以找到明文矩阵  $M$  可逆
3. 当密钥矩阵大小  $n$  未知时，如果其不太大，可以通过遍历猜测，如果得到的密钥对其他明密文对也生效，则认为破解出  $n$  以及密钥

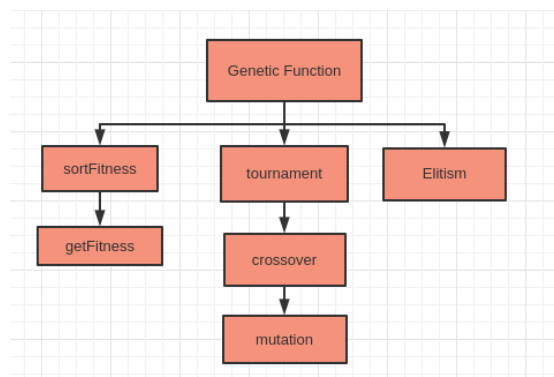


### 3.10 单表代替密码字母频率攻击

#### 3.10.1 算法流程



### 3.10.2 函数调用



### 3.10.3 算法伪代码

---

**Algorithm 13** Main Function

---

**Input:**  $c$ ,  $traintext$

**Output:**  $k$

```
1: get 3Gram frequency from traintext
2: randomly initialize poputation
3: for  $i \leftarrow 1$  to 1000 do
4:   caculate the fitness and the sort
5:   select the Elitism individual and pass them to next poputation
6:   while nextpoputation size < poputation size do
7:     select individual、hold tournament, and then get parents
8:     crossover between parents and get two children
9:     mutation
10:    add the two children to the next poputation
11:   end while
12: end for
```

---

### 3.10.4 测试样例及结果截图

### 3.10.5 总结

**算法准确度** 算法的准确性在文件大小超过 5kB 后基本可以得到唯一正确密钥。



**适应度函数** 选择  $\log_2(\text{trainFre}) \times \text{decryptFre}$  作为适应度函数，去除低概率三元字节的影响，从而让算法在较小文本时仍能保持正确性

**不足** 在对较大的文本，如几百 Kb 的文本，普通频率分析算法已经有一定的准确度，且速度优于遗传算法

## 4 总结

通过本次实验，了解了古典加密算法思想，掌握了常见的古典密码，并且学会应用古典密码。同时对单表代替密码保留词频规律导致易受频率攻击的缺点有了进一步的理解，从而进一步体会到设计密码时扩散的重要性。