

# 信息安全原理作业一

彭子帆 3170105860

## 1. 第一个问题

替代密码 (Substitution Cipher) 是指先建立一个替换表，加密时将需要加密的明文依次通过查表，替换为相应的字符，明文字符被逐个替换后，生成无任何意义的字符串，即密文，替代密码的密钥就是其替换表。置换密码 (Transposition Cipher) 只不过是一个简单的换位，每个置换都可以用一个置换矩阵  $E_k$  来表示。每个置换都有一个与之对应的逆置换  $D_k$ 。

替代密码用不同的位，字符或块替换字符的位，字符或块。置换密码不会用不同的文本替换原始文本，而是移动原始值。它重新排列字符的位，字符或块以隐藏原始含义。

1. 其中替代密码上课讲了很多，比如：

i. 希腊发明了如图所示的替代表

ii. 罗马皇帝 JULIUS CAESAR 发明了他自己的简单的加密

方式：每个字母向后移动三个位置。

iii. 著名的 Vigenère 方阵

iv. 除上课讲的还有比如 Playfair 密码：Playfair 是一种著名的双字母单表替代密码，实际上 Playfair 密码属于一种多字母替代密码，它将明文中的双字母作为一个单元对待，并将这些单元转换为密文字母组合。替代时基于一个  $5 \times 5$  的字母矩阵。字母矩阵构造方法同密钥短语密码

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

类似，即选用一个英文短语或单词串作为密钥，去掉其中重复的字母得到一个无重复字母的字符串，然后再将字母表中剩下的字母依次从左到右、从上往下填入矩阵中，字母 l, j 占同一个位置。

## 2. 置换密码：

一种最经典简单的置换密码加密方式如下：

首先设明文为： Hello, Information security principle.

可以分为 6 组，因此我们可以设置密钥为： 351624

将其按照字符（包括标点，可以包括空格，也可以不包括，这里我们不包含空格）分组，如一组 6 个，则可以排成下列矩阵(最后一行不足用符号填满，这里我使用 ‘&’ ):

	A	B	C	D	E	F
1	H	e	l	l	o	,
2	l	n	f	o	r	m
3	a	t	i	o	n	s
4	e	u	r	i	t	y
5	p	r	i	n	c	i
6	p	l	e	.	&	&

则根据密钥 351624，我们竖着读每一列，则可得密文为：

lfiriorntc&Hlaepp,msyi&enturllooiin.

至此，我们对 Hello, Information security principle.这句话加密完毕。接收方只需要知道密钥和密文就可以进行解码，其中密钥最大的值为其分组大小。

## 2. 编程环境

Microsoft Visual Studio Professional 2017

版本 15.9.4

VisualStudio.15.Release/15.9.4+28307.222

Visual C++ 2017 00369-60000-00001-AA074

Microsoft Visual C++ 2017

## 3. 描述算法设计

首先，根据信息安全原理课上学习的密码学基础的知识，由于 S-BOX 以及 P-BOX 不管几层加密都会起到最后一样的效果，就是简单的替换加密以及简单的换位加密，因此想要设计一个加密性高的加密算法，首先要进行一次 S-BOX, P-BOX 的操作。

其次，明确我们的加密算法的类型，接收方和发送方必须提前约定好密钥，我们进行对称加密算法。其中密钥为 10 位，其中 1、3、5、7、9 位为替换算法的密钥，2、4、6、8、10 位为换位算法的密钥。比如每次偏移一定位数。进行一次置换算法，然后进行一次换位加密。

i. 由于我们要加密的是字母和数字因此首先将其排序：

**abcdefghijklmnopqrstuvwxyz0123456789**

一共 36 个字符，由于有个最近的质数为 37 因此我取字符 '@' 放在字母与数字之间凑成 37 个字符，为后面取模随机性增大。

**abcdefghijklmnopqrstuvwxyz@0123456789**

- ii. 其次我们就要用到密钥了，假设密钥为：securityhw

则 scrth 为替换的密钥，euiyw 为等会要用到的换位的密钥。

根据维吉尼亚方阵我们将这个序列重新排序为：

其中除了密钥部分，其他地方顺序不变。

abdefgijklmnopq**scrth**uvwxyz@0123456789

- iii. 而后我们先进行替换加密算法：

如我的姓名加学号为：pengzifan3170105860

第一个字母'p'向后偏移  $1^2$  即变为'q'

第二个字母'e'向后偏移  $2^2$  即变为'j'

第三个字母'n'向后偏移  $3^2$  即变为'u'

.....

依次类推，如果遇到向后偏移到'9'依然不足，则跳到字母'a'继续偏移。

最后上述偏移结束后结果为：qjuvpgc0thb3nd321yt

- iv. 然后我们对上述序列继续进行重排序的操作。

重排密钥上面我们提到的是：euiyw。其中密钥中每个字母对应 abdefgijklmnopq**scrth**uvwxyz@0123456789 中一个位置（从 0 开始编号），比如 e 对应第 3 个位置，则 euiyw 分别对应 3,20,6,24,22。

由于我们对经过 S-BOX 的明文分组采用 5 个一组，因此分别将 3,20,6,24,22 除以 5 取余，可以得到 3,0,1,4,2。此时数据比较凑巧，没有出现重复值，若出现重复值我们可以按照哈希开放定址法的线

性探测，比如第二个字符取余也为 3，则应取 4 为其值。

则将经过 S-BOX 的明文进行从左往右 0~4 报数，报到相同数字的为的一组。

则（由于序列不是五的倍数，则应补充字符到 5 的倍数，我采用补充字符'&'直到 5 的倍数）：

0 组：qgb2

1 组：jc31

2 组：u0ny

3 组：vtdt

4 组：ph3&

将其按照 30142 连起来，最终加密密文为：vtdtqgb2jc31ph3&u0ny

至此，我们加密算法完成了。

## 4. 实验结果


实现功能：

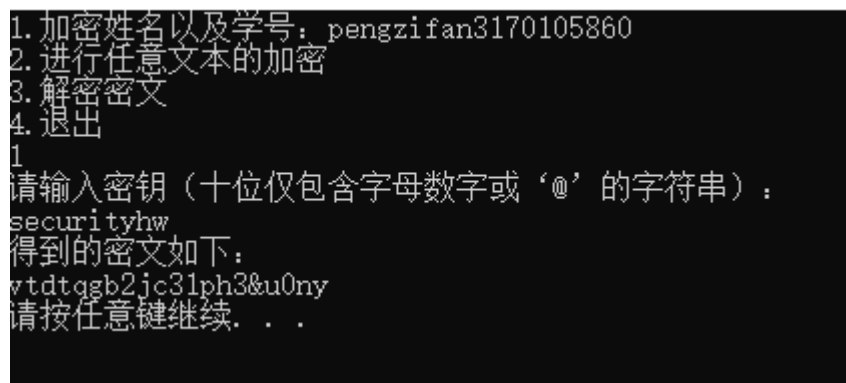
- 1.加密姓名以及学号：pengzifan3170105860
- 2.进行任意文本的加密
- 3.解密密文

a) 加密姓名学号如图所示：

加密 pengzifan3170105860 并输入密钥：securityhw

可以得到密文：vtdtqgb2jc31ph3&u0ny

 D:\大二下\信息安全原理\HW1-3170105860-彭子帆\执行文件.exe



```
1. 加密姓名以及学号: pengzifan3170105860
2. 进行任意文本的加密
3. 解密密文
4. 退出
1
请输入密钥（十位仅包含字母数字或 '@' 的字符串）:
securityhw
得到的密文如下:
vtdtqgb2jc31ph3&u0ny
请按任意键继续. . .
```

b) 如图所示用相同的密文与密钥进行解密即可得到姓名学号

```
D:\大二下\信息安全原理\HW1\Debug\HW1.exe
1. 加密姓名以及学号: pengzifan3170105860
2. 进行任意文本的加密
3. 解密密文
4. 退出
3
请输入密文:
vtdtqgb2jc3lph3&u0ny
请输入密钥（十位仅包含字母（小写）数字或‘@’的字符串）:
securityhw
得到的明文如下:
pengzifan3170105860
请按任意键继续. . .
```

c) 也可加密任意明文:

比如加密 informationsecurityprinciple

并输入密钥: peng@zifan

可以得到密文: 0vpav&alwiivqlikr8hbxgo&nmxoxm

```
D:\大二下\信息安全原理\HW1-3170105860-彭子帆\源代码.exe
1. 加密姓名以及学号: pengzifan3170105860
2. 进行任意文本的加密
3. 解密密文
4. 退出
2
请输入明文（小写或数字）:
informationsecurityprinciple
请输入密钥（十位仅包含字母数字或‘@’的字符串）:
peng@zifan
得到的密文如下:
0vpav&alwiivqlikr8hbxgo&nmxoxm
请按任意键继续. . .
```

然后我们进行解密，输入同样的密文以及密钥可以得到：

```
D:\大二下\信息安全原理\HW1\Debug\HW1.exe
1. 加密姓名以及学号: pengzifan3170105860
2. 进行任意文本的加密
3. 解密密文
4. 退出
3
请输入密文:
0vpav&alwiivqlikr8hbxgo&nmxoxm
请输入密钥（十位仅包含字母（小写）数字或‘@’的字符串）:
peng@zifan
得到的明文如下:
informationsecurityprinciple
请按任意键继续. . .
```

## 5. 总结与经验

简单的加密算法的设计十分容易，但是如果想要考虑到多种因素并且让人难以破解，是十分困难的，需要进行数学上的计算与演算。道高一尺魔高一丈，密码学也是在一次次设计加密算法以及破解中进步的。这是一对矛盾也是一对共同发展进步的学问，相互依赖与依存。我在设计完成后，在具体地输入代码以及实现的过程中也遇到了很多漏洞，需要不断修改来对付这些问题。

总之，这次的作业给了基础地设计加密算法的经验，让我对其产生浓厚兴趣，也对信息安全原理这门课更加充满了期待！