**网络与信息安全实验报告**

**学院名称：软件与微电子学院**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 沈哲宇 | **学号** | 2014303357 | **专业** | 软件工程 |
| **班级** | 14011402 | **实验项目** | 实验3、RSA加解密实验 | | |
| **实验时间** | | 2017/6/11 | | **实验地点** | 毅字楼310 |
| **批改意见** | | | | **成绩** |  |
| **批改教师：** | | | | | |
| **一、实验目的**  1、了解RSA算法。  **二、实验内容和要求**  1、在互联网上查找RSA算法的C/C++（任选一种）实现代码，并调试通过。用该代码生成一对公钥和私钥，并用**私钥**加密一段明文“Today is a nice day”。  2、提交你的私钥、公钥、密文、解密过程、实验报告及源码。  3、形成实验报告，报告和源代码以附件的形式发送到：  **homeworknpu@163.com**  5、邮件标题为：  2017网络与信息安全，姓名，学号，第n次实验  例如：2017网络与信息安全，张三，20138888，第1次实验  **注：以下内容由同学们完成：**  **三、实验过程记录**  1. RSA加密算法  RSA加密算法是一种非对称加密算法，是目前最有影响力的公钥加密算法，能够抵挡目前为址已知的所有密码攻击。RSA加密算法基于一个十分简单的数论事实：将两个大素数相乘十分容易，但想要把它们的乘积进行因式分解却非常困难，因此可以将乘积公开作为加密密钥。  (1) 公钥和密钥的产生  假设Bob通过一个不可靠媒体接收Alice的一条私人消息。通过以下步骤产生一个公钥和私钥：  A. 随意选择两个大的素数和，不等于，计算；  B. 根据欧拉函数，求得；  C. 选择一个小于的整数，求得关于模的模反函数，命名为（模反元素存在，当且仅当e与r互质）；  D. 将和的记录销毁。  是公钥，是私钥。Bob将他的公钥传给Alice，而将他的私钥加密保存起来。  (2) 加密消息  假设Alice想给Bob送一个消息 ，她知道Bob产生的和。她使用与Bob约好的格式将转换为一个小于的整数，例如她可以将每一个字转换为这个字的Unicode码，然后将这些数字连在一起组成一个数字。如果待加密的信息很长，她可以将这个信息分为几段，然后将每一段转换为。  用下面这个公式即可将加密为：  Alice计算出后就可以把它传送给Bob。  (3) 解密消息  Bob收到Alice的消息后，用它的密钥对消息解密。  用下面的公式可以把转换为：  2. 调试程序  源程序地址为https://github.com/iphkwan/SimpleRSA  在源程序中，作者实用了OAEP(Optimal Asymmetric Encryption Padding)算法，在消息中加入随机数据位，使得密文难以统计出规律，并且能防止攻击者解密出部分明文。在本次实验的调试中，将相关代码片段删除。  在源程序中，作者使用了Rabin-Miller素数检验法。在一般情况下生成素数都使用埃拉托斯特尼筛法可以很快地生成小的素数，但这种方法对于大素数生成来说比较困难。比较好的方法是随机生成一个给定位数的奇数，然后采用素性测试方法对它进行检验看它是否为素数，若不是则重新生成另一个奇数，重复这个过程。这样生成素数可以避免素数表或者生成素数时的大量开销，本次实验中保留改方法。  该程序的结构图为  Main  RSA  StringTrans  BigInt  Key  PrimeGen  如上图所示，“Main”是主函数，负责组织程序的运行。“RSA”、“OAEP”、“StringTrans”为算法库，分别封装了RSA算法、OAEP算法与字串剪接算法的核心操作接口，供“View”层调用。“BigInt”为大数类，封装了大整数的表示与运算。至于“Key”和“PrimeGen”两个类是为了系统的安全性而设计的，其保存了RSA算法生成公私钥过程中的中间变量（如：两个大素数p和q）。一旦公私钥生成成功，“Key”实体将会被销毁，以保证第三方最多只能从客户端获取公钥（n，e），而无从获取生成n的素数p与q。  经过代码修改，该程序使用512长度密钥的RSA加密，输入明文后，输出生成的两个素数，密钥以及加密，对密文的解密结果。  3. 运行结果    如上图所示：  公钥为9855CDAF46E9601A8137F265B74C39E195106211E8FB430BCF1C780B06CFE3E7  私钥为24C1A1B57FE8CEB0C0AE828932E9EA278643FF5D5ADCF2B1572484465ACF93B5  6DD4A840C4A5B78584D3D63FD6385CC7B1C9CD4A0CF47  N为8B7AFAF330B471A4333F09453F7C438CC31FB6648594B9944FE4DCF30819E591  C513B65ED3C995B918B0AB398D56D0DB6FEEE15A7A7237F5D57DE56AE1EC5777  加密结果的16进制为895D18A8AE8CA9A3AD12092EC063BAB9F8362567943BB8B7DC  25A956EADF6B4F31D032FB4555D883DEB86BBFDBBB0D364BECF7F4642A1609DAC20E02  65089363  解密结果的ASCII码为Today is a nice day  **四、实验结果分析和结论**  了解了RSA算法的流程。在互联网上查找到合适的代码，并进行相关修改和调试，给出了符合要求的输出。  本次实验代码和实验报告已上传：https://github.com/shenzheyu/RSA | | | | | |
|  | | | | | |

附录代码：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void show1() //主界面

{

printf("\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DES加密解密系统 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t--------------------------------------------------\n");

//printf("\t\t--------------------------------------------------\n");

printf("\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t1.加密\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t2.解密\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t3.退出\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t--------------------------------------------------\n");

}

void show2() //加密界面

{

printf("\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DES加密 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t--------------------------------------------------\n");

printf("\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t请选择明文和密钥的输入方式:\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t1.直接输入\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t2.从文件读取\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t3.退出\t\t\t\t\*\*\n");

printf("\t\t\*\*\t\t\t\t\t\t\*\*\n");

printf("\t\t--------------------------------------------------\n");

printf("\t\t\t选择:");

}

void reader(char str[30],char s[8]) //读取明文和密钥

{

FILE \*fp;

fp=fopen(str,"r");

if(fp==NULL)

{

printf("明文读取失败!\n");

}

else

{

fscanf(fp,"%s",s);

}

fclose(fp);

}

void To2Bin(char p[8],int b[64]) //将字节转换成二进制流

{

int i,k=0;

for(i=0;i<8;i++)

{

int j=0x80;

for(;j;j>>=1)

{

if(j&p[i])

{

b[k++]=1;

}

else

{

b[k++]=0;

}

}

}

}

int IP\_Table[64] = //初始置换(IP)

{

57, 49, 41, 33, 25, 17, 9, 1,

59, 51, 43, 35, 27, 19, 11, 3,

61, 53, 45, 37, 29, 21, 13, 5,

63, 55, 47, 39, 31, 23, 15, 7,

56, 48, 40, 32, 24, 16, 8, 0,

58, 50, 42, 34, 26, 18, 10, 2,

60, 52, 44, 36, 28, 20, 12, 4,

62, 54, 46, 38, 30, 22, 14, 6

};

int E\_Table[] = { //扩展变换E

31, 0, 1, 2, 3, 4,

3, 4, 5, 6, 7, 8,

7, 8, 9, 10, 11, 12,

11, 12, 13, 14, 15, 16,

15, 16, 17, 18, 19, 20,

19, 20, 21, 22, 23, 24,

23, 24, 25, 26, 27, 28,

27, 28, 29, 30, 31, 0

};

int S\_Box[8][4][16] = { //8个s盒

{

{14, 4,13, 1, 2,15,11, 8, 3,10, 6,12, 5, 9, 0, 7},

{ 0,15, 7, 4,14, 2,13, 1,10, 6,12,11, 9, 5, 3, 8},

{ 4, 1,14, 8,13, 6, 2,11,15,12, 9, 7, 3,10, 5, 0},

{15,12, 8, 2, 4, 9, 1, 7, 5,11, 3,14,10, 0, 6,13}

},

{

{15, 1, 8,14, 6,11, 3, 4, 9, 7, 2,13,12, 0, 5,10},

{ 3,13, 4, 7,15, 2, 8,14,12, 0, 1,10, 6, 9,11, 5},

{ 0,14, 7,11,10, 4,13, 1, 5, 8,12, 6, 9, 3, 2,15},

{13, 8,10, 1, 3,15, 4, 2,11, 6, 7,12, 0, 5,14, 9}

},

{

{10, 0, 9,14, 6, 3,15, 5, 1,13,12, 7,11, 4, 2, 8},

{13, 7, 0, 9, 3, 4, 6,10, 2, 8, 5,14,12,11,15, 1},

{13, 6, 4, 9, 8,15, 3, 0,11, 1, 2,12, 5,10,14, 7},

{ 1,10,13, 0, 6, 9, 8, 7, 4,15,14, 3,11, 5, 2,12}

},

{

{ 7,13,14, 3, 0, 6, 9,10, 1, 2, 8, 5,11,12, 4,15},

{13, 8,11, 5, 6,15, 0, 3, 4, 7, 2,12, 1,10,14, 9},

{10, 6, 9, 0,12,11, 7,13,15, 1, 3,14, 5, 2, 8, 4},

{ 3,15, 0, 6,10, 1,13, 8, 9, 4, 5,11,12, 7, 2,14}

},

{

{ 2,12, 4, 1, 7,10,11, 6, 8, 5, 3,15,13, 0,14, 9},

{14,11, 2,12, 4, 7,13, 1, 5, 0,15,10, 3, 9, 8, 6},

{4, 2, 1, 11,10,13, 7, 8,15, 9,12, 5, 6, 3, 0,14},

{11,8,12, 7, 1,14, 2,13, 6,15, 0, 9,10, 4, 5, 3}

},

{

{12, 1,10,15, 9, 2, 6, 8, 0,13, 3, 4,14, 7, 5,11},

{10,15, 4, 2, 7,12, 9, 5, 6, 1,13,14, 0,11, 3, 8},

{ 9,14,15, 5, 2, 8,12, 3, 7, 0, 4,10, 1,13,11, 6},

{ 4, 3, 2,12, 9, 5,15,10,11,14, 1, 7, 6, 0, 8,13}

},

{

{ 4,11, 2,14,15, 0, 8,13, 3,12, 9, 7, 5,10, 6, 1},

{13, 0,11, 7, 4, 9, 1,10,14, 3, 5,12, 2,15, 8, 6},

{ 1, 4,11,13,12, 3, 7,14,10,15, 6, 8, 0, 5, 9, 2},

{ 6,11,13, 8, 1, 4,10, 7, 9, 5, 0,15,14, 2, 3,12}

},

{

{13, 2, 8, 4, 6,15,11, 1,10, 9, 3,14, 5, 0,12, 7},

{ 1,15,13, 8,10, 3, 7, 4,12, 5, 6,11, 0,14, 9, 2},

{ 7,11, 4, 1, 9,12,14, 2, 0, 6,10,13,15, 3, 5, 8},

{ 2, 1,14, 7, 4,10, 8,13,15,12, 9, 0, 3, 5, 6,11}

}

};

int IP\_1\_Table[64] = //逆初始置换IP^-1

{

39, 7, 47, 15, 55, 23, 63, 31,

38, 6, 46, 14, 54, 22, 62, 30,

37, 5, 45, 13, 53, 21, 61, 29,

36, 4, 44, 12, 52, 20, 60, 28,

35, 3, 43, 11, 51, 19, 59, 27,

34, 2, 42, 10, 50, 18, 58, 26,

33, 1, 41, 9, 49, 17, 57, 25,

32, 0, 40, 8, 48, 16, 56, 24

};

int P\_Table[32] = //置换运算P

{

15,6,19,20,

28,11,27,16,

0,14,22,25,

4,17,30,9,

1,7,23,13,

31,26,2,8,

18,12,29,5,

21,10,3,24

};

int PC\_1[56] =

{

56,48,40,32,24,16,8, //密钥置换PC\_1

0,57,49,41,33,25,17,

9,1,58,50,42,34,26,

18,10,2,59,51,43,35,

62,54,46,38,30,22,14,

6,61,53,45,37,29,21,

13,5,60,52,44,36,28,

20,12,4,27,19,11,3

};

int PC\_2[48] = //密钥置换PC\_2

{

13,16,10,23,0,4,

2,27,14,5,20,9,

22,18,11,3,25,7,

15,6,26,19,12,1,

40,51,30,36,46,54,

29,39,50,44,32,47,

43,48,38,55,33,52,

45,41,49,35,28,31

};

void Replacement(int arry1[],int arry2[],int arry3[],int num) //置换函数(初始IP，逆初始IP，

{

int i,tmp;

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

{

tmp=arry2[i];

arry3[i]=arry1[tmp];

}

}

int move\_times[16] = {1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1}; //对左移位的规定

void lif\_move(int arry1[],int arry2[],int n) //左移位实现函数

{

int i;

for(i=0;i<28;i++)

{

arry2[i]=arry1[(n+i)%28];

}

}

int K[16][48]; //存放16轮子密钥

int c[64]; //存放明文或密文

int L[17][32],R[17][32]; //存放加密过程中左右部分

void SubKey(int K0[64]) //子密钥产生函数

{

int i;

int K1[56],K2[56];

int C[17][28],D[17][28];

Replacement(K0,PC\_1,K1,56); //密钥置换PC\_1

for(i=0;i<28;i++) //将PC\_1输出的56比特分为左右两部分

{

C[0][i]=K1[i];

D[0][i]=K1[i+28];

}

i=0;

while(i<16)

{

int j;

lif\_move(C[i],C[i+1],move\_times[i]);

lif\_move(D[i],D[i+1],move\_times[i]);

for(j=0;j<28;j++)

{

K2[j]=C[i+1][j];

K2[j+28]=D[i+1][j];

}

Replacement(K2,PC\_2,K[i],48); //密钥置换PC\_2

i++;

}

/\*printf("\n子密钥生成过程中，左边生成的值:");

for(i=0;i<17;i++)

{

int j;

printf("\nC[%d]:",i);

for(j=0;j<28;j++)

{

if(j%7==0)

{

printf(" ");

}

printf("%d",C[i][j]);

}

}

printf("\n子密钥生成过程中，右边生成的值:");

for(i=0;i<17;i++)

{

int j;

printf("\nD[%d]:",i);

for(j=0;j<28;j++)

{

if(j%7==0)

{

printf(" ");

}

printf("%d",D[i][j]);

}

}\*/

}

void S\_compress(int arry[],int shc[]) //S盒压缩变换，其中数组shc存放经过s盒的结果

{

int h,l; //行，列

int sha[8]; //存放经过s盒的十进制结果

int i,j;

int temp[4];

for(i=0;i<8;i++) //s盒压缩变换

{

h=arry[(1+(i\*6))-1]\*2 + arry[(6+(i\*6))-1];

l =arry[(2+(i\*6))-1]\*8 + arry[(3+(i\*6))-1]\*4 + arry[(4+(i\*6))-1]\*2 + arry[(5+(i\*6))-1];

sha[i]=S\_Box[i][h][l];

}

for(i=0;i<8;i++)

{

for(j=3;j>=0;j--)

{

temp[j]=sha[i]%2;

sha[i]=sha[i]/2;

}

for(j=0;j<4;j++)

{

shc[4\*i+j]=temp[j];

}

}

/\*printf("\n第%d次s盒的输出:",m++);

for(i=0;i<32;i++)

{

if(i%8==0)

{

printf(" ");

}

printf("%d",shc[i]);

}\*/

}

void To10(int a[], int b[],int n)//二进制转十进制

{

int i,j;

int temp;

int arry[16][4];

for(i=0;i<n/4;i++)

{

for(j=0;j<4;j++)

{

arry[i][j]=a[4\*i+j];

}

}

for (i=0;i<n/4;i++)

{

temp=arry[i][0]\*8+arry[i][1]\*4+arry[i][2]\*2+arry[i][3]\*1;

/\*for(j=3;j>=0;j--)

{

if(arry[i][j]==1)

{

t=1;

for(k=0;k<3-j;k++)

{

t=t\*2;

}

temp+=t;

}

}\*/

b[i]=temp;

}

}

void To102(int a[], int b[],int n)//二进制转十进制

{

int i,j;

int temp;

int arry[8][8];

int t=1,k;

for(i=0;i<n/8;i++)

{

for(j=0;j<8;j++)

{

arry[i][j]=a[8\*i+j];

}

}

for (i=0;i<n/8;i++)

{

temp=0;

for(j=7;j>=0;j--)

{

if(arry[i][j]==1)

{

t=1;

for(k=0;k<7-j;k++)

{

t=t\*2;

}

temp+=t;

}

}

b[i]=temp;

}

}

void F\_Function(int a[32],int b[32],int n) //F函数

{

int i;

int tmp[48];

int tep[32];

Replacement(a,E\_Table,tmp,48); //扩展变换E

/\*printf("\n第%d轮E盒扩展结果:",n);

for(i=0;i<48;i++)

{

if(i%8==0)

printf(" ");

printf("%d",tmp[i]);

}\*/

for(i=0;i<48;i++) //与子密钥异或

{

tmp[i] ^= K[n][i];

}

/\*printf("\n进入S盒的48比特:");

for(i=0;i<48;i++)

{

if(i%6==0)

{

printf(" ");

}

printf("%d",tmp[i]);

}\*/

S\_compress(tmp,tep); //压缩变换S

Replacement(tep,P\_Table,b,32); //置换运算P

/\*printf("\n第%d次P置盒输出:",l++);

for(i=0;i<32;i++)

{

if(i%8==0)

printf(" ");

printf("%d",b[i]);

}\*/

/\*printf("\nf[%d]的输出结果:",n);

for(i=0;i<32;i++)

{

if(i%8==0)

{

printf(" ");

}

printf("%d",b[i]);

}\*/

}

void Encryption(int m0[64],int c1[64])

{

int i,k;

int arry[32];

int c0[64],m1[64];

Replacement(m0,IP\_Table,m1,64); //初始置换IP

/\*printf("\n初始置换:");

for(i=0;i<64;i++)

{

if(i%8==0)

printf(" ");

printf("%d",m1[i]);

}\*/

for(i=0;i<32;i++)

{

L[0][i]=m1[i];

R[0][i]=m1[i+32];

}

k=1;

while(k<17)

{

F\_Function(R[k-1],arry,k-1);

for(i=0;i<32;i++)

{

L[k][i]=R[k-1][i];

R[k][i]=L[k-1][i]^arry[i];

}

k++;

}

for(i=0;i<32;i++)

{

c0[i]=R[16][i];

c0[i+32]=L[16][i];

}

Replacement(c0,IP\_1\_Table,c1,64); //逆初始置换

}

void changeKey(int a[16][48])

{

int i,j;

int tmp[16][48];

for(i=0;i<16;i++)

{

for(j=0;j<48;j++)

{

tmp[i][j]=a[i][j];

}

}

for(i=0;i<16;i++)

{

for(j=0;j<48;j++)

{

K[i][j]=tmp[15-i][j];

}

}

}

void Decryption(int c1[],int m[])

{

int c0[64],t[64];

int i,k;

int arry[32];

changeKey(K);

/\*printf("\n交换后的密钥:\n");

for(i=0;i<16;i++)

{

printf("\n");

for(j=0;j<48;j++)

{

if(j%8==0)

{

printf(" ");

}

printf("%d",K[i][j]);

}

}\*/

Replacement(c1,IP\_Table,c0,64); //初始IP

for(i=0;i<32;i++)

{

L[0][i]=c0[i];

R[0][i]=c0[i+32];

}

k=1;

while(k<17)

{

F\_Function(R[k-1],arry,k-1);

for(i=0;i<32;i++)

{

L[k][i]=R[k-1][i];

R[k][i]=L[k-1][i]^arry[i];

}

k++;

}

for(i=0;i<32;i++)

{

t[i]=R[16][i];

t[i+32]=L[16][i];

}

Replacement(t,IP\_1\_Table,m,64); //逆初始置换

}

void print() //输出函数

{

int i;

int a[12],b[12],d[3][16];

int p[64],q[64];

for(i=0;i<32;i++)

{

p[i]=L[15][i];

p[32+i]=R[15][i];

q[i]=R[16][i];

q[i+32]=L[16][i];

}

To10(K[14],a,48);

To10(K[15],b,48);

To10(c,d[0],64);

To10(p,d[1],64);

To10(q,d[2],64);

printf("\n\t\t15轮密钥:");

for(i=0;i<12;i++)

{

printf("%x",a[i]);

}

printf("\t15轮结果:");

for(i=0;i<16;i++)

{

printf("%X",d[1][i]);

}

printf("\n\t\t16轮密钥:");

for(i=0;i<12;i++)

{

printf("%X",b[i]);

}

printf("\t16轮结果:");

for(i=0;i<16;i++)

{

printf("%X",d[2][i]);

}

printf("\n\t\t最后结果:");

for(i=0;i<16;i++)

{

printf("%X",d[0][i]);

}

/\*printf("\n最后结果二进制:");

for(i=0;i<64;i++)

{

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

}\*/

}

int main()

{

int num,i,t;

char s1[8],s2[8];

int m[64]; //m存放二进制明文/密文

int d[64]; //存放二进制密钥

int s[8];

show1();

printf("\t\tinput you choice:");

while(1)

{

scanf("%d",&num);

switch(num)

{

case 1:

system("cls"); //调用清屏函数

show2();

{

scanf("%d",&num);

if(num==1)

{

while(strlen(s1)!=8)

{

printf("\t\t请输入明文(8):");

scanf("%s",s1);

}

To2Bin(s1,m); //将明文转换成二进制流

while(strlen(s2)!=8)

{

printf("\t\t请输入密钥(8):");

scanf("%s",s2);

}

To2Bin(s2,d); //将密钥转换成二进制

/\*printf("\n初始二进制明文:");

for(i=0;i<64;i++)

{

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

}\*/

/\*printf("\n初始二进制密钥:");

for(i=0;i<64;i++)

{

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

}\*/

SubKey(d);

/\*printf("\n16轮子密钥如下:");

for(i=0;i<16;i++)

{

printf("\nK[%d]:",i+1);

for(j=0;j<48;j++)

{

if(j%8==0)

printf(" ");

printf("%d",K[i][j]);

}

}\*/

Encryption(m,c);

/\*printf("\n16次迭代左结果:");

for(i=0;i<17;i++)

{

printf("\nL[%d]:",i);

for(j=0;j<32;j++)

{

if(j%8==0)

{

printf(" ");

}

printf("%d",L[i][j]);

}

}

printf("\n16次迭代右结果:");

for(i=0;i<17;i++)

{

printf("\nR[%d]:",i);

for(j=0;j<32;j++)

{

if(j%8==0)

{

printf(" ");

}

printf("%d",R[i][j]);

}

}\*/

print();

/\*printf("\n二进制密文:");

for(i=0;i<64;i++)

{

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

}\*/

printf("\n\t\t按0将此密文解密,1返回上一层,2返回主界面,其他键退出.....");

scanf("%d",&t);

if(t==0)

{

int s[8];

int a[64];

Decryption(c,a);

/\*printf("\n解密后的二进制:");

for(i=0;i<64;i++)

{

printf("%d",a[i]);

}\*/

To102(a,s,64);

printf("\n\t\t解密结果:");

for(i=0;i<8;i++)

{

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

}

printf("\n\t\t按1加密,2解密,3退出");

}

else if(t==1)

{

system("cls");

show2();

}

else if(t==2)

{

system("cls");

show1();

}

else

{

exit(0);

}

}

else if(num==2)

{

char str1[20],str2[20];

printf("\t\t请输入明文文件名:");

scanf("%s",str1);

reader(str1,s1);

To2Bin(s1,m);

printf("\t\t请输入密钥文件名:");

scanf("%s",str2);

reader(str2,s2);

To2Bin(s2,d);

SubKey(d);

Encryption(m,c);

print();

printf("\n\t\t按0将此密文解密,1将密文存入文件,2返回主界面,其他键退出.....");

scanf("%d",&t);

if(t==0)

{

int a[64];

Decryption(c,a);

/\*printf("\n解密后的二进制:");

for(i=0;i<64;i++)

{

printf("%d",a[i]);

}\*/

To102(a,s,64);

printf("\n\t\t解密结果:");

for(i=0;i<8;i++)

{

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

}

printf("\n\t\t按1加密,2解密,3退出");

}

else if(t==1)

{

int a[16];

char str[30];

FILE \*fw;

printf("\n\t\t请输入文件名:");

scanf("%s",str);

fw=fopen(str,"w");

if(fw==NULL)

{

printf("\n\t\t文件打开失败!\n");

}

else

{

To10(c,a,64);

for(i=0;i<16;i++)

{

fprintf(fw,"%X",a[i]);

}

}

fclose(fw);

printf("\n\t\t密文保存成功!按1加密,2解密,3退出");

}

else if(t==2)

{

system("cls");

show1();

}

else

{

exit(0);

}

}

else if(num==3)

{

system("cls");

exit(0);

}

else

{

printf("\n\t\t输入不正确，请重新输入:");

}

}

break;

case 2:

{

system("cls");

printf("\n\n\t\t------------------DES解密----------------");

while(strlen(s1)!=8)

{

printf("\n\n\n\t\t请输入密文(8):");

scanf("%s",s1);

}

To2Bin(s1,m);

while(strlen(s2)!=8)

{

printf("\t\t请输入密钥(8):");

scanf("%s",s2);

}

To2Bin(s2,d);

SubKey(d);

Decryption(m,c);

print();

printf("\n\t\t按1返回上一层,2返回主界面,其他键退出.....");

scanf("%d",&t);

if(t==1)

{

system("cls");

printf("\n\n\t\t---------------DES解密-------------");

while(strlen(s1)!=8)

{

printf("\n\n\t\t请输入密文(8):");

scanf("%s",s1);

}

To2Bin(s1,m);

while(strlen(s2)!=8)

{

printf("\t\t请输入密钥(8):");

scanf("%s",s);

}

To2Bin(s2,d);

SubKey(d);

Decryption(m,c);

print();

}

else if(t==2)

{

system("cls");

show1();

}

else

{

exit(0);

}

}

break;

case 3:

system("cls");

exit(0);

default:

printf("输入不正确，请重新输入:");

}

}

return 0;

}