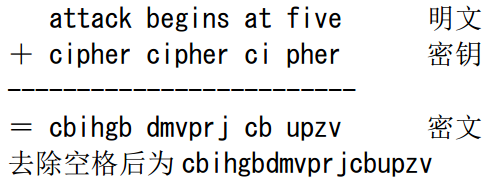
实验 **1** 古典密码体制

一．实验目的  
通过编程实现古典密码体制中的维吉尼亚密码体制， 并了解古典密码的密码分析方式，加深对古典密码体制的理解。

二．实验环境：  
硬件设备： PC 机  
操作系统： windows 2000 操作系统  
编程工具： Matlab VC++

三．维吉尼亚（**Vigenere**）密码  
维吉尼亚（Vigenere）密码是一种多表代换密码算法。它选择一个词组作为密钥，密钥中每个字母用来确定一个代换表，每个密钥字母用来加密一个明文字母， 例如密钥字母为 a，明文字母为c，则密文字母为0+2(mod26)=2，也就是c。直到所有的密钥字母用完，后再从头开始，使用第一个密钥字母加密。也就是说，密钥循环使用。  
例如： 明文为 attack begins at five，密钥为 cipher，  
 

四．实验内容  
1. 使用 matlab 或者其它的编程语言，实现维吉尼亚（Vigenere）密码算法。 要求写入报告中。  
2. 利用你的程序， 针对下列明文， 应用加密算法，选择相应的密钥， 运行程序，记录密文。  
 a) 明文： are you going to scarborough fair  
 密钥： paul simon  
 密文： pryzygczubxnaztgzeqngbicomotzspil  
 b) 明文： without no seams nor needlework  
 密钥： thyme  
 密文： pprtsnazzszzcmqlzlavzucqheluavd  
3. 下列密文使用了维吉尼亚（Vigenere）密码， 密钥长度为 5，由字母组成。写个小程序，用穷举搜索法（亦称暴力法） 破解密码。 记录你的程序花费的时间。 要求将程序写入报告中。  
明文： scientists try to answer questions about the world around us密文： zgtpbamdeg avj ec hrdhsy ufpgamzyg hfzfh alp hcypo lfvyyo iz  
密钥：

五．思考（写入报告）  
一、 NSA 想雇佣你做密码分析工作。 这份工作主要分析使用维吉尼亚（Vigenere）加密的密文，但是仅有密文。要求你能分析出明文。 为了表明你能胜任这份工作， 请回答下面的问题：  
1） 如何分析维吉尼亚（Vigenere） 密码体制加密的密文？  
2） 暴力破解法是否有效，在什么情况下可以使用暴力破解法？  
3） 如果允许你借助计算机辅助分析，但是你只能自己动手设计程序，这个程序该如何设计？  
4） 给出你的工作方案  
二、（选作） 作为测试， NSA 要求你破译下面的这段密文，给出明文和密钥，要求写出分析过程：  
ALPGLECTALZHLASDHVPPZOTCNXSTKIGDAIPHVJNXCMWGPKSIZASTUATASCZJIIDPAM  
DUPIOLLGLCUIGTYFPHHXTHMMPSHWWDUKLHALPCLKCDPWEWLZTRAMXDMXSTBRDELEVPIPP  
WVVCDYWZUWSWXJIMGBXLAPXJLLGLCUIGTYFPHHXTHMMPSHWWDUKLHVYCQVHTTZLPPCCHX  
ALEWLJLIPKFTVJEGHZPAJEYCVXRPPRWDKKTCNMYIOIXDAIWHVJEWLLTVOALNZEYSALPWV  
XPAZSQIOINXAMPHDINPURZIIIDPAMDUPIOPZPZCNEDPUIRGVMYBPWDXZWTEWMNPURZICS  
ETHROPUIRGVMYCLAJDYOMTSMPKLWSTOEDCVXSXUKQDYASXJLEDCSETUSYDDILGLRZIZEE  
XZJTTKEYSDIHXSPYDAFPHHXTHMMPSBREXSNFHAMNTYSWAZHZLUPTZLALILVDPUHCXNLET  
VYDCLWDAPOPPTMRWACDIYILBPEXCVXFCTMYSMYWIOEEHVQPDMCZJOEGTJSXTOICTVYEDM  
KCTHXEGPEWHHROIYMMJSEEXVRDHVQPDMCZJOEGTJSXTMVPHOJCDTRLGYSHYHMWRLPWHHR  
OHVQPDMCZJOEGTJSXTMVZBHVPPZASTYIJDBVBJLWEFBIDIMSCUYIPSVQWTMXJDBFLIAIC  
TKFJIOIDIVVXHVJATYWPRBXTDUEYSZXLVNICTKFJIOIHXUHDDMTZAPGPQYYEPSMENFSFW  
HZPQLIYIOIGTAICPUWZUJVPPAMGTZYQULVTCNGZCAMYJLXZLVVVLPXSIOIQPPXSIOEEJU  
ILGUIOHBJQTYMYVPWCTKIXEAMGTNSMPJOEDTMDHPWDXWTTVVFLRRXZPSEMPTERDIENZAS  
DDBXSRHVZAPRLVVFLRRXZVLSCVPERDIENZASWDBMDXHRLVVFLRRXZIOIDABQDPUHRWLXE  
DZSQDBVYDYXSTYRNXAMPHRRZLPRRIOEEHVQPWVAEWPWDXAYLIPSYRHRLCKATASFPROEYV  
LH  
提示： 明文是英文，加密体制为维吉尼亚（Vigenere）密码体制， 密钥第 2 个字母为 E。

六．附录  
1． 英文字母频率表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J |
| 0.082 | 0.015 | 0.028 | 0.043 | 0.127 | 0.022 | 0.020 | 0.061 | 0.070 | 0.002 |
| K | L | M | N | O | P | Q | R | S | T |
| 0.008 | 0.040 | 0.024 | 0.067 | 0.075 | 0.019 | 0.001 | 0.060 | 0.063 | 0.091 |
| U | V | W | X | Y | Z |  |  |  |  |
| 0.028 | 0.010 | 0.023 | 0.001 | 0.020 | 0.001 |  |  |  |  |

2． 字母可以分为五组：  
E 0.127 TAOINSHR 0.06~0.09  
DL 0.04 CUMWFGYPB 0.015~0.023  
VKJXQZ 小于 0.01  
3． 最常见的两字母组合，依照出现次数递减的顺序排列： TH、 HE、 IN、 ER、 AN、 RE、DE、 ON、 ES、 ST、 EN、 AT、 TO、 NT、 HA、 ND、 OU、 EA、 NG、 AS、 OR、 TI、 IS、 ET、IT、 AR、 TE、 SE、 HI、 OF。  
4．最常见的三字母组合， 依照出现次数递减的顺序排列： THE、 ING、 AND、 HER、 ERE、ENT、 THA、 NTH、 WAS、 ETH、 FOR、 DTH。

七．实验代码

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void run();

char VigenereCrypt(char \*alphabet, int \*key, char message, int pos);

char VigenereDecrypt(char \*alphabet, int \*key, char message, int pos);

int main()

{

while(1)

{

run();

printf("\n======================================\n");

}

return 0;

}

void run()

{

char message[50], messageEncrypted[50], messageDecrypted[50], key[15], alphabet[27] = "abcdefghijklmnopqrstuvwxyz";

int x, i, j, messageLenght, keyLength, pos = 0, \_key[50];

printf("请输入信息:");

gets(message);

printf("请输入密钥: ");

gets(key);

messageLenght = strlen(message);

keyLength = strlen(key);

for (j = 0; j < keyLength; j++)

{

for (i = 0; i <= 25; i++)

{

if (key[j] == alphabet[i])

{

\_key[j] = i;

}

}

}

printf("\n1 = 加密信息.\n");

printf("2 = 解密信息.\n");

printf("3 = 退出系统.\n");

scanf("%d", &x);

switch (x)

{

case 1:

for (i = 0; i < messageLenght; i++)

{

if (pos >= keyLength)

{

pos = 0;

}

messageEncrypted[i] = VigenereCrypt(alphabet, \_key, message[i], pos);

pos++;

}

messageEncrypted[messageLenght] = '\0';

printf("\n加密密文为: %s\n", messageEncrypted);

break;

case 2:

for (i = 0; i <= messageLenght; i++)

{

if (pos >= keyLength)

{

pos = 0;

}

messageDecrypted[i] = VigenereDecrypt(alphabet, \_key, message[i], pos);

pos++;

}

messageDecrypted[messageLenght] = '\0';

printf("\n解密明文为: %s\n", messageDecrypted);

break;

case 3:

exit(0);

default:

printf("\n 输入错误!!! \n");

}

getchar();

}

char VigenereCrypt(char \*alphabet, int \*\_key, char message, int pos)

{

int i, messageEncrypted;

for (i = 0; i <= 25; i++)

{

if (message == alphabet[i])

{

messageEncrypted = (i + \_key[pos]) % 26;

return alphabet[messageEncrypted];

}

}

}

char VigenereDecrypt(char \*alphabet, int \*\_key, char message, int pos)

{

int i, messageDecrypted;

for (i = 0; i <= 25; i++)

{

if (message == alphabet[i])

{

messageDecrypted = (i - \_key[pos]) % 26;

return alphabet[messageDecrypted];

}

}

}

//密钥破解

#include<stdio.h>

#include<math.h>

//寻找密钥长度

int find\_key\_lenth(char\*pass,int len)

{

//pass密文,len 密文长度；移位统计相等的密文，取其最大的步数为d;

int d=0,count,MaxCount=0;

for(int step=1;step<10;step++) //移动步数从1-10;

{

count=0;

for(int j=0;j<len&&(j+step)<len;j++)

{

if(pass[j]==pass[j+step])

count++;

}

if(count>MaxCount)

{

MaxCount=count;

d=step;

}

}

return d;

}

// 发现密钥并解密。。。

void decode(char\*pass,char\*ming,int d,int len)

{

float v[26]={0}; //V或W向量组；

int per\_len=len/d; //每组长度；

double A[26]={0.082,0.015,0.028,0.043,0.127, //英文字母频率表A

0.022,0.02,0.061,0.07,0.002,0.008,

0.04,0.024,0.067,0.075,0.019,0.001,

0.06,0.063,0.091,0.028,0.01,0.023,0.001,0.02,0.001};

double B[26]={0}; //存储W\*A值

char\*key; //密钥

key=new char[d];

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

{

int j=0;

while(true) //统计每组a--z出现的频率存在V[26]中

{

if((i+d\*j)>=len) break;

v[pass[i+d\*j]-'a']+=1;

j++;

}

for(int k=0;k<26;k++) //计算W

v[k]=v[k]/per\_len;

for(int k=0;k<26;k++) //计算B[i]=Ai\*V;

{

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

B[k]+=A[l]\*v[(l+k)%26];

}

//找出B中的与0.065最接近的值其的下标即为密钥

double max=1;

int c;

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

{

if(fabs(B[k]-0.065)<max)

{

max=fabs(B[k]-0.065);c=k;

}

}

key[i]=c;

//清空B，V；

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

{

B[k]=0;

v[k]=0;

}

printf("%c",'a'+key[i]);

}

//解密并显示

printf("\n\n明文:\n");

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

{

int tmp;

tmp=pass[i]-'a';

ming[i]=(tmp-key[i%d]+26)%26+'a';

printf("%c",ming[i]);

}

printf("\n\n");

return ;

}

int main()

{

char password[1000]={0}; //密文

char mingwen[1000]={0}; //明文

FILE\* fp;

if((fp=fopen("message.txt","r+"))==NULL)

{

printf("文件打开错误!\n");

return 1;

}

int i=0,d;

printf("密文:");

while((password[i++]=fgetc(fp))!=EOF);

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

{

if(j%59==0) printf("\n");

printf("%c",password[j]);

}

d=find\_key\_lenth(password,i-1);

// d=5;

printf("\n\nd==%d\n",d);

printf("密钥:");

decode(password,mingwen,d,i-1);

fclose(fp);

return 0;

}