



南开大学  
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

密码学 (1091) 第一次实验

---

## 古典密码算法及攻击方法

---

苗发生

年级：2020 级

专业：信息安全 & 法学双学位

指导教师：古力

2022 年 11 月 13 日

# 目录

<b>一、 实验内容</b>	<b>1</b>
(一) 实验目的 . . . . .	1
(二) 实验环境 . . . . .	1
(三) 实验内容和步骤 . . . . .	1
(四) 实验原理 . . . . .	1
1. 移位密码 . . . . .	1
2. 对移位密码的攻击 . . . . .	1
3. 单表置换密码 . . . . .	1
4. 单表置换密码的攻击 . . . . .	2
<b>二、 实验具体流程</b>	<b>2</b>
(一) 移位密码加密 . . . . .	2
(二) 移位密码解密 . . . . .	3
(三) 互相攻击对方用移位密码加密获得的密文 . . . . .	6
(四) 构建置换表 . . . . .	6
(五) 利用置换表进行加密 . . . . .	8
(六) 利用置换表进行解密 . . . . .	10
(七) 利用频率统计方法破译密文 . . . . .	12
1. 前置知识 . . . . .	12
2. 统计密文中字母出现的次数 . . . . .	13
3. 第一步 . . . . .	14
4. 第二步 . . . . .	15
5. 第三步 . . . . .	15
6. 第四步 . . . . .	15
7. 第五步 . . . . .	16
8. 总结 . . . . .	18

## 一、 实验内容

### (一) 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。

### (二) 实验环境

运行 Windows 操作系统的 PC 机, 具有 VC 等语言编译环境

### (三) 实验内容和步骤

1) 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

2) 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

3) 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

4) 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 JBFM-  
PQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJHAY JBR-  
CGZPC GINBBCA JB RZGI N VNY SINS SIC MPJEJBNA QCRRNEC GNB MBAY HC  
PCGMTCPCD HY SIC PJEISFZA PCGJXJCSR SIC XNPSJGJXNSR JB SIC SPN-  
BRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF SIC QCRRNEC HMH SIC PCGCJTCP  
NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRICR SM ENJB ZBNZSIMPJOCD  
GMBSPMA MF SIC QCRRNEC

### (四) 实验原理

#### 1. 移位密码

将英文字母向前或向后移动一个固定位置。例如向后移动 3 个位置, 即对字母表作置换 (不分大小写)

#### 2. 对移位密码的攻击

移位密码是一种最简单的密码, 其有效密钥空间大小为 25。因此, 很容易用穷举的方法攻破。穷举密钥攻击是指攻击者对可能的密钥的穷举, 也就是用所有可能的密钥解密密文, 直到得到有意义的明文, 由此确定出正确的密钥和明文的攻击方法。对移位密码进行穷举密钥攻击, 最多只要试译 25 次就可以得到正确的密钥和明文

#### 3. 单表置换密码

单表置换密码就是根据字母表的置换对明文进行变换的方法, 例如, 给定置换

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

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

明文: public keys, 则有

密文: mkebw qxuo

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

#### 4. 单表置换密码的攻击

在单表置换密码中,由于置换表字母组合方式有  $26!$  种,约为  $4.03 \times 10^{26}$ 。所以采用穷举密钥的方法不是一种最有效的方法。对单表置换密码最有效的攻击方法是利用自然语言的使用频率:单字母、双字母组/三字母组、短语、词头/词尾等,这里仅考虑英文的情况。英文的一些显著特征有短单词,常用单词,字母频率。这样,攻击一个单表置换密码,首先统计密文中最常出现的字母,并据此猜出两个最常用的字母,并根据英文统计的其他特征(如字母组合等)进行试译

## 二、实验具体流程

### (一) 移位密码加密

移位密码加密流程图如下所示:

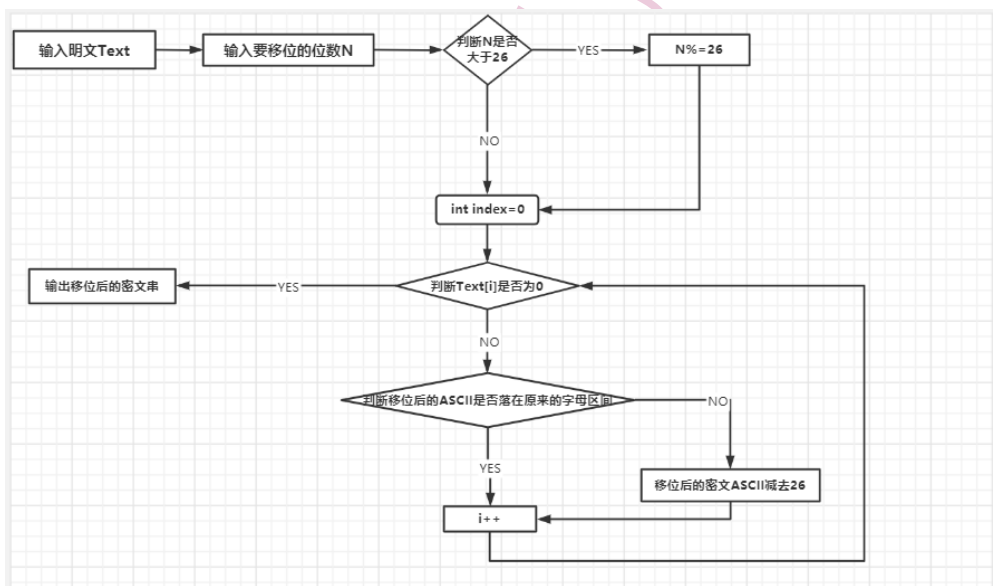


图 1: 移位密码加密流程图

根据上面移位密码加密流程图实现的 C++ 代码如下:

#### 移位加密

```

1 void Shiftencryption()
2 {
3     char Text[1000];
4     int shift = 0;
5     cout << "请输入您要加密的明文"<<endl;
6     cin >> Text;
7     cout << "请输入偏移位数 (右移) :";
8     cin >> shift;
  
```

```

9      shift %= 26;
10     int after = 0;
11     int shift_65 = shift + 65;
12     int shift_90 = shift + 90;
13     int shift_97 = shift + 97;
14     int shift_122 = shift + 122;
15     for (int i = 0; int(Text[i]) != 0; i++)
16     {
17         //计算移位后的ascii
18         after = int(Text[i]) + shift;
19         //判断移位后的
20         if (after >= shift_65 && after <= 90)
21         {
22             Text[i] = char(after);
23         }
24         if (after > 90 && after <= shift_90)
25         {
26             Text[i] = char(after - 26);
27         }
28         if (after >= shift_97 && after <= 122)
29         {
30             Text[i] = char(after);
31         }
32         if (after > 122 && after <= shift_122)
33         {
34             Text[i] = char(after - 26);
35         }
36     }
37     cout << "移位后的密文为: " << endl << Text << endl;
38     cout << "-----" << endl;
39 }

```

输入明文 I love Nankai University I am Miaofasheng 进行测试，测试结果如下：

```

请输入您要加密的明文
I love Nankai University I am Miaofasheng
请输入偏移位数（右移）:10
移位后的密文为：
S vyfo Xkxuks Exsfobcsdi S kw Wskypkroxq

```

图 2: 移位密码测试

得到密文结果：S vyfo Xkxuks Exsfobcsdi S kw Wskypkroxq  
经检验，移位加密正确！

## (二) 移位密码解密

由于移位密码加密的有效密钥空间大小为 25，故采用暴力破解的方式进行解密  
移位密码解密流程图如下所示：

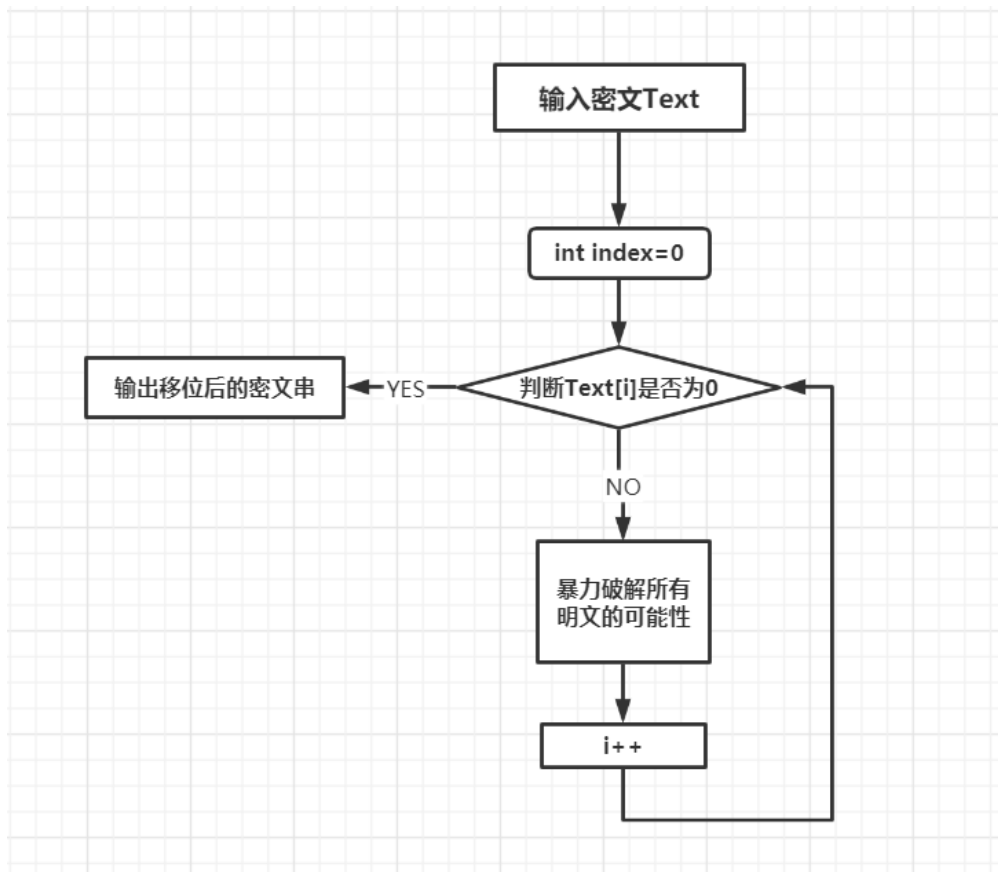


图 3: 移位密码解密流程图

根据上面移位密码解密流程图实现的 C++ 代码如下:

移位解密

```

1 void Shiftencryption()
2 {
3     char Text[1000];
4     int shift = 0;
5     cout << "请输入您要加密的明文"<<endl;
6     cin >> Text;
7     cout << "请输入偏移位数 (右移) :";
8     cin >> shift;
9     shift %= 26;
10    int after = 0;
11    int shift_65 = shift + 65;
12    int shift_90 = shift + 90;
13    int shift_97 = shift + 97;
14    int shift_122 = shift + 122;
15    for (int i = 0; int(Text[i]) != 0; i++)
16    {
17        //计算移位后的ascii
18        after = int(Text[i]) + shift;
19        //判断移位后的
  
```

```

20     if (after >= shift_65 && after <= 90)
21     {
22         Text[i] = char(after);
23     }
24     if (after > 90 && after <= shift_90)
25     {
26         Text[i] = char(after - 26);
27     }
28     if (after >= shift_97 && after <= 122)
29     {
30         Text[i] = char(after);
31     }
32     if (after > 122 && after <= shift_122)
33     {
34         Text[i] = char(after - 26);
35     }
36 }
37 cout << "移位后的密文为: " << endl << Text << endl;
38 cout << "-----" << endl;
39 }

```

对上面加密所得的密文 S vyfo Xkxuks Exsfobcsdi S kw Wskypkcroxq 进行解密, 解密结果如下图所示:

```

请输入你要解密的密文
S vyfo Xkxuks Exsfobcsdi S kw Wskypkcroxq
移1位解密结果: T wzgp Ylyvlt Fytpcdtej T lx Xtlzqldspyr
移2位解密结果: U xahq Zmzwmu Gzuhqdeufk U my Yumarmetqzs
移3位解密结果: V ybir Anaxnv Havirefvgl V nz Zvnbsnfurat
移4位解密结果: W zcjs Bobyow Ibwjsfgwhm W oa Awoctogvsbu
移5位解密结果: X adkt Cpczpx Jcxktghxin X pb Bxpduhwtecv
移6位解密结果: Y belu Dqdaqy Kdyluhiyjo Y qc Cyqevqixudw
移7位解密结果: Z cfmv Erebrz Lezmviyzkp Z rd Dzrfwrjyvex
移8位解密结果: A dgnw Psfcsa Mfanwjkalq A se Easgxskzwfy
移9位解密结果: B ehox Gtgdth Ngboxklbmr B tf Fbthytla xgz
移10位解密结果: C fipy Huheuc Ohcpylmcns C ug Gcuizumbyha
移11位解密结果: D gjqz Ivifvd Pidqzmdot D vh Hdvjavnczib
移12位解密结果: E hkra Jwjgwe Qjeranoepu E wi Iewkbwodajc
移13位解密结果: F ilsb Kxkhxf Rkfsbopfqv F xj Jfxlcxpebkd
移14位解密结果: G jmtc Lyliyg Slgtcpqgrw G yk Kgyndyqfcle
移15位解密结果: H knud Mzmjzh Tmhudqrhsx H zl Lhznezrgdmf
移16位解密结果: I love Nankai University I am Miaofasheng
移17位解密结果: J mpwf Obolbj Vojwfstjuz J bn NjbpGBTifoH
移18位解密结果: K nqXg Pcpmck Wpkxgtukva K co Okcqhcujgpi
移19位解密结果: L oryh QdqndL Xqlyhuvlwb L dp Pldrldvkhqj
移20位解密结果: M pszi Reroem Yrmzivwmx c M eq Qmesjewlirk
移21位解密结果: N qtaj SfspfN Zsnajwxnyd N fr RnftkfxmjSl
移22位解密结果: O rubk Tgtqgo Atobkxyoze O gs Sogulgynktm
移23位解密结果: P svcl Uhurhp Bupelyzpa f P ht Tphvmhzolun
移24位解密结果: Q twdm Vvisiq Cvqdmzaqbg Q iu Uqiwniapmvo
移25位解密结果: R uxen Wjwjtjr Dwrenabrch R jv Vrxojbqnwp
*****

```

图 4: 移位解密演示

对暴力破解得到的 25 个明文进行比对分析, 发现只有移位 16 解得的明文 I love Nankai

University I am Miaofasheng 具有实际意义，故猜测其为明文，解密成功！

### (三) 互相攻击对方用移位密码加密获得的密文

从一位同学得到其加密后的密文为 W qcac tfca Lwblwobu W zcjs bobyow W kobh hc ps ghfcbusf，利用自己所编写的解密程序对其进行暴力破解，破解结果如下：

```

请输入你要解密的密文
W qcac tfca Lwblwobu W zcjs bobyow W kobh hc ps ghfcbusf
移1位解密结果: X rdbt ugdb Mxcmxpcv X adkt cpezpx X lpci id qt higdcvtg
移2位解密结果: Y secu vhec Nydnyqdw Y belu dqdaqy Y mqdj je ru ijhedwuh
移3位解密结果: Z tfdv wifd Ozeozrex Z cfmv erebrz Z nrek kf sv jkifexvi
移4位解密结果: A ugew xjge Pafpasfy A dgnw fsfcsa A osfl lg tw kljgfywj
移5位解密结果: B vhfz ykhf Qbgqbtgz B ehox gtgdtb B ptgm mh ux lmkhgzzk
移6位解密结果: C wigy zlig Rchrcuha C fipy huheuc C quhn ni vy mnlihayl
移7位解密结果: D xjhz amjh Sdisdvib D gjqz ivifvd D rvio oj wz nomjibzm
移8位解密结果: E ykia bnki Tejtewjc E hkra jwlgwe E swjp pk xa opnkjcan
移9位解密结果: F zljb colj Ufkufxkd F ils b kxkhxf F txkq ql yb pqolkdbo
移10位解密结果: G amkc dpmk Vglvgyle G jmtc lyliyg G uylr rm zc qrpmlecp
移11位解密结果: H bnld eqnl Whmwhzmf H knud mzmjzh H vzms sn ad rsqnmfdq
移12位解密结果: I come from Xinxiang I love nankai I want to be stronger
移13位解密结果: J dpnf gspn Yjoyjboh J mpwf obolbj J xbou up cf tuspohfs
移14位解密结果: K eqog htqo Zkpzkpci K nqxs pcpmck K ycpv vq dg uvtqpigf
移15位解密结果: L frph iurp Alqaldqj L oryh qdqdnl L zdqw wr eh vwurqjhu
移16位解密结果: M gsqi jvsq Bmr bmerk M pszi reroem M aerx xs fi wxvsrkiv
移17位解密结果: N htrj kwtr Cnscnfs l N qtaj sfspfn N bfsy yt gj xywtsljw
移18位解密结果: O iusk lxus Dotdogtm O rubk tgtqgo O egtz zu hk yzxutmkx
移19位解密结果: P jvtl myvt Epuephun P svcl uhurhp P dhua av il zayvunly
移20位解密结果: Q kwum nzwu Fqvfvqivo Q twdm viysiq Q eivb bw jm abzwvomz
移21位解密结果: R lxvn oaxv Grwgrjwp R uxen wjwtr R fjwc cx kn bcaxwpna
移22位解密结果: S mywo pbyw Hsxhskxq S vyfo xkxuks S gkxd dy lo cdyxqob
移23位解密结果: T nzxp qczz Itytilyr T wzgp ylyvlt T hlye ez mp deczyrpc
移24位解密结果: U oayq rday Juzjumzs U xahq zmzwmu U imzf fa nq efdazsqd
移25位解密结果: V pbzr sebz Kvakvnat V ybir anaxnv V jnag gb or fgebatre
*****

```

图 5: 破解对方移位加密的密文

通过对所有破解结果进行分析，只有这一条解密结果具有实际意义

移 12 位解密结果: I come from Xinxiang I love nankai I want to be stronger

故推测其明文为 I come from Xinxiang I love nankai I want to be stronger, 密钥  $A=14+26n$  ( $n=0,1,2,3,\dots$ )

### (四) 构建置换表

构建置换表的方法如下：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，为使得结果统一，我们假设置换表中只有大写字母根据上面置换表的构造方法实现的 C++ 代码如下：

构造置换表并打印

```

1 void zhihuanbiao()
2 {
3     char Text[1000];
4     cout << "请输入你的文本" << endl;
5     cin.getline(Text, 1000);
6     bool flag[26] = { 0 };
7     int k = 0;
8
9     for (int i = 0; int(Text[i]) != 0; i++)
10    {

```



```
11     if (int(Text[i]) >= 97 && int(Text[i]) <= 122)
12     {
13         Text[i] -= 32;
14     }
15     if (int(Text[i]) >= 65 && int(Text[i]) <= 90)
16     {
17         int num = int(Text[i]) - 'A';
18         if (flag[num] != true)
19         {
20             result[k] = Text[i];
21             k++;
22         }
23         flag[num] = true;
24     }
25 }
26
27 for (int i = 0; i < 26; i++)
28 {
29     if (flag[i] != true)
30     {
31         result[k] = char(int('A')+i);
32         k++;
33     }
34     flag[i] = true;
35 }
36 for (int i = 0; i < 26; i++)
37 {
38     cout << char('A' + i) << ":" << result[i] << endl;;
39 }
40 }
```

输入一个句子 I love nankai University, 进行构造置换表, 结果如下:

```
请输入你的文本
I love nankai University
A:I
B:L
C:O
D:V
E:E
F:N
G:A
H:K
I:U
J:R
K:S
L:T
M:Y
N:B
O:C
P:D
Q:F
R:G
S:H
T:J
U:M
V:P
W:Q
X:W
Y:X
Z:Z
C:\Users\Calypso\source\repos\Crack\Crack.exe (进程 25548) 已退出。 代码为 0
```

图 6: 破解对方移位加密的密文

得到的置换表为:

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

经检验, 得到的置换表正确无误

## (五) 利用置换表进行加密

置换加密的流程图如下所示:

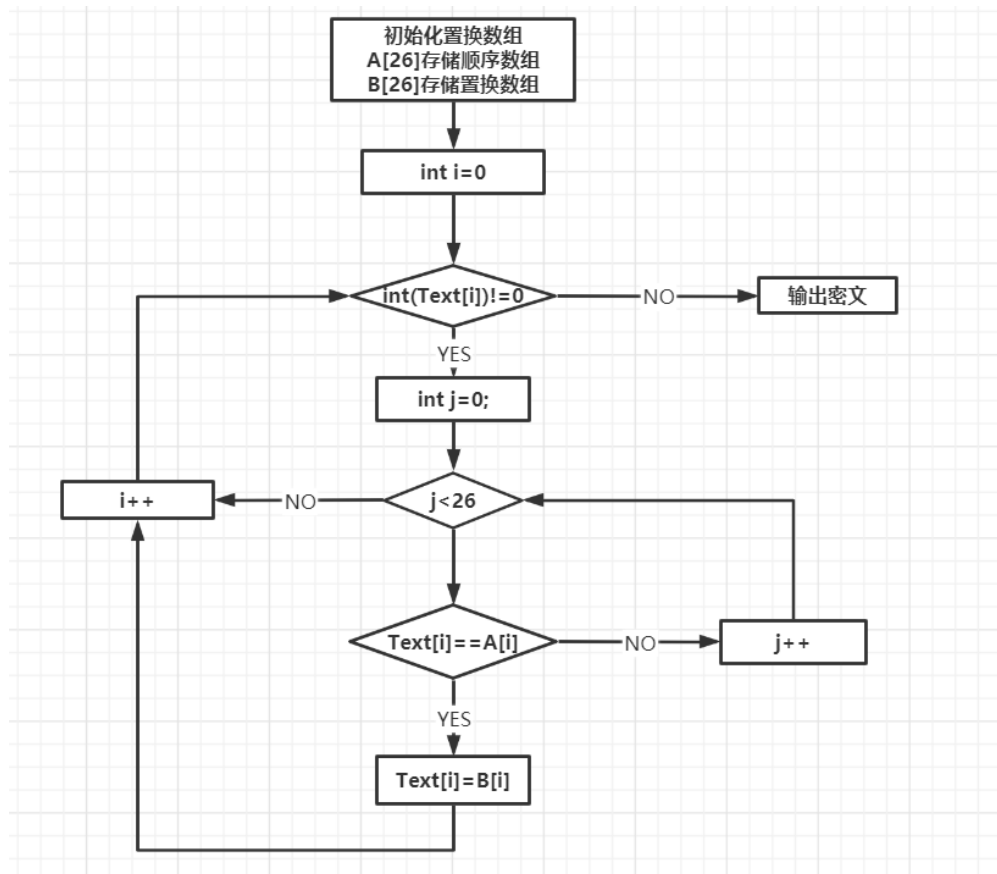


图 7: 置换加密流程图

根据置换加密的流程图，用 C++ 对其进行实现

#### 置换加密

```

1 void ReplaceEncryption(char Text[])
2 {
3     char A[26];
4     for (int i = 0; i < 26; i++)
5     {
6         A[i] = char(65 + i);
7     }
8     char B[26];
9     for (int i = 0; i < 26; i++)
10    {
11        B[i] = result[i];
12    }
13    for (int i = 0; int(Text[i]) != 0; i++)
14    {
15        for (int j = 0; j < 26; j++)
16        {
17            if (Text[i] == A[j])
18            {
19                Text[i] = B[j];

```

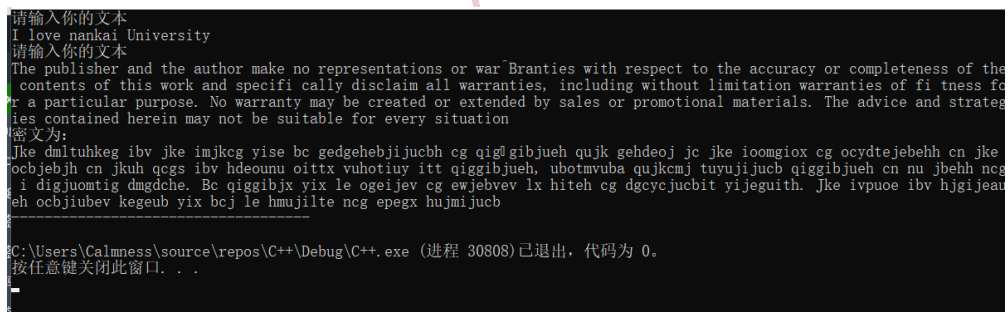
```

20         break;
21     }
22     if (Text[i] == char(int(A[j])+32))
23     {
24         Text[i] = char(int(B[j])+32);
25         break;
26     }
27
28
29
30     }
31 }
32 cout << "密文为: " << endl << Text << endl;
33 cout << "-----" << endl;
34 }

```

输入明文 The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation.

利用上面所得的置换表进行加密, 所得密文如下:



```

请输入你的文本
I love nankai University
请输入你的文本
The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the
contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness fo
r a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strateg
ies contained herein may not be suitable for every situation
密文为:
Jke dmltuhkeg ibv jke imjckg yise bc gedgehebjiucbh cg qiggibjueh qujk gehdeoj jc jke ioomgiox cg ocydtejebhh cn jke
ocbjebjh cn jkuh qcgs ibv hdeounu oittx vuhotiu y itt qiggibjueh, ubotmvuba qujckmj tuyujijucb qiggibjueh cn nu jbehh ncg
i digjuomtig dmgdche. Bc qiggibjx yix le ogeijev cg ewjebvev lx hiteh cg dgcycjucbit yijeguith. Jke ivpuoe ibv hjgijeaueh ocbjiubev
kegeub yix bcj le hmujielte ncg epegx hujmijucb
C:\Users\Calmness\source\repos\C++\Debug\C++.exe (进程 30808)已退出, 代码为 0。
按任意键关闭此窗口。 . . .

```

图 8: 对输入明文进行置换加密

Jke dmltuhkeg ibv jke imjckg yise bc gedgehebjiucbh cg qiggibjueh qujk gehdeoj jc jke ioomgiox cg ocydtejebhh cn jke ocbjebjh cn jkuh qcgs ibv hdeounu oittx vuhotiu y itt qiggibjueh, ubotmvuba qujckmj tuyujijucb qiggibjueh cn nu jbehh ncg i digjuomtig dmgdche. Bc qiggibjx yix le ogeijev cg ewjebvev lx hiteh cg dgcycjucbit yijeguith. Jke ivpuoe ibv hjgijeaueh ocbjiubev kegeub yix bcj le hmujielte ncg epegx hujmijucb

## (六) 利用置换表进行解密

置换加密与置换解密的本质相同, 均为一种替换, 利用所得置换表, 对该密文进行解密, 解密过程只需要将该置换表反过来使用即可, 其流程图如下所示:

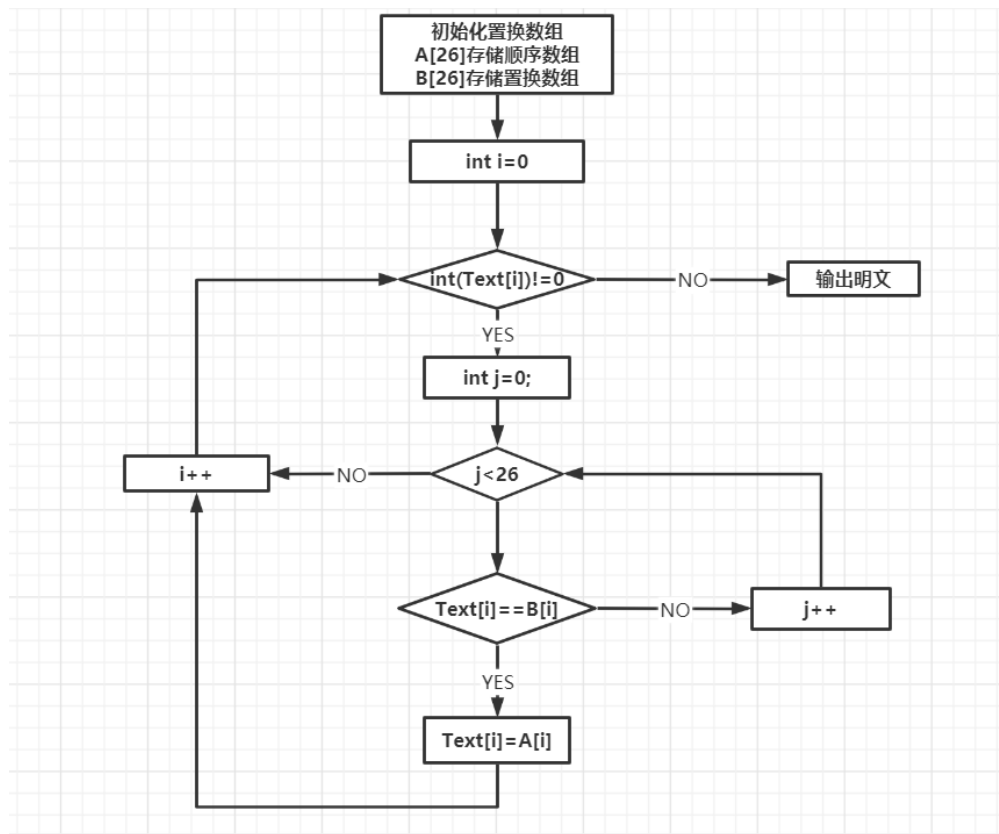


图 9: 置换解密流程图

利用所得置换表, 对该密文进行解密, 解密过程只需要将该置换表反过来使用即可,C++ 代码如下:

#### 置换解密

```

1 void ReplaceDecryption(char Text[])
2 {
3     char A[26];
4     for (int i = 0; i < 26; i++)
5     {
6         A[i] = char(65 + i);
7     }
8     char B[26];
9     for (int i = 0; i < 26; i++)
10    {
11        B[i] = result[i];
12    }
13    for (int i = 0; int(Text[i]) != 0; i++)
14    {
15        for (int j = 0; j < 26; j++)
16        {
17            if (Text[i] == B[j])
18            {
19                Text[i] = A[j];
            }
        }
    }
}
  
```

```

20         break;
21     }
22     if (Text[i] == char(int(B[j]) + 32))
23     {
24         Text[i] = char(int(A[j]) + 32);
25         break;
26     }
27 }
28 }
29 cout << "明文为: " << endl << Text << endl;
30 cout << "-----" << endl;
31 }

```

```

请输入你的文本
I love nankai University
请输入你的文本
Jke dmltuhkeg ibv jke imjkg yise bc gedgehebjjucbh cg qig'Bgibjueh qujk gehdeoj jc jke ioomgiox cg ocydtejebehh cn jke
ocbjebjh cn jkuh qcgs ibv hdeounu oittx vuhotiuy itt qiggibjueh, ubotmruba qujkmj tuyujijucb qiggibjueh cn nu jbehhh nc
g i digjuomtig dmgdche. Bc qiggibjx yix le ogeijev cg ewjebvev lx hiteh cg dgcyecjucbit yijeguith. Jke ivpuoe ibv hjgijsa
ueh ocbjiubev kegeub yix bcj le hmujihte ncg epegx hujmijucb
明文为:
The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the
contents of this work and specifi cally disclaim all warranties, including without limitation warranties of fi tness for
a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategi
es contained herein may not be suitable for every situation
-----
C:\Users\Calmness\source\repos\C++\Debug\C++.exe (进程 6600) 已退出, 代码为 0。
按任意键关闭此窗口。 . . .

```

图 10: 对所得密文进行置换解密

解密结果为: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifi cally disclaim all warranties, including without limitation warranties of fi tness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation.

解密正确!

## (七) 利用频率统计方法破译密文

要破译的密文为: SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPN-BRQJSSJBE JBFMPQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNRB MF N XMRRJHAY JBRGZPC 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 ZBNZSIM-PJOCD GMBSPMA MF SIC QCRRNEC

### 1. 前置知识

**短单词 (small words):** 在英文中只有很少几个非常短的单词。因此, 如果在一个加密的文本中可以确定单词的范围, 那么就能得出明显的结果。一个字母的单词只有 a 和 I。如果不计单词的缩写, 在从电子邮件中选取 500k 字节的样本中, 只有两个字母的单词仅出现 35 次, 而两个

字母的所有组合为  $26 \times 26 = 676$  种。而且，还是在那个样本中，只有三个字母的单词出现 196 次，而三个字母的所有组合为  $26 \times 26 \times 26 = 17576$  种。

**常用单词 (common words):** 再次分析 500k 字节的样本，总共有 5000 多个不同的单词出现。在这里，9 个最常用的单词出现的总次数占总单词数的 21%，20 个最常用的单词出现的总次数占总单词数的 30%，104 个最常用的单词占 50%，247 个最常用的单词占 60%。样本中最常用的 9 个单词占总词数的百分比为：

the 4.65 to 3.02 of 2.61 I 2.2 a 1.95 and 1.82 is 1.68 that 1.62 in 1.57

**字母频率 (character frequency):** 在 1M 字节旧的电子文本中，对字母“A”到“Z”（忽略大小写）分别进行统计。发现近似频率（以百分比表示）：

e 11.67 t 9.53 o 8.22 i 7.81 a 7.73 n 6.71 s 6.55 r 5.97 h 4.52 l 4.3 d 3.24 u 3.21 c 3.06 m 2.8 p 2.34 y 2.22 f 2.14 g 2.00 w 1.69 b 1.58 v 1.03 k 0.79 x 0.30 j 0.23 q 0.12 z 0.09

## 2. 统计密文中字母出现的次数

将密文串进行遍历，遇到对应的字母，则字母数 ++，实现的 C++ 代码如下：

统计字母次数

```

1 void ReplaceDecryption(char Text[])
2 {
3     char A[26];
4     for (int i = 0; i < 26; i++)
5     {
6         A[i] = char(65 + i);
7     }
8     char B[26];
9     for (int i = 0; i < 26; i++)
10    {
11        B[i] = result[i];
12    }
13    for (int i = 0; int(Text[i]) != 0; i++)
14    {
15        for (int j = 0; j < 26; j++)
16        {
17            if (Text[i] == B[j])
18            {
19                Text[i] = A[j];
20                break;
21            }
22            if (Text[i] == char(int(B[j]) + 32))
23            {
24                Text[i] = char(int(A[j]) + 32);
25                break;
26            }
27        }
28    }
29    cout << "明文为: " << endl << Text << endl;
30    cout << "-----" << endl;
31 }

```

统计结果如下：

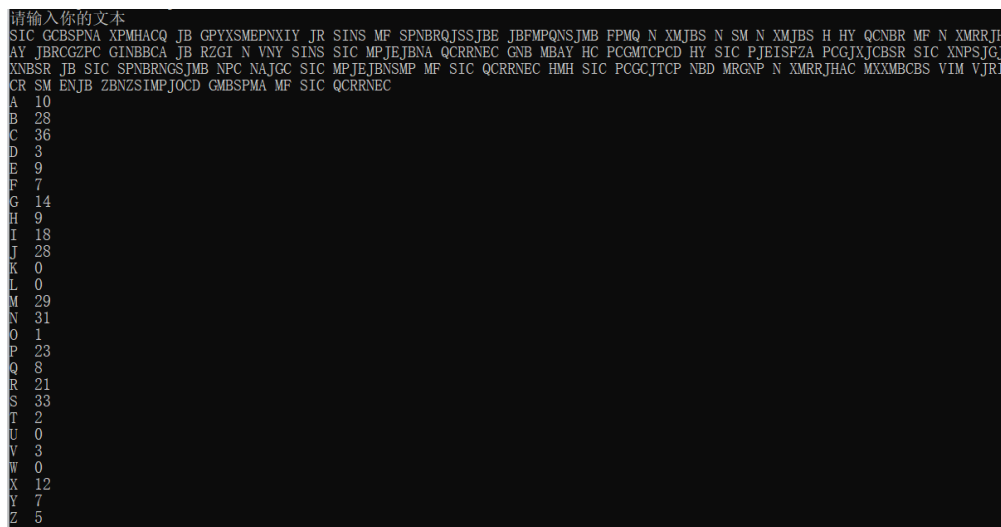


图 11: 密文中字母出现的次数

### 3. 第一步

我们可知，字母 C 出现的次数最多，36 次，字母 S 出现的次数第二多，33 次，字母 N 出现的次数第三多，31 次，依据现有文本统计的字母频率，我们可以先假定 C 对应 E，S 对应 T（如果不对，后期进行推到再进行替换），暂时可以得到如下替换表：

原字母	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
对应字母			E																T							

图 12: 替换表

利用暂时得到的替换表，对密文进行替换，替换结果如下：

TIE GEBTPNA XPMHAEQ JB GPYXTMEPNXIY JR TINT MF TPNBRQJTTJBE  
JBFMPQNTJMB FPMQ N XMJBT N TM N XMJBT H HY QENBR MF N XMRRJHAY  
JBREGZPE GINBBEA JB RZGI N VNY TINT TIE MPJEJBNA QERRNEE GNB MBAY  
HE PEGMTEPED HY TIE PJEITFZA PEGJXJEBTR TIE XNPTJGJXNBTR JB TIE TPN-  
BRNGTJMB NPE NAJGE TIE MPJEJBNTMP MF TIE QERRNEE HMH TIE PEGEJTEP  
NBD MRGNP N XMRRJHAE MXXMBEBT VIM VJRIER TM ENJB ZBNZTIMPJOED  
GMBTPMA MF TIE QERRNEE

现在密文中出现三个字母的单词“TIE”，根据短单词分析，我们猜测 I 对应 E，观察发现，密文中单字母单词有 6 个 N，一个 H，根据现有短单词的经验，一般长句子中容易出现 a 和 an 等冠词，我们猜测，N 对应 A，H 对应 I，得到现有替换表如下：

原字母	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
对应字母			E					I	H					A					T							

图 13: 替换表



#### 4. 第二步

根据现有的替换表，我们先对密文进行替换，得到如下结果：

```
请输入你的文本
SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPNBRQJSSJBE JBFMPQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJI
AY JBREGZPC GINBBCA JB RZGI N VNY SINS SIC MPJEJBNA QRRNEC GNB MBAY HC PCGMCPCD HY SIC PJEISFZA PCGJXJCSR SIC XNPSJGJ
XNBSR JB SIC SPNBRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF SIC QRRNEC HHM SIC PCGCJTCP NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRI
CR SM ENJB ZBNZSIMPJOCD GMBSPMA MF SIC QRRNEC
替换后的结果
THE GEBTPAA XPMIAEQ JB GPYXTMEPAXHY JR THAT MF TPABRQJTTJBE JBFMPQATJMB FPMQ A XMJBT A TM A XMJBT I IY QEABR MF A XMRRJI
AY JBREGZPE GHABBEA JB RZGH A VAY THAT THE MPJEJBAA QERRAEE GAB MBAY IE PEGMTEPED IY THE PJEHTFZA PEGJXJEBTR THE XAPTJGJ
XABTR JB THE TPABRAGTJMB APE AAJGE THE MPJEJBATMP MF THE QERRAEE IMI THE PEGEJTEP ABD MRGAP A XMRRJIAE MXXMBEET VHM VJRHER
ER TM EAJB ZBAZTHMPJOED GMBTPMA MF THE QERRAEE
C:\Users\Calmness\source\repos\C++\Debug\C++.exe (进程 28776) 已退出，代码为 0。
按任意键关闭此窗口。...
```

图 14: 密文替换结果

结果为: THE GEBTPAA XPMIAEQ JB GPYXTMEPAXHY JR THAT MF TPABRQJTTJBE JBFMPQATJMB FPMQ A XMJBT A TM A XMJBT I IY QEABR MF A XMRRJIAY JBREGZPE GHABBEA JB RZGH A VAY THAT THE MPJEJBAA QERRAEE GAB MBAY IE PEGMTEPED IY THE PJEHTFZA PEGJXJEBTR THE XAPTJGJXABTR JB THE TPABRAGTJMB APE AAJGE THE MPJEJBATMP MF THE QERRAEE IMI THE PEGEJTEP ABD MRGAP A XMRRJIAE MXXMBEET VHM VJRHER TM EAJB ZBAZTHMPJOED GMBTPMA MF THE QERRAEE

#### 5. 第三步

我们统计密文中两个字母的短单词，共出现 13 次，分别为 JB JR MF TM IY MF JB IE IY JB MF TM MF

MF 出现 4 次，JB 出现 3 次，TM 出现 2 次，IY 出现 2 次，JR 和 IE 分别出现 1 次

按照频率，因为 T 已经确定，所以根据 to, of, is, in 这几个单词，将 M 确定为 O，F 确定为 F

原字母	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
对应字母			E			F		I	H				O	A					T							

图 15: 替换表

#### 6. 第四步

根据第三步所得替换表，对密文进行替换，得到结果如下：

```
Microsoft Visual Studio 调试控制台
请输入你的文本
SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPNBRQJSSJBE JBFMPQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJI
AY JBREGZPC GINBBCA JB RZGI N VNY SINS SIC MPJEJBNA QRRNEC GNB MBAY HC PCGMCPCD HY SIC PJEISFZA PCGJXJCSR SIC XNPSJGJ
XNBSR JB SIC SPNBRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF SIC QRRNEC HHM SIC PCGCJTCP NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRI
CR SM ENJB ZBNZSIMPJOCD GMBSPMA MF SIC QRRNEC
替换后的结果
THE GEBTPAA XPOIAEQ JB GPYXTOEPAXHY JR THAT OF TPABRQJTTJBE JBFOPQATJOB FPOQ A XOJBT A TO A XOJBT I IY QEABR OF A XORRJI
AY JBREGZPE GHABBEA JB RZGH A VAY THAT THE OPJEJBAA QERRAEE GAB OBAY IE PEGOTEPED IY THE PJEHTFZA PEGJXJEBTR THE XAPTJGJ
XABTR JB THE TPABRAGTJOB APE AAJGE THE OPJEJBATOP OF THE QERRAEE IOI THE PEGEJTEP ABD ORGAP A XORRJIAE OXXOBEET VHO VJRHER
ER TO EAJB ZBAZTHOPJOED GOBTOA OF THE QERRAEE
C:\Users\Calmness\source\repos\C++\Debug\C++.exe (进程 7644) 已退出，代码为 0。
按任意键关闭此窗口。...
```

图 16: 根据现有替换表进行替换

得到的结果为：

THE GEBTPAA XPOIAEQ JB GPYXTOEPAXHY JR THAT OF TPABRQJTTJBE JB-FOPQATJOB FPOQ A XOJBT A TO A XOJBT I IY QEABR OF A XORRJIAY JBREGZPE GHABBEA JB RZGH A VAY THAT THE OPJEJBAA QERRAEE GAB OBAY IE PEGOTE-PED IY THE PJEHTFZA PEGJXJEBTR THE XAPTJGJXABTR JB THE TPABRAGTJOB APE AAJGE THE OPJEJBATOP OF THE QERRAEE IOI THE PEGEJTEP ABD ORGAP A XORRJIAE OXXOEBT VHO VJRHER TO EAJB ZBAZTHOPJOED GOBTPOA OF THE QERRAEE

观察到短单词 FPOQ, 由于 F 和 O 的对应关系已经确定, 我们猜测这个短单词的原文是 FRPM, 故我们猜测 P 对应 R, Q 对应 M

得到现有的替换表如下:

原字母	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
对应字母			E			F		I	H				O	A		R	M		T							

图 17: 替换表

## 7. 第五步

依据现有替换表, 替换后所得的密文串如下:

THE GEBTRAA XROIAEM JB GRYXTOERAXHY JR THAT OF TRABRMJTTJBE JBFORMATJOB FROM A XOJBT A TO A XOJBT I IY MEABR OF A XORRJIAY JBREGZRE GHABBEA JB RZGH A VAY THAT THE ORJEJBAA MERRAEE GAB OBAY IE REGOTERED IY THE RJEHTFZA REGJXJEBTR THE XARTJGJXABTR JB THE TRABRAGTJOB ARE AAJGE THE ORJEJBATOR OF THE MERRAEE IOI THE REGEJTER ABD ORGAR A XORRJIAE OXXOEBT VHO VJRHER TO EAJB ZBAZTHORJOED GOBTROA OF THE MERRAEE

观察到短单词 VHO, 因为 H 和 O 已经确定, 故我们猜测 VHO 对应单词 WHO, V 对应的是 W, 依据此进行替换, 替换后的结果为

THE GEBTRAA XROIAEM JB GRYXTOERAXHY JR THAT OF TRABRMJTTJBE JBFORMATJOB FROM A XOJBT A TO A XOJBT I IY MEABR OF A XORRJIAY JBREGZRE GHABBEA JB RZGH A WAY THAT THE ORJEJBAA MERRAEE GAB OBAY IE REGOTERED IY THE RJEHTFZA REGJXJEBTR THE XARTJGJXABTR JB THE TRABRAGTJOB ARE AAJGE THE ORJEJBATOR OF THE MERRAEE IOI THE REGEJTER ABD ORGAR A XORRJIAE OXXOEBT WHO WJRHER TO EAJB ZBAZTHORJOED GOBTROA OF THE MERRAEE

观察到短单词 JR, 而 R 已经确定, 故猜测 J 为 O, 对其进行替换, 替换后的结果如下:

THE GEBTRAA XROIAEM OB GRYXTOERAXHY OR THAT OF TRABRMOTTOBE OBFORMATOEB FROM A XOOBT A TO A XOOBT I IY MEABR OF A XORROIAY OBREGZRE GHABBEA OB RZGH A WAY THAT THE OROEOBAA MERRAEE GAB OBAY IE REGOTERED IY THE ROEHTFZA REGOXOEBTR THE XARTOGOXABTR OB THE TRABRAGTOEB ARE AAOGE THE OROEOBATOR OF THE MERRAEE IOI THE REGEOTER ABD ORGAR A XORROIAE OXXOEBT WHO WORHER TO EAOB ZBAZTHOROOED GOBTROA OF THE MERRAEE

观察到单词 MERRAEE, 而 M, E, A 已经确定, 故我们猜测该单词是 MESSAGE, E 对应 G, R 对应 S, 依据此进行替换, 替换后的结果为

THE GEBTRAA XROIAEM IB GRYXTOGRAXHY IS THAT OF TRABSMITTIBG IBFORMATIOB FROM A XOIBT A TO A XOIBT I IY MEABS OF A XOSSIAY IBSEGZRE GHABBEA IB SZGH A WAY THAT THE ORIGIBAA MESSAGE GAB OBAY IE REGOTERED IY THE RIGHTFZA REGIXIEBTS THE XARTIGIXABTS IB THE TRAB-SAGTIOB ARE AAIGE THE ORIGIBATOR OF THE MESSAGE IOI THE REGEITER ABD OSGAR A XOSSIIE OXXOBEBT WHO WISHES TO GAIB ZBAZTHORIOED GOBTROA OF THE MESSAGE

观察现阶段得到的密文, 我们发现词组 SEG H A WAY, 由于 A, WAY, S 以及 H 都已经确定, 我们推测 Z 为 U, G 为 C, 该词组为 SUCH A WAY, 以及次进行推断, 推断结果如下

THE CEBTRAA XROIAEM IB CRYXTOGRAXHY IS THAT OF TRABSMITTIBG IBFORMATIOB FROM A XOIBT A TO A XOIBT I IY MEABS OF A XOSSIAY IBSECURE CHABBEA IB SUCH A WAY THAT THE ORIGIBAA MESSAGE CAB OBAY IE RECOTERED IY THE RIGHTFUA RECIXIEBTS THE XARTICIXABTS IB THE TRABSACTION ARE AAICE THE ORIGIBATOR OF THE MESSAGE IOI THE RECEITER ABD OSCAR A XOSSIIE OXXOBEBT WHO WISHES TO GAIB UBAUTHORIOED COBTROA OF THE MESSAGE

我们根据 IB SUCH A WAY 以及 ABD 在这个词组中除了 B 其他都已经确定, 故我们推断 B 为 N, 以及此进行替换, 得到如下结果

THE CENTRAA XROIAEM IN CRYXTOGRAXHY IS THAT OF TRANSMITTING INFORMATION FROM A XOINT A TO A XOINT I IY MEANS OF A XOSSIAY INSECURE CHANNEA IN SUCH A WAY THAT THE ORIGINAA MESSAGE CAN ONAY IE RECOTERED IY THE RIGHTFUA RECIXIENTS THE XARTICIXANTS IN THE TRANSACTION ARE AAICE THE ORIGINATOR OF THE MESSAGE IOI THE RECEITER AND OSCAR A XOSSIIE OXXONENT WHO WISHES TO GAIN UNAUTHORIOED CONTROA OF THE MESSAGE

我们发现现在的密文中 CENTRAA, 而除了最后一个 A 外, 所有字母都为已经确定的字母, 我们猜测该单词为 CENTRAL, 故推断 A 为 L, 以此替换可得如下结果:

THE CENTRAL XROILEM IN CRYXTOGRAXHY IS THAT OF TRANSMITTING INFORMATION FROM A XOINT A TO A XOINT I IY MEANS OF A XOSSIILY INSECURE CHANNEL IN SUCH A WAY THAT THE ORIGINAL MESSAGE CAN ONLY IE RECOTERED IY THE RIGHTFUL RECIXIENTS THE XARTICIXANTS IN THE TRANSACTION ARE ALICE THE ORIGINATOR OF THE MESSAGE IOI THE RECEITER AND OSCAR A XOSSIIE OXXONENT WHO WISHES TO GAIN UNAUTHORIOED CONTROL OF THE MESSAGE

我们根据 CRYXTOGRAXHY, 该单词中只有 X 是未确定的, 我们猜测该单词的明文是 CRYPTOGRAPHY (密码学), X 对应 P, 以此替换可得如下结果:

THE CENTRAL PROILEM IN CRYPTOGRAPHY IS THAT OF TRANSMITTING INFORMATION FROM A POINT A TO A POINT I IY MEANS OF A POSSIILY INSECURE CHANNEL IN SUCH A WAY THAT THE ORIGINAL MESSAGE CAN ONLY IE RECOTERED IY THE RIGHTFUL RECIPIENTS THE PARTICIPANTS IN THE TRANSACTION ARE ALICE THE ORIGINATOR OF THE MESSAGE IOI THE RECEITER AND OSCAR A POSSIIE OPPONENT WHO WISHES TO GAIN UNAUTHORIOED CONTROL OF THE MESSAGE

现在的密文基本已经可读了, 我们可以轻易的推断出 PROILEM 是 PROBLEM, 其中 I 对

应 B, 而 I 已经被 H 所对应过, 故之前的推断错误, H 应该对应 B, 以此进行替换可得如下结果:

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

我们依据 RECOVERED, 由于除 T 外, 所有字母均已确定, 我们猜测该单词是 RECOVERED, T 对应 D, 以此推断可得如下结果:

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

我们依据 RECEIVER, 虽然该单词中所有单词均以有对应, 但是该单词仍旧没有实际意义, 故猜测之前的推断有错误, 将 T 改为对应 V, 可得 RECEIVER, 以此推断可得如下结果:

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

通读利用现在的替换表解密得到的文字, 只有 UNAUTHORIZED 没有实际意义, 我们猜测该单词实际为 UNAUTHORIZED, O 对应 Z, 依次进行推断, 可得如下结果:

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

此时解密后的文字已经全部具有了实际意义, 整个段落也非常有逻辑, 故我们猜测解密成功!

## 8. 总结

最终得到的置换表如下:

原字母	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
对应字母	L	N	E	V	G	F	C	B	H	I	*	*	O	A	Z	R	M	S	T	V	*	W	*	P	Y	U

图 18: 最终得到的替换表

由于 K,L,U,W 四个字母在密文中根本没有出现, 我们也无法推断其对应关系, 依据已经推出的 22 对对应关系, 我们已经可以成功破解!

成功破解得到的明文为:

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

破解该单表代换的 C++ 代码如下:

破译所用代码

```
1  for (int i = 0; int(Text[i]) != 0; i++)
2      {
3          if (Text[i] == 'C')
4              cout << 'E';
5          else if (Text[i] == 'S')
6              cout << 'T';
7          else if (Text[i] == 'H')
8              cout << 'B';
9          else if (Text[i] == 'I')
10             cout << 'H';
11          else if (Text[i] == 'N')
12             cout << 'A';
13          else if (Text[i] == 'M')
14             cout << 'O';
15          else if (Text[i] == 'P')
16             cout << 'R';
17          else if (Text[i] == 'Q')
18             cout << 'M';
19          else if (Text[i] == 'V')
20             cout << 'W';
21          else if (Text[i] == 'J')
22             cout << 'I';
23          else if (Text[i] == 'E')
24             cout << 'G';
25          else if (Text[i] == 'R')
26             cout << 'S';
27          else if (Text[i] == 'Z')
28             cout << 'U';
29          else if (Text[i] == 'G')
30             cout << 'C';
31          else if (Text[i] == 'B')
32             cout << 'N';
33          else if (Text[i] == 'A')
34             cout << 'L';
```

```
35     else if (Text[i] == 'X')
36         cout << 'P';
37     else if (Text[i] == 'T')
38         cout << 'V';
39     else if (Text[i] == 'O')
40         cout << 'Z';
41     else
42         cout << Text[i];
43 }
```

NIKU