

# 密码学实验报告 3

张天辰 17377321

2019 年 3 月 20 日

## 1 仿射密码

### 1.1 简介

仿射密码就是基于  $c \equiv a * m + b \pmod{26}$  的加密方法，其中要求  $\gcd(a, 26) = 1$ 。

### 1.2 算法实现

---

**Algorithm 1** 仿射密码

---

**输入:** 密钥  $a, b$ , 明文  $m$  (加密) 或密文  $c$  (解密)

**输出:** 密文  $c$  (加密) 或由密钥得到的明文  $m'$

```
1: function AFFINEENCRYPT( $message, a, b$ )
2:    $cipher \leftarrow []$ 
3:   for each  $m \in message$  do
4:      $cipher.append((a * int(m) + b) \bmod 26)$ 
5:   end for
6:   return  $cipher \rightarrow string$ 
7: end function
8: function AFFINEDECRYPT( $cipher, a, b$ )
9:    $message \leftarrow []$ 
10:  for each  $c \in cipher$  do
11:     $message.append((int(c) - b) * REVERSE(a) \bmod 26)$ 
12:  end for
13:  return  $message \rightarrow string$ 
14: end function
```

---

### 1.3 测试样例

```

Input the message:
ARRIVEATNORMANDIEATHALFPASTSIX
Input a: 5
Input b: 66
Encrypt finished. The cipher is:
OVVCPIOFBGVWOBDCIOFXORNLOAFACZ

=====

Input keya: 3
Input keyb: 25
TOOJWBTSQVORTGMJBTSUTCQKTDSDJA
Seem to be the wrong key.
Input keya: 21
Input keyb: 66
ARRIVEATNORMANDIEATHALFPASTSIX

```

图 1: Affine

## 2 Vigenere 密码

### 2.1 简介

此加密方案即将明文和密钥（循环使用）的对应字母相加再模 26 得到的文字作为密文。

---

#### Algorithm 2 Vigenere 密码

---

**输入:** 密钥  $key$ , 明文  $message$  (加密) 或密文  $cipher$  (解密)

**输出:** 密文  $cipher$  (加密) 或明文  $message$  (解密)

```

1: function VIGENEREENCRYPT( $message, key$ )
2:    $cipher \leftarrow []$ 
3:    $lenK \leftarrow len(key)$ 
4:   for each  $i \in [0, len(message))$  do
5:      $temp \leftarrow (message[i] + key[i \% lenK]) \bmod 26$ 
6:      $cipher.append(temp)$ 
7:   end for
8:   return  $cipher \rightarrow string$ 
9: end function
10: function VIGENEREDECRYPT( $cipher, key$ )
11:    $message \leftarrow []$ 
12:    $lenK \leftarrow len(key)$ 
13:   for each  $i \in [0, len(cipher))$  do
14:      $temp \leftarrow cipher[i] - key[i \% lenK]$ 

```

```

15:     message.append(temp)
16: end for
17: return message → str
18: end function

```

---

## 2.2 测试样例

```

Input message:
AkagiandKagahaveallbeenpwned
Input key:
Enterprise
Encrypt finished. The cipher is:
EXTKZPELCEKNAEMTRTDFIRGTNCVL

=====

Input the key to decrypt:
YorkCity
Seem to be the wrong key.
Input the key to decrypt:
Saratoga
Seem to be the wrong key.
Input the key to decrypt:
Enterprise
AKAGIANDKAGAHAVEALLBEENPWNE

```

图 2: *Virgenere*

## 3 Vernam 密码

### 3.1 算法简介

该密码方案将明文和密钥转换为二进制码，然后按位异或（密钥循环）得到密文。解密方法与加密相同。特别地，为了实现中文加密，如果输入语言为中文则每个字符占 16 位，如果为英文则为 8 位。

### 3.2 算法实现

---

**Algorithm 3** Vernam 密码

---

输入: 密钥 *cipher*, 明文 *message* (加密) 或密文 *cipher* (解密)

输出: 明文 *message* (解密) 或密文 *cipher* (加密)

```

1: function VERNAMENCRYPT(message, key)
2:   cipher  $\leftarrow \square$ 
3:   tempm  $\leftarrow \text{bin}(\text{message})$ 
4:   tempk  $\leftarrow \text{bin}(\text{key})$ 
5:   lenK  $\leftarrow \text{len}(\text{key})$ 
6:   for each i  $\in [0, \text{tempm})$  do
7:     cipher.append(tempm  $\oplus$  tempk  $\rightarrow \text{str}$ )
8:   end for
9:   return cipher
10: end function
11: function VERNAMDECRYPT(cipher, key)
12:   message  $\leftarrow \square$ 
13:   tempc  $\leftarrow \text{bin}(\text{cipher})$ 
14:   tempk  $\leftarrow \text{bin}(\text{key})$ 
15:   lenK  $\leftarrow \text{len}(\text{key})$ 
16:   for each i  $\in [0, \text{tempc})$  do
17:     cipher.append(tempc  $\oplus$  tempk  $\rightarrow \text{str}$ )
18:   end for
19:   return message
20: end function

```

### 3.3 测试样例

```

You are encrypting Chinese(input 1) or English(input 2) ?
1
曹将军与文若守城，五千兵士与吾趁夜直捣乌巢，烧尽袁本初粮草
宁我负人，毋人负我
0011110101111000001111100001011111011100100001000000000101101001001101010001011
11101001001011100001010100110010110110101101000110011101000111010001010100010101
00110000101001010110110001101010000101100101000110110001000000100011111111110101
110000110011101110101000000011000101001110010100111000111000100010110001011101
1101000011111101011000110110110100011111110101100110111111101101100011000111011
1110101000110011001100000000110000100111001011111110000101011000
=====
攬二喬于東南兮，樂朝夕之與共
宏花鎗为毀器謀、許燒架劫鍾[?]專駝驗媼盾緝膺人爰僂蹶#總癩芥
Seem like the wrong key.
东临碣石，以观沧海
揪灌[?]睇文[?]鳩臥[?]嬌孌[?]憐与焙配習祿盾扩[?]發燒燬侄姪嶧棍習
Seem like the wrong key.
宁我负人，毋人负我
曹将军与文若守城，五千兵士与吾趁夜直捣乌巢，烧尽袁本初粮草
Successfully decrypted!

```

图 3: *Vernam*

## 4 单表代替密码

### 4.1 简介

利用词频分析可以攻击单表代替密码。通过给不同的字母频率分级并进行枚举猜测,可以实现这种攻击。首先将高频的 e, t 字母单独处理,在高频的两个密文字母中轮换;再处理 a、e、i、o 四个中频字母,在四个中频密文字母中轮换;最后按照概率顺序处理其余的所有字母。

输出顺序也与概率相关。因为 e 概率大于 t, a、e、i、o 概率依次递减,所以按照全排列生成的顺序就是理论上概率从大到小的顺序。本算法按照这个顺序进行输出。

### 4.2 算法实现

---

**Algorithm 4** Hill 密码及其破解

---

输入: 密文 *cipher*

输出: 明文 *message*, 密钥 *key*

```

1: dict_high  $\leftarrow$  e, t 两个高频字母
2: dict_medium  $\leftarrow$  a, o, i, n 四个中频字母
3: dict_low  $\leftarrow$  其余字母
4: function STATISTICS(cipher)
5:   statis  $\leftarrow$ 
6:   将每个字母写入字典的 key, 出现频率作为 value
7:   statis 按照 value 排序
8: end function
9: function CRACK(cipher)
10:  probab  $\leftarrow$  STATISTICS(cipher)
11:  keyd  $\leftarrow$ 
12:  for probab[0, 1] 分别对应 e, t do
13:    对应写入 keyd
14:    for probab[2, 5] 分别对应 a, o, i, n 全排列 do
15:      对应写入 keyd
16:      其余按概率排序分别写入 keyd
17:      对应字典解密, 输出
18:    end for
19:  end for
20: end function

```

---

### 4.3 测试样例

gentlemen of the jury, order the prisoner to be released! Mr. president, have me arrested. He is not the man whom you are in search of; it is I: I am Jean Valjean.

a: 'a', 'b', 't', 'c', 'd', 'd', 'a', 'e', 'z', 'f', 'u', 'g', 'v', 'h', 'v', 'j',  
'i', 'n', 'j', 'a', 'k', 'r', 'l', 'j', 'm', 'b', 'n', 'l', 'o', 'c', 'p', 'f',  
'q', 'e', 'r', 'p', 's', 'k', 't', 'o', 'u', 'h', 'v', 's', 'w', 'g', 'x', 'f', 'm',  
'y', 'x', 'z', 'w';

vzljzblzcuoyzqzpczpozpyzfpnkclzpoctzpzjzkbzpbzfkazlzoyszjbzjppzkozaynzlcooyz  
zbilgycbxhczpiznlzqzpydumoknknbzqizlsjzqizl  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
penicdenomihewuftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo  
peicdeceadmmhuwftlntlethegtsrnetiobetaceareltdgtersleinhoyedeaternelhesriioah  
eadivhdfowatesrleatkhnmsrssdzoeyuacuo

图 4: *SubstitutionCipher*

## 5 Hill 密码

## 5.1 简介

此加密方案用矩阵作为密钥，对明文消息利用矩阵乘法加密。解密时利用乘矩阵逆元解密。同时，如果获得足够多明密文对，可以对其进行攻击。设明文列向量排成矩阵  $M$ ，其加密结果排成矩阵  $C$ ，密钥为  $K$ ，则有：

$$KM = C \quad K = CM^{-1}$$

可以实现对密钥的攻击。

为了提高程序运行效率，本算法使用 `python` 自带函数求解矩阵行列式。如果自己写代码，在矩阵阶数较高（如 256）时，其计算量将是惊人的。此外，为了实现支持中文加密，本方案放弃了  $\text{mod } 26$  的矩阵，而是选择了在  $GF_2$  上构造矩阵。这样的优点是保证了所有运算能够在域上完成。

矩阵求逆算法采用高斯消元法，复杂度为  $O(n^3)$ 。

## 5.2 算法实现

---

**Algorithm 5** Hill 密码及其破解

输入: 密钥 *cipher*, 明文 *message* (加密) 或密文 *cipher* (解密)

**输出:** 明文 *message* (解密) 或密文 *cipher* (加密)

```

1: function REVERSE( $a$   $n$ )
2:   for each  $k \in [0, n)$  do
3:     找到  $a[k][k]$  及其右下角子式中最大元, 并与通过行列变换与  $a[k][k]$  交换
4:     交换的行列保存在  $rmax[], cmax[]$ 
5:     for each  $j \in [0, n)$  do
6:       if  $j \neq k$  then  $a[k][j] \leftarrow a[k][j] * a[k][k]$ 

```

```

7:         end if
8:     end for
9:     for each  $i \in [0, n)$  do
10:         if  $i \neq k$  then
11:             for each  $j \in [0, n)$  do
12:                 if  $j \neq k$  then  $a[i][j] \leftarrow a[i][j] \oplus (a[i][k] * a[k][j])$ 
13:             end if
14:         end for
15:     end if
16: end for
17: for each  $i \in [0, n)$  do
18:     if  $i \neq k$  then  $a[i][k] \leftarrow a[i][k] * a[k][k]$ 
19: end if
20: end for
21: end for
22: 将之前每次循环开始做的行、列变换按照后到先出顺序做其逆变换
23: return  $a[][]$ 
24: end function
25: function HILLENCRYPT( $key, message, size$ )
26:      $intm \leftarrow message \rightarrow bit$ 
27:      $cipher \leftarrow []$ 
28:     for  $i = 0; i < len(intm); i += size$  do  $cipher.append(keyintm[i : i + size])$ 
29: end for
30: return  $cipher \rightarrow str$ 
31: end function
32: function HILLDECRYPT( $cipher, key, size$ )
33:      $intc \leftarrow cipher \rightarrow bit$ 
34:      $message \leftarrow []$ 
35:      $key\_1 \rightarrow REVERSE(key, size)$ 
36:      $tempk \leftarrow bin(key)$ 
37:      $lenK \leftarrow len(key)$ 
38:     for  $i = 0; i < len(intc); i += size$  do
39:          $message.append(key\_1intc[i : i + size])$ 
40:     end for
41:     return  $message \rightarrow str$ 
42: end function
43: function HILLPWN( $message, cipher, size$ )
44:      $intm \leftarrow message \rightarrow bit$ 

```

```
45:    $intc \leftarrow cipher \rightarrow bit$ 
46:   repeatmessage 中取  $size^2bit$  组成矩阵  $m_{mat}$   $cipher$  中取  $size^2bit$  组成矩阵  $c_{mat}$ 
47:   until  $DET(m_{mat}) \neq 0$ 
48:   return  $c_{mat}m_{mat}$ 
49: end function
```

---

5.3 测试样例

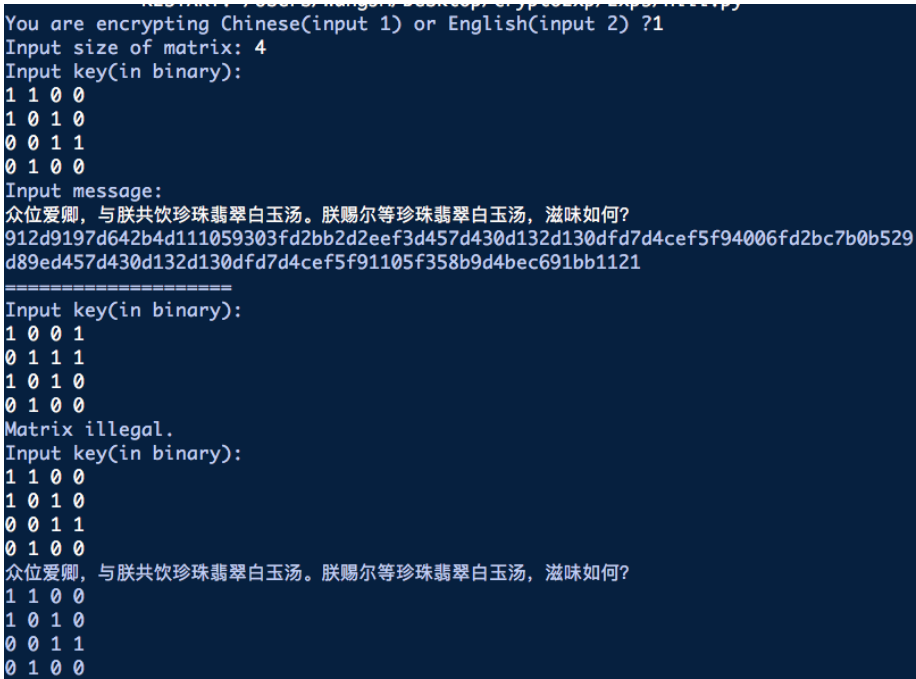


图 5: Hill

6 感想

作业量好大，时间好短。

单表代替密码果然没有那么容易破译，这样破解出来的东西完全没有可读性。如果想要破解这种密码，还是需要语言学家参与进行推断。这样简单的推断还是不够的，需要大量的语言学字典才可以。

希望这门课能变好。