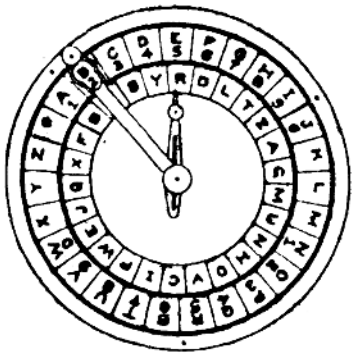
古典代替密码体制的计算机模拟

R·F· 劳厄著

齐忠涛 刘力农 译 关义章 校



通信保密编辑部

原　序

本书是为学密码编码学的高年级大学生写的；具体说来，它是一本工具书，而不是历史述评。鼓励读者积极使用本书，而不仅仅读一读而已。如果读者手边没有微处理机，那末他应该设法弄一台。现在，小型高效的家庭计算机相当便宜，至少比前几年便宜；而且他们学习密码学的新领域。应该编写、运行和修改程序。应该解题而且应该试着编写新的密码分析的程序，解题的新途径有你们去努力探索。于是读者要记住，这是一本未完结的书，是一本激励人们在一个激发的未开发的领域里进一步工作的教科书。读者将发现，用来模拟大多数古典代替编码体制的原则，一般说，是一再用过的标准，尽管它常根据模拟的体制以修正的形式出现。但愿你发展的密码分析的方法也能在许多的其它密码体制中找到应用。

本书假定读者熟悉包括术语和统计试验在内的密码编码学内容。如果你还不熟悉用BASIC编写程序，那末本书恰好是推动你朝现代密码编码学和密码分析学的新水平前进的动力。

译者的话

七十年代后期，西方的一些密码经典著作解密后，美国开始出版一套密码丛书。本书是八一年出版的，该丛书的第32本也是目前见到的内容最新的一本书。

本书的目的是引导读者用计算机来模拟和分析各种密码体制。书中提供了大量的用BASIC语言编写的模拟和分析程序，可供读者移植和参考。本书对各种古典代替密码体制做了历史的评述，提供了资料的查找线索和可供进一步研究的问题。正如著者所指出的“是一本未完结的书”。阅读本书必须具备一定的密码和计算机的知识。

当今的社会是信息社会。研究传输中信息保密的密码学已超越军事、政治和外交的范围，正蓬勃发展。这形势已引起了许多实业家和学者的关注。我们把劳厄的这本书译出来。为关心这方面情况的读者提供一些资料。

由于我们水平有限，翻译中不妥之处，谨请批评指正。

译者一九八四年四月于北京

目　录

引　言

几乎所有认真的密码编码学研究者都希望有一天能拥有自己专用的密码机，作者本人也不例外，设计出非同时代其它密码机可比拟的新机器，从而在密码史上占有独特的一页，当然是非常值得自豪的。然而，对一个体制进行体制攻击的研究和发展之前，显然应先具有能产生所需密码序列的设备。尽管我在这方面花费了很多的时间来为我自己装置这些设备，但最终只获得了情况比较好的哈格林密码（M—209）。权衡利弊，我认为在微计算机上对密码设备进行模拟比起装置那些实际设备要好。理由如下：

1、易录：除哈格林密码（M—209）外，其它设备没有一个是容易得到的。主要原因是生产的数量较少或由于政府的限制。例如。上次战争中的恩尼格马机至今仍未找到。

2、经济：除开始建立微计算机系统的价格外。如果只做计划中的应用，则不再需要更多的付加费用。反之，购置一套单独的设备（假定你能在市场上找到）的付加开支会接连不断。现在，一套典型的微计算机系统大约价值1000美元，并且还会降低。而密码设备的价格，如果你能买到，则会很高，并不断上涨。

3、打印结果：许多情况下，密码设备只以某种形式指示字母，需要操作者先读出结果，再用手抄下来。这就存在着两个可能出错的环节。而计算机驱动打印机输出，就没有这个问题。

4、操作简便：记入或更换“密钥”，在计算机程序中比在一些密码设备中要相对简单得多。用计算机时操，作者只要先键入“Key”或“Reys”，然后就可输入相应的信息，最后回车等待打印结果。而这在许多密码设备中相当于移动“滑尺”，用时要特别注意它们字母表间的相对移动，由此也增加了出错的可能。

5、响应迅速：在模拟密码设备时，计算机打印结果（输出）通常只要几分钟，有时只要几秒钟。另一方面，实际设备在时间上却有很大的耽搁和浪费。例如，一个人能设想用惠斯通设备在10分钟内写500个字母，并没有错误吗？

6、容易改进：在利用计算机的情况下，简单的程序改动常常只要几分钟便可完成。反之，复杂的齿轮系统的改动常要几天或几个星期才能实现。因此，使用计算机能使发明者”立即看到由于某一改动而产生的结果。此外，新系统的设计或系统彻底地改变常常用很短的时间就可完成，并且其加密结果的保密程度也可以被迅速地检验。没有计算机系统，要设计每个棒上有两个以上焊片的哈格林（M－209）密码机要多少时间？

正文共十章，文中不仅仅是阐述了对各种特殊设备的模拟方法，而且还剖析了各类密码机的原理。因此，用于一个机器的方法也可同样用于它同类的机器。例如，除克吕阿（KRYHA）密码机外，两个字母表不规则移位的原理是早期密码打字机的共同特点。各章的顺序是按由简单到复杂的代替体制安排的，在一些例子中，虽然某些体制可认为是别种体制的变型，但这区别是根据一般解法提出的。

正如标题所指明的，文中讨论的所有体制都属代替类型。这主要是因为只有三个美国专利是移位体制的(156851，13111457，1370870)。并且早期机械的或电子的设备没有有效的方法来“存贮”字母。在考察美国专利和密码学历史文献两方面的基础上，选

定了（用微计算机进行模拟的）机器。大多数情况下，这些机器是在实际中使用过的。也有几个例子，仅在专利权中说明了设备的操作方法，因而不知道实际机器是否被制造和使用过。因为早期的设备或是机械的。或是电动机械的，所以本书中没有讨论现代称为“计算机”的体制。希望能有一系列后继的书籍来填充这一重要的知识领域。由于计算机的模2操作，弗纳姆(Vernam)体制有意地省略了。还有待于类似的文章去阐述。

每一章由五部分组成：基本情况、程序、密码分析、致读者和问题。

基本情况包括设备及其工作方式的简单说明，为进一步研究提供参考，并列出了该机器和其它原理相同的机器的美国专利简表。多数情况下，这些专利的分类仅是根据它们的说明，还有很大讨论的余地。

程序部分主要是叙述程序工作方式和在不同时刻输入的正确方法。没有给出仔细的程序分析。但对一些特殊点地行了讨论，如特定的变量或数据修改的范围。在每种情况下，都给出了例子及其密钥以便检验程序工作方式。

密码分析部分尽量地短而精，因为这一领域不是本著作的重点所在。为使读者能够学会写分析程序，各章都列出了参考资料以供进一步研究。这部分也提出了一些具有普遍性，可适用于多种设备的密码分析程序。程序的工作方式也进行了全面地解释。

致读者是对所涉及的设备进行各式各样的改进的简短的汇集，同时给出了其它程序的思路。所提到的修改或改进决不是仅有的可能。它只是提供一个出发点，使读者能为各自的模拟和密码分析而进行进一步的试验。

问题的数量很少，主要目的是激励读者去做更多的工作。同时，有些问题可利用文中已给出的密码分析程序来解决，其它的则需要编写新的程序。有些问题给了提示，所有的问题都可由文中所给出的程序生成。顺带说明，明文是非常简单的内容。

所提供的程序是为在具有8K内存的Commodore PET微计算机上运行的。用Commodore 2023型针式打印机列出了程序清单。它对旧式和新式的ROM可兼容。程序都能在8K内存中运行。而且，经过一些小的改进后，程序也可在任何其它微机系统内运行。其工作方式仅就PET机进行了说明。如在使用中出现问题，可参考D.A.Lien所著的《The Basic Handbook》。语句OPEN 1,4和CLOSE 1。如有用其它命令者，也可以删改。当打印语句仅是为了在监视屏上显示时，应写PRINT #1。所有这些，必要时都可加以改变。

由于作者的习惯，程序中没有使用REM语句。每行程序可有一条以上的语句，中间用冒号分隔。所有没用的空格都省略了。尽管这样会使程序不太容易读，但对有经验的程序员来说不会有什么困难。模块化的程序结构也使细心的读者容易了解其工作情况。

虽然花费了很多时间来寻找和纠正程序中的错误，但还可能有些缺陷和错误未被发现。如读者发现了实际存在的错误，恳请通知作者。

如欲获得专利的拷贝，每份专利可寄50美分到华盛顿D.C.20231，美国商业部专利及商标办公室。

最后，尽管作者在前面曾提到对密码机进行模拟（一般来说）比拥有自己的实际密码机要好，但作者不想留下这样的印象，好象他不再寻找密码机了。而实际是，作者不

仅仍在寻找实际的密码设备，而且准备以适当的价格购买。

R.F.劳厄

程　序　清　單

第一章　简单代替(Simple Substitution)

基本情况

简单代替加密体制是最基本的加密体制。明文字母表中的每一个字母，都有一个或多个密文等价物(Cipher equivalents)与之对应。密文等价物可以是字母、字母组、数字或任意的符号，最普通的是明文字母表中的每一个字母唯一地对应密文字母表中的一个字母。考察这些字母表的基本分量[1](Primary components)，有下列七种情形：

1．两表的基本分量都是自然顺序(normal sequences)[2]并且两表的排序方向相同，

2．两表的基本分量都是自然顺序排列，但它们的排序方向相反。

3．明文分量是自然顺序，密文分量是乱序序列。

4．密文分量是自然顺序，明文分量是乱序序列。

5．两表的分量是相同的乱序序列，并且排列方向相同，

6．两表的分量是相同的乱序序列，但排列方向相反。

7．分量是不同的乱序序列。

如果密文中保留着明文的单词长度，我们就得到熟悉的报纸型密码(Aristocrats)。当采用标准军用方法时就没有单词间的空格并且密文都分为五字母组(Patristocrats)。上述第1种情形中的密表常被称为“凯撒密表”（“Caesar” alphabet），这种加密体制也被称为“凯撒密码”。这种密码也可能会有各种各样的改进和变化，但这里仅对上述七种情形进行讨论［2—16］。

美国专利档案中记载着许多根据简单代替原理设计的设备。它们中有同轴转盘、滑动条、图表、特制打印设备，还有配在老式打印机上的附件。这些各种各样的设备中比较有代表性的几种是：

发明者	专利号
R•Harte	527112
E•H•Hebern & F•Hoffman	1086823
E•H•Hebern & F•Hoffman	1123738
E•H•Hebern	1141055
S•M．Kintner	1210656
J．P．Griffiths	1389559
S．H．Huntington	150077

①分量的意义是指：可以将简单代替体制中的明文和密文字母表看成两个向量。英文共26个字母，则两表同为26维向量，表中每一个字母都是这个向量的一个分量。—译者注。

②自然顺序是指按英文字母表中的字母顺序排列，即A，B，C.D，…Y，Z。而乱序序列则是指以任意的方式排列的字母表。后文相同，不再注。—译者注。

L. A. Nemcovsky	1562120
J. P. Pessoa	1564268
F. Prevost	1570178

进行简单代替最常用的方法是密码园盘［17］。这种体制之所以能保密是因为对26个字母的字母表来说有26! (4. 03×10^{26})种可能的不同字母排列顺序。

程序

不论加密或脱密过程，所有的简单代替设备都可以由程序SIMPLESUB(表1)进行模拟。

程序第100行中，PRINT "reverse heart" (1)是清除屏上当前显示内容并使光标回到屏左上角。

第300—330行将报文以字符串(M$)形式输入。根据这种BASIC版本，一个字符串最多可允许256个字符(包括空格)，因此，对更长的报文必须分段输入。

第310行中用来建立M$的GET命令是唯一的，引号中奇特的符号是为建立一个"人为的"闪烁光标而用的。它们是"reverse field on－space－reverse field off－cursor left left"和space－cursor left"。如不需要光标，则这一行可缩减到只有GET命令。本书中几乎所有程序都采用的是这种输入方法。

WITH WORD SPACINGS? (2)(Y或N) 如输入N，将得到五个字母一组的输出。

PLAIN COMPONENT? 输入任意顺序的明文字母表。如输入的字母数不正确，你会得到错误消息。

CIPHER COMPONENT? 输入密文字母表。在按RETURN(回车)键前，错误处应予更正。

TEXT: 输入消息。不论是明文，或是密文均可。注意：输入了一个字母之后，就不能再更改了。输入报文的长度最多为256个字母。当输入的是明文时，如在输出的密文中不存在字间的空格，则可于输入时略去空格以节省内存容量。记住，空格本身也是一个符号。

ENCIPHER OR DECIPHER (E或D)

CONTINUE? (Y或N) 如为长报文分段输入情况，则于此处输入Y。

CONTINUE TEXT? (Y或N)

NEW TEXT? (Y或N)

CHANGE CIPHER COMPONENT? (Y或N) 此时明文字母表并不改变。

CHANGE PLAN COMPONENT? (Y或N) 此时密文字母表应重新输入。

---

① 这里的"reverse heart"和下文中的reverse field on－space－reverse field off－Cursor left"和Space－cursor－left是指程序第100行和310行中PRINT后引号里的方形黑色符。这些符号是本书作者所用计算机所特有的，一般BASIC语言中没有。将这些语句删去也不会影响程序的主要功能，所以此处不逐行将其译成中文。本书中其它程序也有这些符号，情况与此相同一译者注。

②这是程序在运行过程中显示出的询问语句，用以实现人一机对话。后面括号中的Y和N是提供选择的回答键盘命令。以后同，不再注。一译者注。

```
  100 PRINT"□":CLR:OPEN1,4:DIMP(25),C(25)
  110 TT=0:PRINT:INPUT"WITH WORD SPACINGS:";A$:IFA$="N"THENTT=1
  200 PRINT:PRINT"PLAIN COMPONENT:":INPUTA$:IFLEN(A$)<>26THENPRINT"ERROR":GOTO200
  210 FORI=1TO26:K=ASC(MID$(A$,I,1))-P(I-1)=K:NEXT:PRINT
  220 PRINT:PRINT"CIPHER COMPONENT:":INPUTA$:IFLEN(A$)<>26THENPRINT"ERROR":GOTO22
0
  230 FORI=1TO26:K=ASC(MID$(A$,I,1)):C(I-1)=K:NEXT:PRINT
  250 LL=0:VV=0
  300 PRINT:PRINT:PRINT"TEXT:":A$="":M$=""
  310 PRINT"■ ■";:FORI=1TO100:NEXT:PRINT" ■";:FORI=1TO100:NEXT:GETA$:IFA$=""GOTO
310
  320 IFA$=CHR$(13)GOTO400
  330 PRINTA$;:M$=M$+A$:GOTO310
  400 L=LEN(M$):IFVV<>0GOTO407
  405 PRINT:PRINT:INPUT"ENCIPHER OR DECIPHER";ED$
  407 IFED$="D"GOTO500
  410 PRINT#1,"CRYPTOGRAM: (";L;" LETTERS )":VV=1
  430 FORI=1TOL
  435 C=0
  440 K=ASC(MID$(M$,I,1)):IFK<65THENPRINT#1," ";:GOTO470
  450 IFK<>P(C)THENC=C+1:GOTO450
  460 PRINT#1,CHR$(C(C));
  465 IFTT=1THENLL=LL+1:IFLL=5THENPRINT#1," ";:LL=0
  470 NEXTI
  480 GOTO600
  500 PRINT#1,"PLAIN TEXT: (";L;" LETTERS )":VV=1
  520 FORI=1TOL
  525 C=0
  530 K=ASC(MID$(M$,I,1)):IFK<65THENPRINT#1," ";:GOTO560
  540 IFK<>C(C)THENC=C+1:GOTO540
  550 PRINT#1,CHR$(P(C));
  560 NEXTI
  600 FORI=1TO10:PRINT#1:NEXT
  603 PRINT:INPUT"CONTINUE:";A$:IFA$="N"THENCLOSE1:END
  605 PRINT:INPUT"CONTINUE TEXT:";A$:IFA$="Y"GOTO300
  610 PRINT:INPUT"NEW TEXT:";A$:IFA$="Y"THENVV=0:GOTO300
  620 PRINT:INPUT"CHANGE CIPHER COMPONENT:";A$:IFA$="Y"GOTO220
  630 PRINT:INPUT"CHANGE PLAIN COMPONENT:";A$:IFA$="Y"GOTO200
  700 GOTO603
READY.
```

（注：以上五种应答输入在所有的模拟程序中都将采用。）

为检验程序，输入：

明表：RHEQCWMAVNSGLBUYFKDZOXTIPJ

密表：HNESCQLBXZRMGJUYDAKWOIFVPT

报文：THE QUICK BROWN FOX JUMPS OVER THE LAZY WHITE DOG。

密文：TNE SUVCA JHOQZ DOI TULPR OXFH FNE GBWY QNVFE KOM

(40)

密码分析

用简单代替方法加密的密文是较容易破译的〔2－16，18〕。程 序RUNDOWN(表2)可取字母串长达75个字母的密码并用自然顺序或是逆序的密表实现密表的扫描，这个程序对某些推移和非周期体制的密码分析也是很有用的。

CMD1(第40行和147行）与打印机有关。

在采用相同的乱序序列时(情况 5），从恢复的字母表中用手工确定密钥，是一件非

RUNDOWN - Listing #2

```
 5 OPEN1,4:DIMN(75)
 10 PRINT"□":PRINT"CIPHER":INPUTC$
 20 PRINT:INPUT"STANDARD";T$
 30 IFT$="N"GOTO140
 40 CMD1:PRINT:FORJ=1TO26
 50 FORI=1TOLEN(C$)
 60 A=ASC(MID$(C$,I,1)):IFA=32GOTO100
 80 A=A+J:IFA>90THENA=A-26
 100 PRINTCHR$(A);
 110 NEXTI:PRINT
 130 NEXTJ
 140 PRINT:PRINT:PRINT:PRINT:PRINT#1:INPUT"REVERSED";T$
 143 IFT$="N"GOTO275
 147 CMD1
 150 PRINT:FORI=1TOLEN(C$):C=ASC(MID$(C$,I,1))
 160 IFC=32GOTO180
 170 C=155-C
 180 PRINTCHR$(C);:N(I)=C
 190 NEXTI:PRINT
 195 FORJ=1TO26
 200 FORI=1TOLEN(C$):C=N(I)
 210 IFC=32GOTO240
 220 C=C+J
 230 IFC>90THENC=C-26
 240 PRINTCHR$(C);:NEXTI:PRINT
 270 NEXTJ
 273 PRINT:PRINT:PRINT:PRINT:PRINT#1
 275 INPUT"AGAIN";T$
 277 IFT$="Y"GOTO20
 278 INPUT"NEW CIPHER";T$
 279 IFT$="Y"GOTO10
 280 PRINT#1:CLOSE1:END
READY.
```

常冗长的工作，它涉及构造字母表的字母链，而后以各种不同取样间隔写出字母链直到密钥明确地显露出来为止。参阅[15]卷1第3～9页。程序K3（表3）可自动进行这项工作。输入最长的字母序列，而后对可识别的序列结果进行分析。这个程序与RUNDOWN(表2)一样，运行中自行注释(Self—explanatary)。

当着手处理一种未知的密码时，首先要做的是确定它的类型，ANALYSLS(表4)将有助于你确定密码的类型，并还可为分析简单代替类型的密码提供一些有参考价值的数据。

注意：第100行和第200行的功能与表1的是相同的。

CIPHER：输入最长为254个字母的密文。注意必须将“空格”作为第一个和最后一个符号输入。由于字母在输入后不能再更改，所以要注意避免错误；文中不必要的空格应略去。

由这个程序，你可以得到密文中每个字母的频率，总字母数，重合指数，所用不同字母数，元音字母、高频和低频辅音字母的频率。这些数据有助于判别密文是属于移位还是属于单表或多表体制。

```
100 PRINT"□":CLR:OPEN1,4:DIMA(25)
110 PRINT"SEQUENCE":INPUTA$:L=LEN(A$)
200 FORI=3TO11STEP2
210 GOSUB1000
220 NEXTI
230 FORI=15TO23STEP2
240 GOSUB1000
250 NEXTI
260 IFL>13GOTO300
270 FORI=2TO21STEP2
280 GOSUB1000
290 NEXTI
300 FORI=1TO8:PRINT#1:NEXT
310 PRINT:PRINT:INPUT"ANOTHER";A$:IFA$="Y"GOTO110
320 CLOSE1:END
1000 PRINT#1,I;" ";:N=0
1010 FORJ=1TOL
1020 K=ASC(MID$(A$,J,1)):A(N)=K:N=N+I
1030 IFN>25THENN=N-26
1040 NEXTJ
1050 FORJ=0TO25
1060 IFA(J)=0THENPRINT#1,"-";:GOTO1080
1070 PRINT#1,CHR$(A(J));
1080 NEXTJ:PRINT#1
1090 FORJ=0TO25:A(J)=0:NEXT
1100 RETURN
READY.
```

GRAPH? (Y或N) 提供直方图。

FREQUENCY SORT? (Y或N) 按频率递减的次序打印出字母。

TRIGRAPHIC CONTACTS? (Y或N) 横向打印每一个字母的前后三字母连缀。上面一行表示该字母的前一个字母，下面一行表示后一个字母。没有字母的地方打印“-”。重复的字母用“*”表示。然后列出连缀的“种类”表。即与每个字母相连的字母种类数[18]。也用递减的次序列出。然后给出总和数以及各个连缀变化数除以总字母数之商。这个数值可用来区分加密体制的某些不同类型。

VF SORT? (Y或N)以递减次序给出各个字母的频率和连缀种类数之积，并给出它们的总和，以及这个和数除以总字母数之商。它的用途仍是用于密码的分类。VF乘积是作者用以从负数密码中发现或识别明文字母E的有效方法。当然，从“不习惯”和“不常用”的字构成的密文中，识别E会更困难些。

为说明这一方法的有效性，下面列出了几份密文中VF值较大的前五个字母样本值：

1	T	E	H	O	I
	117	72	48	42	35
2	E	S	T	O	I
	108	88	81	72	63

3	E	T	N	H	A
	132	130	64	56	45

从左到右，在值差最大的两相邻字母间画了一条竖线，明文E通常(90%以上可能)是线左边字母中的一个或是线右侧第一个字母，希望读者对这一方法做进一步的实验。因为很可能从VF SORT中能获取其它更重要的信息。

LETTER POSITIONS? (Y或N) 这种选择仅用于字分割的密文的分析。对每个字

ANALYSIS - Listing #4

```
100 PRINT"D":CLR:OPEN1,4:PRINT"CIPHER"
110 DIMF(25),FS(6,25),B(25),L(80)
200 INPUT"R 或I";:FORI=1TO100:NEXT:PRINT" H";:FORI=1TO100:NEXT:GETA$:IFA$=""GOTO
200
210 IFA$=CHR$(13)GOTO300
220 PRINTA$;:C$=C$+A$:GOTO200
300 L=LEN(C$):PRINT#1,"CIPHER":PRINT#1,C$
310 FORI=1TOL:J=ASC(MID$(C$,I,1))-65:IFJ>=0THENF(J)=F(J)+1
320 NEXT
330 PRINT#1:PRINT#1,"LETTER FREQUENCIES":PRINT#1:T=0:A=0:D=0
340 FORI=0TO24STEP5:FORJ=0TO5
350 K=I+J:IFK>25GOTO380
360 PRINT#1,CHR$(K+65);F(K),
370 T=T+F(K):A=((F(K)-1)*F(K))+A:IFF(K)<>0THEND=D+1
380 NEXT:PRINT#1:NEXT
390 PRINT#1:PRINT#1,"TOTAL LETTERS";T
395 PRINT#1,"IC";A/(T*T)
400 PRINT#1,"DIFFERENT LETTERS";D:PRINT#1
410 A=F(0)+F(4)+F(8)+F(14)+F(20)+F(25)
420 PRINT#1,"AEIOUY";A;(INT((A/T)*1000+.5))/10;"%";"  (EXPECTED 40%)"
430 A=F(7)+F(13)+F(17)+F(18)+F(19)
440 PRINT#1,"HNRST";A;(INT((A/T)*1000+.5))/10;"%";"  (EXPECTED 34%)"
450 A=F(9)+F(10)+F(16)+F(23)+F(25)
460 PRINT#1,"JKQXZ";A;(INT((A/T)*1000+.5))/10;"%";"  (EXPECTED 2%)"
500 PRINT:PRINT:INPUT"GRAPH";A$:IFA$="N"GOTO600
510 PRINT#1:PRINT#1:PRINT#1,"GRAPH":PRINT#1
520 FORI=0TO25:PRINT#1,CHR$(I+65);F(I);TAB(1);
530 IFF(I)<>0THENFORK=1TOF(I):PRINT#1,"■";:NEXT
540 PRINT#1," ":NEXT
600 PRINT:INPUT"FREQUENCY SORT";A$:IFA$="N"GOTO800
610 FORI=0TO25:FS(0,I)=F(I):FS(1,I)=I:NEXT
620 P=0:FORI=0TO24:IFFS(0,I)>=FS(0,I+1)GOTO650
630 A=FS(0,I):K=FS(1,I):FS(0,I)=FS(0,I+1):FS(1,I)=FS(1,I+1)
640 FS(0,I+1)=A:FS(1,I+1)=K:P=1
650 NEXT:IFP=1GOTO620
660 PRINT#1:PRINT#1,"FREQUENCY SORT"
670 FORI=0TO24STEP5:FORJ=0TO5
680 K=I+J:IFK>25GOTO700
690 PRINT#1,CHR$(FS(1,K)+65);FS(0,K);TAB(7);
700 NEXT:PRINT#1:NEXT
710 PRINT#1:PRINT#1,"TOTAL LETTERS";T
800 PRINT:PRINT:INPUT"TRIGRAPHIC CONTACTS";A$:IFA$="N"GOTO1300
805 PRINT#1:PRINT#1:PRINT#1,"TRIGRAPHIC CONTACTS":PRINT#1
810 FORI=0TO25
820 G=0:IFF(I)=0GOTO970
830 PRINT#1,CHR$(I+65);" ";
835 P=0
840 FORJ=1TOL:IFCHR$(I+65)<>MID$(C$,J,1)GOTO890
850 K=ASC(MID$(C$,J-1,1))-65:IFK<0THENPRINT#1,"-";:GOTO880
860 IFK=ITHENPRINT#1,"*";:GOTO880
870 PRINT#1,CHR$(K+65);:B(K)=B(K)+1
880 L(P)=ASC(MID$(C$,J+1,1))-65:P=P+1
890 NEXTJ:PRINT#1:PRINT#1,TAB(2);
900 FORJ=0TOP-1
910 IFL(J)<0THENPRINT#1,"-";:GOTO940
```

```
 920 IFL(J)=ITHENPRINT#1,"*";:GOTO940
 930 PRINT#1,CHR$(L(J)+65);:B(L(J))=B(L(J))+1
 940 NEXT:PRINT#1:PRINT#1
 950 FORK=0TO25:IFB(K)<>0THENG=G+1
 960 NEXTK
 970 FS(0,I)=G:FS(1,I)=I:FORK=0TO25:B(K)=0:NEXTK:NEXTI
 980 P=0:FORI=0TO24:IFFS(0,I)>=FS(0,I+1)GOTO1010
 990 A=FS(0,I):K=FS(1,I):FS(0,I)=FS(0,I+1):FS(1,I)=FS(1,I+1)
1000 FS(0,I+1)=A:FS(1,I+1)=K:P=1
1010 NEXT:IFP=1GOTO980
1015 P=0:PRINT#1:PRINT#1,"VARIETY"
1020 FORI=0TO24STEP6:FORJ=0TO5
1030 K=I+J:IFK>25GOTO1045
1040 PRINT#1,CHR$(FS(1,K)+65);FS(0,K);TAB(7);:P=P+FS(0,K)
1045 NEXT:PRINT#1:NEXT
1050 PRINT#1:PRINT#1,"TOTAL";P;"(";(INT(((P/T)*1000)+.5))/1000;")"
1100 PRINT:PRINT:INPUT"VF SORT";A$:IFA$="N"GOTO1300
1110 FORI=0TO25:FS(0,I)=FS(0,I)*F(FS(1,I)):NEXT
1120 P=0:FORI=0TO24:IFFS(0,I)>=FS(0,I+1)GOTO1150
1130 A=FS(0,I):K=FS(1,I):FS(0,I)=FS(0,I+1):FS(1,I)=FS(1,I+1)
1140 FS(0,I+1)=A:FS(1,I+1)=K:P=1
1150 NEXT:IFP=1GOTO1120
1160 P=0:PRINT#1:PRINT#1:PRINT#1,"VF SORT"
1170 FORI=0TO24STEP6:FORJ=0TO5
1180 K=I+J:IFK>25GOTO1200
1190 PRINT#1,CHR$(FS(1,K)+65);FS(0,K);TAB(7);:P=P+FS(0,K)
1200 NEXT:PRINT#1:NEXT
1210 PRINT#1:PRINT#1,"TOTAL";P;"(";(INT((P/T)*1000+.5))/1000;")"
1300 IFT+2=LGOTO1500
1310 PRINT:PRINT:INPUT"LETTER POSITIONS";A$:IFA$="N"GOTO1500
1320 FORI=0TO25:FS(0,I)=0:FS(1,I)=0:NEXT
1330 I=1:J=2
1340 IFASC(MID$(C$,I,1))<>32THENI=I+1:GOTO1340
1350 A$=MID$(C$,J,I-J):G=INT(LEN(A$)/2):FORK=1TOLEN(A$):P=ASC(MID$(A$,K,1))-65
1360 IFLEN(A$)=1GOTO1390
1370 IFK<=GANDK<=3THENFS(K-1,P)=FS(K-1,P)+1:GOTO1400
1380 A=K-LEN(A$)+7:IFK>GANDK<LEN(A$)-GTHENFS(A-1,P)=FS(A-1,P)+1:GOTO1400
1390 FS(3,P)=FS(3,P)+1
1400 NEXT
1410 I=I+1:J=I:IFI=LEN(C$)+1GOTO1430
1420 GOTO1340
1430 PRINT#1:PRINT#1:PRINT#1,"LETTER POSITIONS":PRINT#1
1440 PRINT#1,TAB(7);"1";TAB(7);"2";TAB(7);"3";TAB(7);"0";TAB(7);"3";
1450 PRINT#1,TAB(7);"2";TAB(7);"1":PRINT#1
1460 FORI=0TO25:IFF(I)=0GOTO1495
1470 PRINT#1,CHR$(I+65);TAB(5);FS(0,I);TAB(5);FS(1,I);TAB(5);FS(2,I);
1480 PRINT#1,TAB(5);FS(3,I);TAB(5);FS(4,I);TAB(5);FS(5,I);TAB(5);FS(6,I)
1495 NEXT
1500 PRINT:PRINT:INPUT"REPETITIONS";A$:IFA$="N"GOTO1600
1505 PRINT#1:PRINT#1,"REPETITIONS":PRINT#1
1510 FORI=2TOL-6
1520 FORJ=I+3TOL-3:A=2:IFMID$(C$,I,1)=MID$(C$,J,1)GOTO1540
1530 NEXT:NEXT:GOTO1600
1540 IFMID$(C$,I+1,1)<>MID$(C$,J+1,1)GOTO1530
1550 IFMID$(C$,I+A,1)<>MID$(C$,J+A,1)GOTO1570
1560 A=A+1:GOTO1550
1570 IFA=2GOTO1530
1580 PRINT#1,MID$(C$,I,A);TAB(5);:GOTO1530
1600 FORI=1TO9:PRINT#1:NEXT:CLOSE1:END
READY.
```

的前面第一，二，三个字母和倒数第一、二、三个字母频率进行计数。有些明文字母更多地出现在一个字的某些特定位置。参阅〔18〕第 8 页。

REPETITIONS？（Y或N）列出所有的三字母或更多字母构成的重复字母组。注意，如存在一个六个字母的重复，则其中的五字母、四字母、三字母也还应作为重复列出。除了上面给出的信息外，建议参考〈18〉中的应用。

致　读　者

就现代密码分析技术而言，本章(以及下面各章)中给出的计算机程序是很重要的。然而也象其它事情一样，它们还可以进一步改进。此外，用以模拟更多密码体制的计算机程序也会被陆续设计出来。简单地说，本书中给出的程序只能作为读者自己实践的一个“出发点”。同时，一方面为提高读者兴趣，另一方面也作为一些初步设想为读者提供进一步的工作方向，，下面列出一些程序(或要求)供读者实习。

1、一个能产生数字密码等价的程序。

2、一个能产生用任意符号作为密码等价的程序。

3、一个分析数字密码的程序。

4、能提供一个以上数字密码等价的程序，参阅〔13〕第45—47页。

5、一个密码序列是一个有重复字母的短语的程序，参阅〔13〕第39页。

6、一个能产生基于坐标系的，一对字母作为密码等价的程序，参阅〔13〕第43页。

7、如Bruce Schatz〔20〕所发表的那种基于微系统的圆盘。

8、一个进行单值分解的程序〔20〕。

9、多名码体制的分析〔21〕。

10、复制MIT “DECRYPTOR” 程序〔22〕。

问　题

1. XG XR RQXT GWQG QBNPXSQC XCTXQCR JNPN DCN DU GWN UXPRG XCWQOXGQCGR DU GWXR SDHCGPL ONSQHRN GWNL WQT PNRNPIQGXDCR. (94)

2. KSOIO EJWNO ITZDB COKEK ENOIO WAOJK WKOYO NOADG BOCKZ DBGWC TXMEA YECRS DBOJE CKSOJ MXYEN EJEDC YDPCK SOIDW YKSOE IJADR WCEJW ADKUD IWAEK KAO. (118)

3. CWCTX PVWXY VPIHC ACWJB RNVAD IACSV ITECT BCPLN VWCPP LDIGX ICTAJ UBNJT HJHBR ACYYI TSVXX NDJMJ HHITS CINDJ YCGNW ITBNJ UJACB GJDJS VNAVY HXJJN. (120)

4. FDZYC NDCGN JEYKN PGNRE JELQX ZJENZ ZODED YQNGJ NWNJE RMANN CANZZ. (55)

5. XHIVN DQIZS XTPVS AMSDW XPISA WIKSZ BYNEX MSAUX ZIVEX SAMSA PBLYS VPXAP PSQQS AUSES AEBXU TXDDB LEICF STXDI HIVXT ESWID SYITS IHIEG IPVSA MSDZX TTIPE ICFST XWBZM SAUYS VP. (142)

第二章　周期性多表代替(PERIODIC POLYALPHABETIC SUBSTITUTION)

基本情况

使用单一的加密密表只能提供很低的保密度。即使在单表的基础上加以某些改进也是如此。因而，为获得更高的保密度，按预先约定的次序轮流使用几个密表进行加密，是单表加密的合乎逻辑的自然发展；最常用的是使用一个较短的重复的密钥字或短语〔2－14，16，19，23，25〕。明文分量和密文分量分别由各自的园盘或滑条所描述，这些园盘和滑动条的相对位置则由密钥字母决定。字母或各个分量的顺序可任取第5页提到的七种形式之一。

当加密体制中是采用自然顺序的序列时，则这种加密体制常称为维吉尼亚(Vigenere)，变异的维吉尼亚(Variant)博福特(Beaufort)，波他(Porta)和格龙斯弗尔德(Gronsfeld)体制，最后一种体制是维吉尼亚体制的一种特例。

几乎任何一种为进行简单代替而设计的密码设备也都适应多表代替，下列专利则更适合进行多表代替加密：

发明者	专利号
D.R.Smith	312665
W.C.Van Horn	637049
G.B.N.Valvasori	641481
L.H.Westorn	723566
S.T.Maryc	1201486
J.W.Wulf	1271000
A.Newell	1441109
F.R.Seaver	1679380
F.Schimmel &E.A.Bedault	1921327
A.L.Patton	2055702
H.O.Rugh	2110149
M.C.MCM.O′Brien	2270137
R.M.Mitchell	2439413

应当注意的是，上述设备似乎一个也没有真正在实践中被采用。

将两个基本分量相对平行移动，就产生了一系列相互联系的派生密表。这个结论可由密码方程式表述。参阅〔11〕中第161～171页和〔23〕第140页。在下面的方程式中，P、K和C分别代表明文、密钥和密码中字母的位置数。等式揭示了它们相互之间有趣的联系，更重要的是，这些等式可以在计算机上由程序迅速实现。

	加密	解密
维吉尼亚	P+K=C	C-K=P
变异维吉尼亚	P-K=C	C+K=P
博福特	K-P=C	K-C=P

在此基础上还可加以各种各样的变异，常见的有：在用下一个密钥字母之前加密的字母组长度可以相同，也可以不同；用不同的密钥字母加密每个字。除此之外，其它种种变化也是可能的。

程序

程序POLYALPHABET(表5)采用自然顺序的字母表，分别根据上述加密方程式实现维吉尼亚、变异维吉尼亚、博福特等体制的加密。

第100行和210行前面已经解释过了。第270行和280行中的那些奇特的符号是为显示器用作显示标志的符号。

POLYALPHABET - Listing #

```
 100 PRINT"J":OPEN1,4
 200 PRINT:PRINT"TEXT:":C$=""
 210 PRINT"■ ■■■";:FORI=1TO180:NEXT:PRINT" ■";:FORI=1TO100:NEXT:GETA$:IFA$=""GOTO
210
 220 IFA$=CHR$(13)GOTO240
 230 PRINTA$;:C$=C$+A$:GOTO210
 240 PRINT:L=LEN(C$):IFT<>0GOTO290
 250 PRINT:PRINT:INPUT"KEYWORD:";KW$:LL=LEN(KW$):T=1
 260 PRINT:PRINT"SYSTEM TYPE:"
 270 PRINT"■1■. VIGENERE":PRINT"■2■. BEAUFORT":PRINT"■3■. VAR!ENT
 275 PRINT:INPUT"WHICH:";S
 280 PRINT:INPUT"■E■NCIPHER OR ■■ECIPHER:";ED$
 290 IFED$="D"GOTO500
 300 PRINT#1,"CRYPTOGRAM: (";L;" LETTERS )"
 305 ONSGOTO310,370,430
 310 FORI=1TOL
 320 P=ASC(MID$(C$,I,1))-65:K=ASC(MID$(KW$,T,1))-65
 330 C=P+K:IFC>25THENC=C-26
 340 PRINT#1,CHR$(C+65);:T=T+1:IFT>LLTHENT=1
 350 NEXT
 360 GOTO600
 370 FORI=1TOL
 380 P=ASC(MID$(C$,I,1))-65:K=ASC(MID$(KW$,T,1))-65
 390 C=K-P:IFC<0THENC=C+26
 400 PRINT#1,CHR$(C+65);:T=T+1:IFT>LLTHENT=1
 410 NEXT
 420 GOTO600
 430 FORI=1TOL
 440 P=ASC(MID$(C$,I,1))-65:K=ASC(MID$(KW$,T,1))-65
 450 C=P-K:IFC<0THENC=C+26
 460 PRINT#1,CHR$(C+65);:T=T+1:IFT>LLTHENT=1
 470 NEXT
 480 GOTO600
 500 PRINT#1,"PLAIN TEXT: (";L;" LETTERS )"
 510 ONSGOTO430,370,310
 600 FORI=1TO9:PRINT#1:NEXT
 610 PRINT:INPUT"CONTINUE:";A$:IFA$="N"THENCLOSE1:END
 620 PRINT:INPUT"CONTINUE TEXT:";A$:IFA$="Y"GOTO200
 625 PRINT:INPUT"NEW TEXT:";A$:IFA$="Y"THENT=0:GOTO200
 640 PRINT:INPUT"CHANGE SYSTEM TYPE:";A$:IFA$="Y"GOTO260
 650 PRINT:INPUT"NEW KEYWORD:";A$:IFA$="Y"GOTO250
 660 GOTO610
READY.
```

TEXT：输入报文，最长为256个字符，省略所有的空格。

KEYWORD：输入不超过75个字母的单词或短语。