**古典密码实验--仿射密码**

17116060120 马耀文

1. **算法原理**

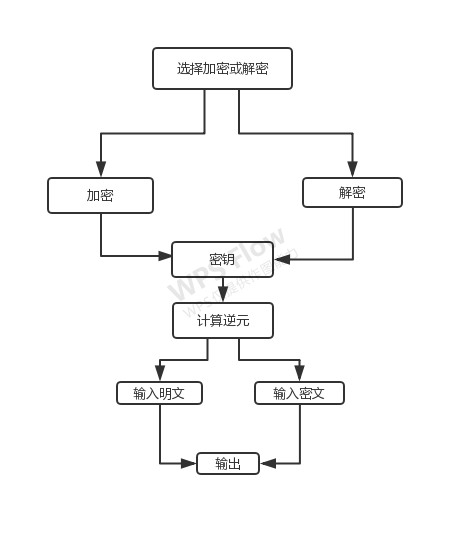
仿射变换（及乘积密码）的加密解密分别是：

c = E（m） ≡ a + b(mod 26)

m = D(c) ≡ a-1 (c - b)(mod 26)

其中，a,b是密钥，为满足0≤a,b≤25和gcd(a,26)等于1的整数。其中gcd(a,26)表示a和26的最大公因数，gcd(a,26)=1表示a和26是互素的，a-1表示a的逆元，即a-1\*a ≡ 1 mod 26。

1. **算法参数**
2. 参数m[1024]和c[1024]数组存放明文或密文的字符串，
3. 秘钥空间K=（key\_1，key\_2）
4. n为最大公约数
5. t0,s0为乘法逆元
6. choice判断加密解密操作，当choice=1时，为加密，choice=2为解密
7. **算法流程**



4**.算法实现**

#include<stdio.h>

#include<string.h>

int gcd(int n,int m) #通过gcd算法求出最大公约数

{

if(n>m)

{

int t;

t=n;

n=m;

m=t;

}

if(m%n == 0)

return n;

else

gcd(m%n,n);

}

int re(int a,int b) #通过re函数算出key\_1,key\_2密钥的乘法逆元

{

int t,q,r0,r1,r2,s0=1,s1=0,t0=0,t1=1,t2,s2,s=0;

int h = gcd(a,b);

if(h == 1){

if(a<b)

{

s = 1;

t = a;

a = b;

b = t;

}

r0 = a;

r1 = b;

while(r1 != 0)

{

q = r0/r1;

r2 = r0-q\*r1;

s2 = s0-q\*s1;

t2 = t0-q\*t1;

r0 = r1;r1=r2;

s0 = s1;s1=s2;

t0 = t1;t1=t2;

if(r0 == 1)

{

if(s0 < 0)

{

s0 = s0+b;

}

if(t0 < 0)

{

t0 = t0+a;

}

if(s == 1)

return t0;

else

return s0;

}

}

}

else

{

return 0;

}

}

void encrypt() #加密算法

{

char m[1024],c[1024];

int key\_1,key\_2,key\_re;

printf("Please input key\_1 and key\_2：");

scanf("%d%d",&key\_1,&key\_2);

key\_re = re(key\_1,26);

if(key\_re == 0)

{

printf("key\_1 have not inverse element!\n");

return;

}

else

{

printf("Please input message：");

getchar();

gets(m);

int len = strlen(m);

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

{

if(m[i] >= 'A'&&m[i] <= 'Z')

c[i] = 'A'+(key\_1\*(m[i]-'A')+key\_2)%26;

else if(m[i] >= 'a'&&m[i] <= 'z')

c[i] = 'a'+(key\_1\*(m[i]-'a')+key\_2)%26;

else

c[i] = m[i];

}

printf("Cipher：");

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

{

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

}

}

}

void decrypt() #解密算法

{

char m[1024],c[1024];

int key\_1,key\_2,key\_re;

printf("Please input key\_1 and key\_2：");

scanf("%d%d",&key\_1,&key\_2);

key\_re = re(key\_1,26);

if(key\_re == 0)

{

printf("key\_1 have not inverse element!\n");

return;

}

else{

printf("Please inpuy cipher：");

getchar();

gets(c);

for(int i=0;i<strlen(c);i++)

{

if(c[i] >= 'A'&&c[i] <= 'Z')

{

if((key\_re\*((c[i]-'A')-key\_2))%26<0)

m[i] = 'A'+(key\_re\*((c[i]-'A')-key\_2))%26+26;

else

m[i] = 'A'+(key\_re\*((c[i]-'A')-key\_2))%26;

}

else if(c[i] >= 'a'&&c[i] <= 'z')

{

if((key\_re\*((c[i]-'a')-key\_2))%26<0)

m[i] = 'a'+(key\_re\*((c[i]-'a')-key\_2))%26+26;

else

m[i] = 'a'+(key\_re\*((c[i]-'a')-key\_2))%26;

}

else

m[i] = c[i];

}

printf("Message：");

for(int i=0; i<strlen(c); i++)

{

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

}

}

}

int main()

{

int choice;

printf("1.encrypt\n");

printf("2.decrypt\n");

printf("Please input your choice:");

scanf("%d",&choice);

switch(choice){

case 1:encrypt();break;

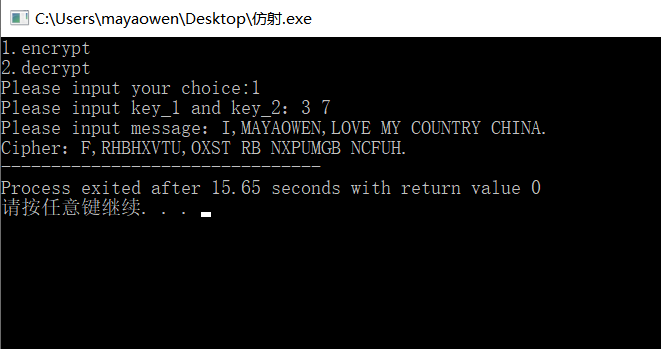
case 2:decrypt(); break;

}

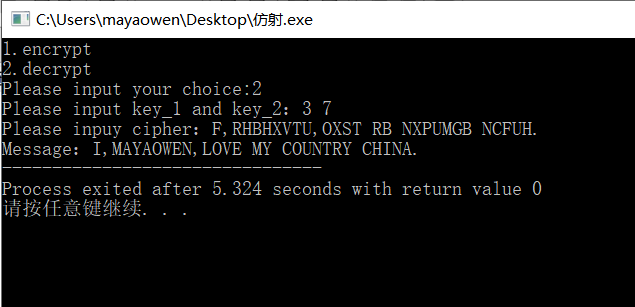
}

5.**加密解密过程演示**

加密，密钥key\_1 = 3，key\_2 = 7，明文：I,MAYAOWEN,LOVE MY COUNTRY CHINA.



解密，密钥key\_1 = 3，key\_2 = 7，密文：F,RHBHXVTU,OXST RB NXPUMGB NCFUH.



1. **总结**

通过本次古典密码-仿射密码的编程实验，既巩固了仿射密码的原理，同时也提升了用变成解决问题的操作能力，过程中遇到过一些算法上的错误，但是通过询问老师和上网百度很快解决了。同时，对于仿射密码，之前一直都是利用工具或者脚本直接跑出答案，但是稍加改动就很难解出，原因就是没用深刻理解算法原理，本次实验弥补了对算法原理理解上的不足。