密码学实验报告

学号：2111033 姓名：艾明旭 年级：2021 专业：信息安全

⼀、实验内容说明

1、实验⽬的

通过C＋＋编程实现移位密码和单表置换密码算法，加深对经典密码体制的了解。并通过对这两种密码实施

攻击，了解对古典密码体制的攻击⽅法。

2、实验要求

要求上述密码算法提供最后的算法流程图，并写出明⽂、加解密的结果。字⺟频率统计攻击⽅法要求写明置

换表中确定每个字⺟的原因和攻击的步骤。

3、实验步骤

. 根据实验原理部分对移位密码算法的介绍，⾃⼰创建明⽂信息，并选择⼀个密钥，编写移位密码算法

实现程序，实现加密和解密操作。

\_x0019\_. 两个同学为⼀组，互相攻击对⽅⽤移位密码加密获得的密⽂，恢复出其明⽂和密钥。

. ⾃⼰创建明⽂信息，并选择⼀个密钥，构建置换表。编写置换密码的加解密实现程序，实现加密和解

密操作。

. ⽤频率统计⽅法，试译下⾯⽤单表置换加密的⼀段密⽂：

SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPNBRQJSSJBE JBFMPQNSJMB

FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJHAY JBRCGZPC GINBBCA JB

RZGI N VNY SINS SIC MPJEJBNA QCRRNEC GNB MBAY HC PCGMTCPCD HY SIC PJEISFZA

PCGJXJCBSR SIC XNPSJGJXNBSR JB SIC SPNBRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF

SIC QCRRNEC HMH SIC PCGCJTCP NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRICR SM

ENJB ZBNZSIMPJOCD GMBSPMA MF SIC QCRRNEC

⼆、实验环境

操作系统：macOS Montery 12.4

软件系统：Visual studio

编译⼯具：Visual studio 2022

编程语⾔：C++

实验一：移位密码

算法流程图

主函数，根据输入提供三种不同功能



输入3，进行攻击

只需要提供密文，然后使用for循环，先判断是否是字母，再判断字母的大小写，然后移位1位，输出结果，此步骤共重复25次，得到25个结果，然后人工判断正确的明文

输入2，进行加密

需要提供秘钥以及密文，然后使用for循环，先判断是否是字母，再判断字母的大小写，然后按照秘钥进行反向移位解密。

输入1，进行加密

需要提供秘钥以及明文，然后使用for循环，先判断是否是字母，再判断字母的大小写，然后按照秘钥进行移位。

加密解密过程：

加密函数⾸先需要⽤户选择密钥

然后输⼊⾃⼰需要加密的密⽂

通过for循环来对其加密，不是字⺟的字符不会变化，是字⺟的字符进⾏移位加密

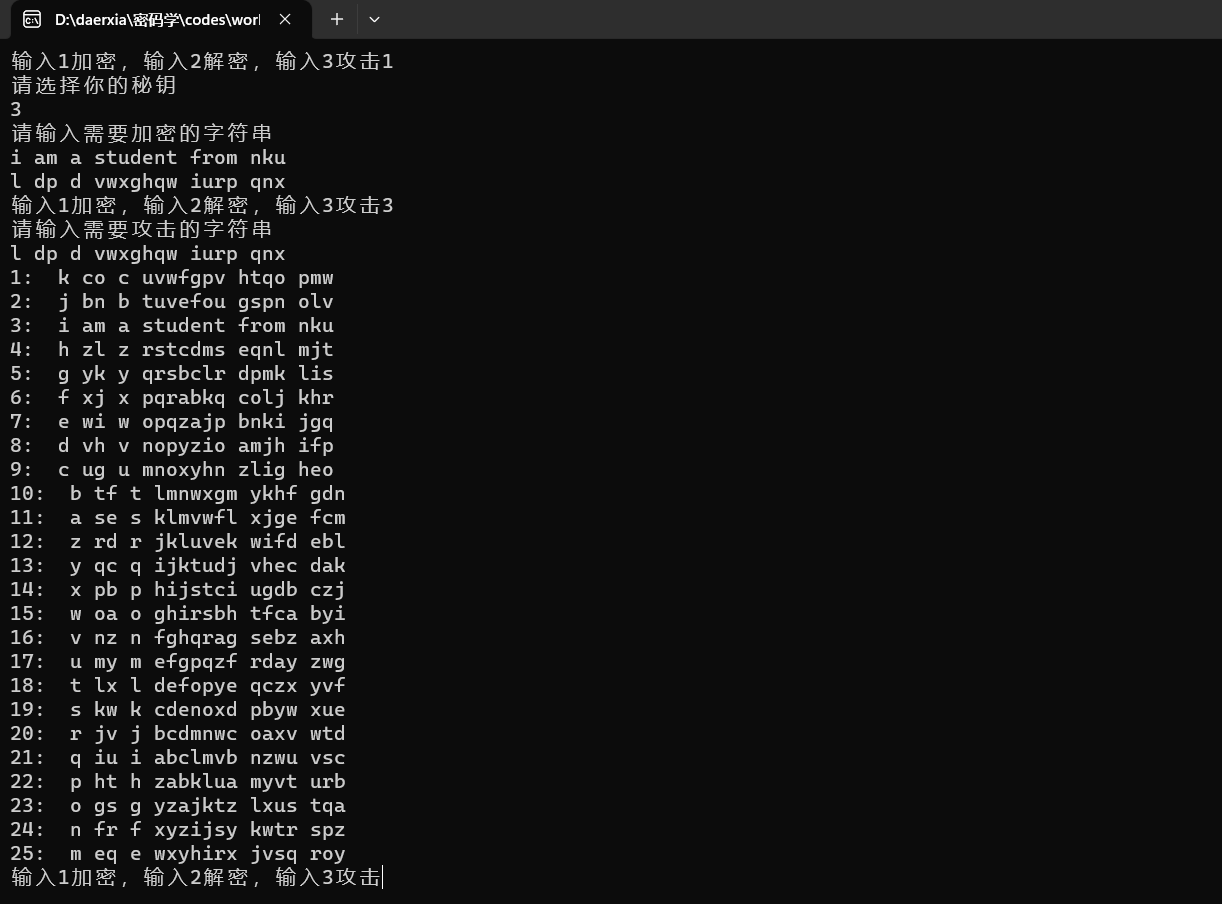
最后输出加密结果

明文 i am a student from nku

秘钥 3

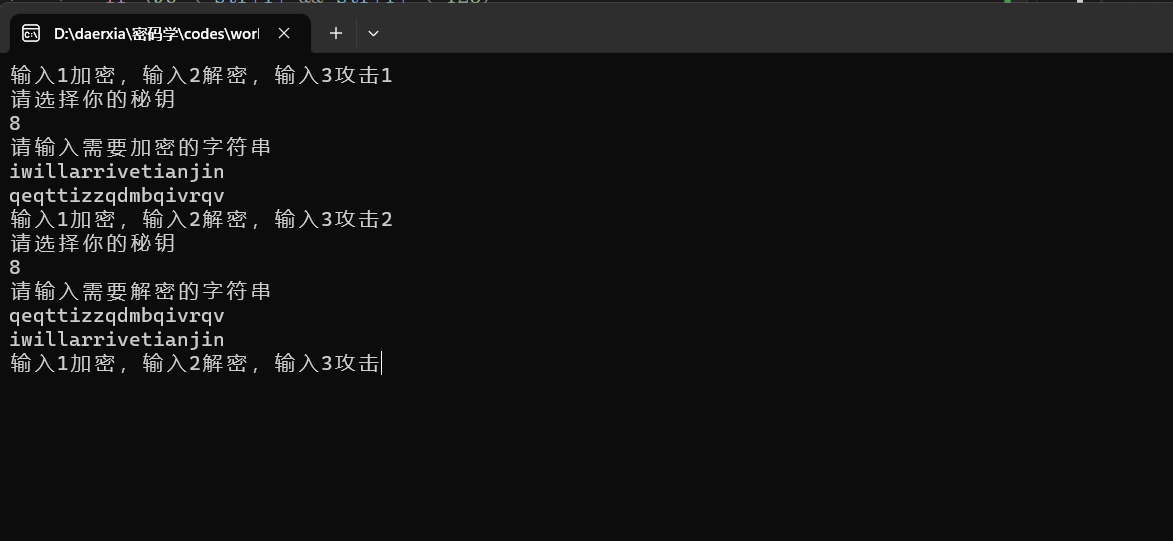
密文l dp d vwxghqw iurp qnx

加密解密结果



解密过程

解密运算是加密运算的逆运算，例如我们得到qeqttizzqdmbqivrqv，输入密钥进行解密得到iwillarrivetianjin



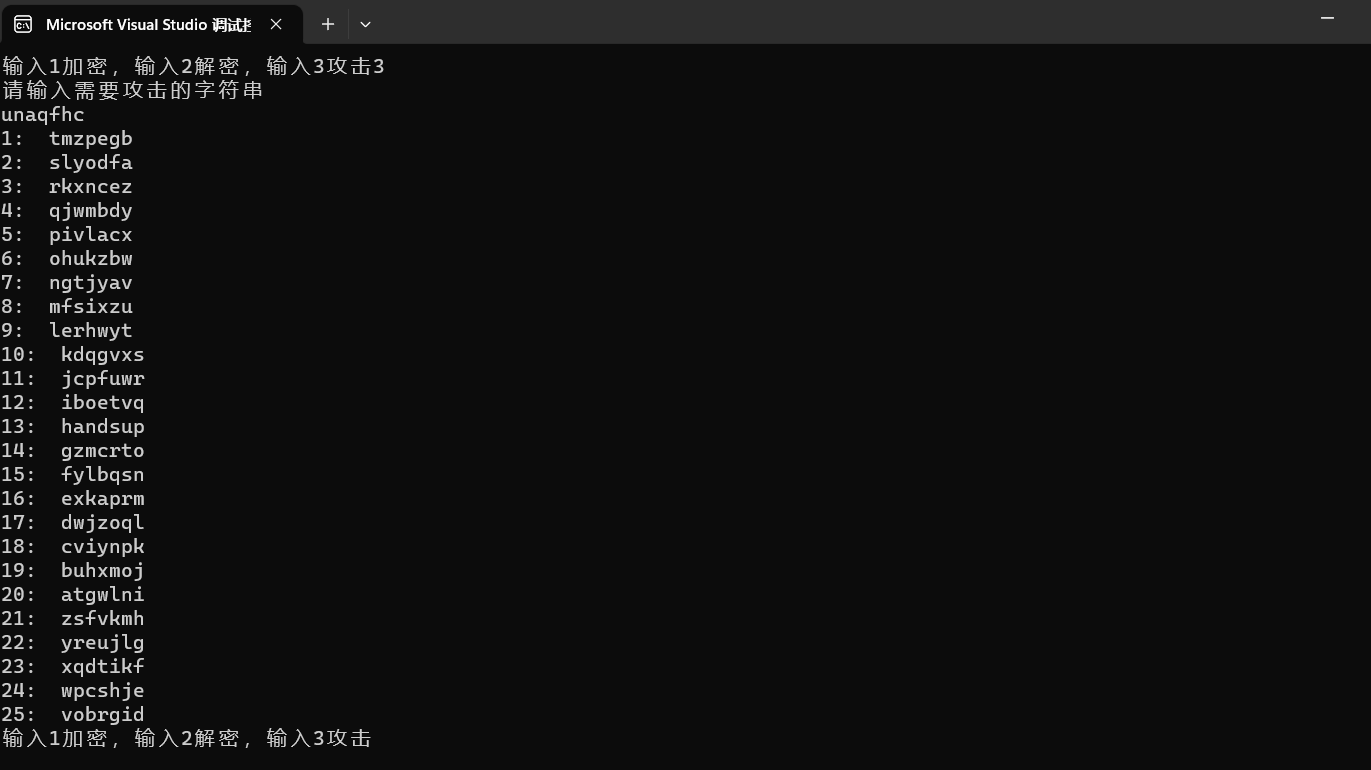
攻击过程

攻击函数⾸先需要⽤户输⼊需要攻击的字符串

通过for循环来对其攻击，⼀共需要进⾏25次

最后输出25个结果，并由⼈⼯判断正确的结果，具体函数如下：

例如我们得到密文unaqfhc，使用程序进行攻击



我们可以判断明文为handsup，秘钥为13

实验源码如下：

#include<iostream>

#include<string>

using namespace std;

void attack()

{

cin.ignore();

cout << "请输入需要攻击的字符串" << endl;

string str;

getline(cin, str);

for (int j = 1; j < 26; j++)

{

for (int i = 0; i < str.length(); i++)

{

if (64 < str[i] && str[i] < 91)

{

str[i] = str[i] - 1;

if (str[i] < 65)

str[i] += 26;

}

else

{

if (96 < str[i] && str[i] < 123)

{

str[i] = str[i] - 1;

if (str[i] < 97)

str[i] += 26;

}

}

}

cout << j << ": " << str << endl;

}

}

void lock()

{

cout << "请选择你的秘钥" << endl;

int a;

cin >> a;

cin.ignore();

a = a % 26;

cout << "请输入需要加密的字符串" << endl;

string str;

getline(cin, str);

for (int i = 0; i < str.length(); i++)

{

if (64 < str[i] && str[i] < 91)

{

str[i] = (str[i] + a) % 91;

if (str[i] < 65)

str[i] += 65;

}

else

{

if (96 < str[i] && str[i] < 123)

{

str[i] = (str[i] + a) % 123;

if (str[i] < 97)

str[i] += 97;

}

}

}

cout << str << endl;

}

void unlock()

{

cout << "请选择你的秘钥" << endl;

int a;

cin >> a;

cin.ignore();

a = a % 26;

cout << "请输入需要解密的字符串" << endl;

string str;

getline(cin, str);

for (int i = 0; i < str.length(); i++)

{

if (64 < str[i] && str[i] < 91)

{

str[i] = str[i] - a;

if (str[i] < 65)

str[i] += 26;

}

else

{

if (96 < str[i] && str[i] < 123)

{

str[i] = str[i] - a;

if (str[i] < 97)

str[i] += 26;

}

}

}

cout << str << endl;

}

int main()

{

while (true)

{

cout << "输入1加密，输入2解密，输入3攻击";

int a;

cin >> a;

switch (a)

{

case 1:

lock();

break;

case 2:

unlock();

break;

case 3:

attack();

}

}

}

置换密码表原理

单表置换实现的⼀个关键问题是关于置换表的构造。置换表的构造可以有各种不同的途径，主要考虑的是记

忆的⽅便。如使⽤⼀个短语或句⼦，删去其中的重复部分，作为置换表的前⾯的部分，然后把没有⽤到的字

⺟按字⺟表的顺序依次放⼊置换表中，本次实验就是⽤这种⽅式来构造的。

加密解密过程

加密函数⾸先需要⽤户输⼊⼀个字符串⽤来构造密码表

然后调⽤get\_secret()函数来构造密码表并在屏幕上显示

然后输⼊⾃⼰需要加密的字符串

通过for循环并按照密码表来对其加密，不是字⺟的字符不会变化，是字⺟的字符进⾏加密

解密函数⾸先需要⽤户输⼊⼀个字符串⽤来构造密码表

然后调⽤get\_secret()函数来构造密码表并在屏幕上显示

然后输⼊⾃⼰需要解密的字符串

通过for循环并按照密码表来对其解密，不是字⺟的字符不会变化，是字⺟的字符进⾏解密

实验二单标置换密码

主函数，根据输入提供三

两种不同功能

程序流程图

输入2，进行加密

用户输入置换密码表

输入1，进行加密

用户输入置换密码表

Get\_secret()

程序根据用户的输入构造完整的置换密码表。系统顺序的遍历用户输入的字符串,若遇到的是字母，首先判断是否是大写字母，如果不是则转换成大写字母，然后判断已构造的密码表中是否有这个字母，如果没有，则把它添加到密码表中，当遍历完后，如果还有字母没有添加到表中，则按照英语字母顺序添加到密码表中。

Get\_secret()

程序根据用户的输入构造完整的置换密码表。系统顺序的遍历用户输入的字符串,若遇到的是字母，首先判断是否是大写字母，如果不是则转换成大写字母，然后判断已构造的密码表中是否有这个字母，如果没有，则把它添加到密码表中，当遍历完后，如果还有字母没有添加到表中，则按照英语字母顺序添加到密码表中。

用户输入需要解密的密文

用户输入需要加密的明文

解密时我们需要先把小写字母转换为大写字母查表，然后再转换成小写字母替换，大写字母则直接查找替换即可

由于加密查找的时候，密码表是按照abcdefg正常顺序排列的，所以我们只需要判断段字母大小写，大写字母减去65（A的ascii）小写字母减去97（a的 ascii），然后根据下标替换即可

实验源码如下：

#include<iostream>

#include<string>

using namespace std;

string str1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ", str2, a;

string get\_secret(string str)

{

string secret(26, ' ');

int num = 0;

for (int i = 0; i < str.length(); i++)

{

if ((96 < str[i] && str[i] < 123) || (64 < str[i] && str[i] < 91))

{

if (96 < str[i] && str[i] < 123)

str[i] -= 32;

for (int j = 0; j <= num; j++)

{

if (str[i] == secret[j])

break;

else

{

if (j == num)

{

secret[j] = str[i];

num++;

break;

}

}

}

}

}

for (int i = 0; i < 26; i++)

{

for (int j = 0; j < num; j++)

{

if (str1[i] == secret[j])

break;

else

{

if (j == num - 1)

{

secret[j + 1] = str1[i];

num++;

break;

}

}

}

}

cout << "====================================================" << endl;

cout << "根据输入所构造的密码表为：" << endl << str1 << endl << secret << endl;

cout << "====================================================" << endl;

return secret;

}

void lock()

{

cin.ignore();

cout << "请输入一个你喜欢的短语以构造置换密码表：" << endl;

getline(cin, str2);

str2 = get\_secret(str2);

cout << "请输入你想要加密的字符串：" << endl;

getline(cin, a);

for (int i = 0; i < a.length(); i++)

{

if (64 < a[i] && a[i] < 91)

{

a[i] = str2[a[i] - 65];

}

else

{

if (96 < a[i] && a[i] < 123)

a[i] = str2[a[i] - 97] + 32;

}

}

cout << "====================================================" << endl;

cout << "加密结果为：" << endl;

cout << a << endl;

cout << "====================================================" << endl;

}

void unlock()

{

cin.ignore();

cout << "请输入一个你喜欢的短语以构造置换密码表：" << endl;

getline(cin, str2);

str2 = get\_secret(str2);

cout << "请输入你想要解密的字符串：" << endl;

getline(cin, a);

for (int i = 0; i < a.length(); i++)

{

if (64 < a[i] && a[i] < 91)

{

for (int j = 0; j < 26; j++)

{

if (a[i] == str2[j])

{

a[i] = str1[j];

break;

}

}

}

else

{

if (96 < a[i] && a[i] < 123)

{

for (int j = 0; j < 26; j++)

{

if (a[i] - 32 == str2[j])

{

a[i] = str1[j] + 32;

break;

}

}

}

}

}

cout << "====================================================" << endl;

cout << "解密结果为：" << endl;

cout << a << endl;

cout << "====================================================" << endl;

}

int main()

{

// cout << 1 << endl;

while (true)

{

cout << "====================================================" << endl;

cout << "这是我的单表置换加密器，以下为你可以进行的操作：" << endl;

cout << "1.加密" << endl;

cout << "2.解密" << endl;

cout << "3.退出" << endl;

cout << "====================================================" << endl;

cout << "请输入相应数字进行你想要的操作：" << endl;

int a;

cin >> a;

switch (a)

{

case 1:

lock();

break;

case 2:

unlock();

break;

case 3:

cout << "bye!" << endl;

return 0;

}

}

return 0;

}

秘钥：Hello,how are you?

密码表：

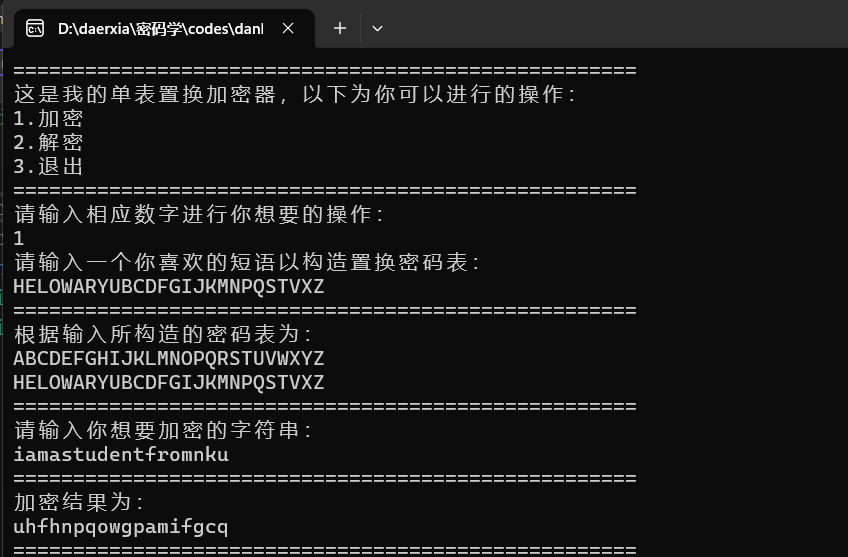
ABCDEFGHIJKLMNOPQRSTUVWXYZ

HELOWARYUBCDFGIJKMNPQSTVXZ

明文：Hello,how are you?

密文：Ywddi,yit hmw xiq?

加密解密结果



实验三 单表置换密码攻击

|  |
| --- |
| SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPNBRQJSSJBE JBFMPQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJHAY JBRCGZPC GINBBCA JB RZGI N VNY SINS SIC MPJEJBNA QCRRNEC GNB MBAY HC PCGMTCPCD HY SIC PJEISFZA PCGJXJCBSR SIC XNPSJGJXNBSR JB SIC SPNBRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF SIC QCRRNEC HMH SIC PCGCJTCP NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRICR SM ENJB ZBNZSIMPJOCD GMBSPMA MF SIC QCRRNEC |

3、破译密⽂

密⽂在实验步骤处已经列出，这⾥就不再进⾏赘述，先对其进⾏分析并给出最后解密出的明⽂：

对密⽂进⾏分析

1*、单独字⺟*

我们可以看到图中单独出现的字⺟是N和H，各个字⺟出现的次数如下表所示：



但是正常字⺟出现的概率为：



2*、攻击分析*

1 ⾸先我们对表中的单个单词做分析，在英语中，可以单个出现的只有I、A，对应题⽬中的单词H、N

\_x0019\_2 接着我们尝试找到密⽂中出现频率最⾼的两个字⺟，我们可以看到分别是Cº36次，Sº33次，他们可能

对应的是e、t

3 接着我们观察单词SIC，其在句⼦⾥出现的频率极⾼，推测其对应单词the sic --- the

4 我们观察的单词SINS，由于已知S和I，所以我们可以推断出这个单词为that，即N对应字⺟a

Cina --- etha

\_x001D\_5 我们观察单词NPC 则我们可以判断P对应字⺟ r c s i n p m e t h a r o

6 我们观察SM，很容易判断它对应单词to c s i n p m e t h a r o

7 我们来查看VIM，他的对应的是who c s i n p m v e t h a r o w

8 我们观察MF多次出现，判断其为of c s i n p m v f e t h a r o w f

9 接着我们观察JB多次出现，同时也有JR出现，我们再观察频率，可以判断JB为in，JR为is

c s i n p m v f j b r e t h a r o w f i n s

10 我们观察FPMQ，很容易判断他为from c s i n p m v f j b r q ---e t h a r o w f i n s m

11我们观察XMJBS，很容易判断他为pointc s i n p m v f j b r q x---e t h a r o w f i n s m p

12 我们将FPMQ N XMJBS N SM N XMJBS H翻译过来，from a point a to a point H,这样我们

很容易判断H为b，同时后⾯的HY为by c s i n p m v f j b r q x h y

e t h a r o w f i n s m p b y

13 我们很容易判断XPMHACQ为problem c s i n p m v f j b r q x h y a

e t h a r o w f i n s m p b y l

14观察ENJB，我们很容易判断段为gain c s i n p m v f j b r q x h y a e

e t h a r o w f i n s m p b y l g

15观察GNB我们很容易判断其为can c s i n p m v f j b r q x h y a e g

e t h a r o w f i n s m p b y l g c

16观察NBD，我们很容易判断为and c s i n p m v f j b r q x h y a e g d

e t h a r o w f i n s m p b y l g c d

17观察PCGCJTCP，很容易判断为receiver c s i n p m v f j b r q x h y a e g d t

e t h a r o w f i n s m p b y l g c d v

18观察RZGI，很容易判断为such c s i n p m v f j b r q x h y a e g d t z

e t h a r o w f i n s m p b y l g c d v u

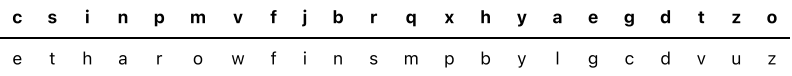
19最后我们观察ZBNZSIMPJOCD，很容易判断为unauthorized

c s i n p m v f j b r q x h y a e g d t z o

e t h a r o w f i n s m p b y l g c d v u z

3*、置换密码表*

最后得到的置换密码表如下表所示：

  
解密原⽂

于是我们便可根据以上关系进⾏解密，得到原⽂如下所示：

The central problem in cryptography is that of transmitting information

from a point a to a point b by means of a possibly insecure channel in

such a way that the original message can only be recovered by the rightful

recipients.

The participants in the transaction are alice , the originator of the

message; bob, the receiver and oscar ,a possible opponent who wishes to

gain unauthorized control of the message.

四、总结与展望

1、总结

本次是密码学的第⼀次实验，在这次实验中对理论课上讲解的移位密码以及单表置换密码进⾏编程，使得对

于这两种密码算法的原理和攻击⽅法更加的了解，掌握了相关的密码编写的逻辑思维和应变能力以及相关密码的复杂度和破译难度。也对密码学⽅⾯的编程更加的熟悉。同时我还对相应的单表置换密码进行了攻击以及利用日常用语进行了分析，从而得到了一段密文的解。

2、展望

在这次实验后，对密码学⽅⾯的知识更加的期待，也对这些原理和攻击⽅法更加的感兴趣，而在学习完一些现代密码方法之后再回顾古典密码的编写，我感受到了现代科学技术的进步和密码学的发展，对接下来的学习有了更强的动力。