128 位 RSA 与 8 轮 DES 加密体系

一、算法概要

DES 是一种按分组密码工作的对称密码体制，以 64 位为分组长度，64 位一组的明文作为算法的输入，通过一系列主要是**换位**和**置换**的复杂操作，输出同样 64 位长度的密文。加密过程主要由**五部分**组成：

1） 异或初始向量 IV/前一组密文（图中未画出）
2） 初始 IP 置换
3） 子密钥 Ki 的获取
4） 基于轮函数 F 的 8 轮迭代
5） 尾置换 IP^{-1}

传统的对称（单钥）密码体制在加密解密时使用同一个密钥；在通信之前必须提前获得密钥；无法通过电子签名对发方和收方的身份进行证实。所以提出了非对称（双钥）密码体制：乙方生成两把密钥（公钥和私钥），公钥是公开的，任何人都可以获得，私钥则是保密的；甲方获取乙方的公钥，然后用它对信息加密；乙方得到加密后的信息，用私钥解密。如果设计好单向函数，使得公钥加密的信息只有私钥解得开，那么只要私钥不泄漏，通信就是安全的。**RSA** 是比较成功的公钥密码体制，**理论基础**是：要求两个大素数的乘积是容易的，但是分解一个合数为两个大素数的乘积则在计算上是不可能的。**步骤**如下：

1） 寻找两个大素数 p, q
2） 计算出 n=p*q 和 φ(n) = (p-1) * (q-1)
3） 选择一个随机数 d（0<d<φ(n)），满足(d, φ(n))=1
4） 计算 d 关于 φ(n) 的逆模 e，即满足 ed≡1 mod φ(n)
5） 于是得到密钥对(e, d, n)，公钥是(e, n)，私钥是(d, n)，B 要公开 n 和 e，保存 d
6） 用户 A 向 B 发送信息 m 时，利用 B 的公钥加密明文 m：c≡m^e mod n
用户 B 收到 c 后，用自己的私钥解密密文 c：m≡c^d mod n

二、实现思路

大数库

理论上，RSA 的安全性取决于模 n 分解的困难性，因此选择越大的 n，则越难被攻破。C++语言没有定义大数库，需要引入第三方库。选择在 vs2017 下调用 miracl 函数库。

首先下载并解压 miracl 压缩包, 创建新文档 miracl, 将解压包中所有单文件拷贝到 miracl 目录中。创建空项目 CompileMiracl，将 miracl 文件夹中所有文件拷贝到 CompileMiracl 的工程目录，添加现有项中部分头文件和源文件，将项目的目标文件扩展名改为静态库(.lib)，生成项目，在工程目录得到 CompileMiracl.lib 和 CompileMiracl.pdb

使用时只需将 miracl.h，mirdef.h 以及 miracl.lib (CompileMiracl.lib)，miracl.pdb (CompileMiracl.pdb) 添加到新项目的工程目录, 即可通过 miracl 中大数结构 big 调用 miracl 库中的函数, 例如 mr_compare()用于比较大小, copy()用于拷贝赋值, add()为加法，multiply()为乘法，divide()为除法等，下面是 RSA 算法三个主要函数，miracl 库中有也相应实现，这里按照函数功能自行编写

文件及函数

1） const.h

仅用于 DES 算法各变换矩阵的定义

2） miracle.h 和 mirdef.h

miracl 库头文件

3） rsa.h

由于代码已进行注释来帮助理解，这里只进行大体功能介绍

RSA 三个主要函数：

- **void pow_mod (big a, big b, big n, big z)**

  快速幂模算法其实是用到了二分的思想，把 b 按照二进制展开

  b = p(n)*2^n + p(n-1) *2^(n-1) +…+ p(1) *2 + p(0)。其中 p(i) (0<=i<=n)为 0 或 1

  a^b = a^ (p(n)*2^n + p(n-1)*2^(n-1) +…+ p(1)*2 + p(0))

=a^(p(n)*2^n) * a^(p(n-1) *2^(n-1)) *…* a^(p(1)*2) * a^p(0)

对于 p(i)=0 的情况不用处理，因为 a^(p(i) * 2^(i-1)) = a^0 = 1；需要考虑的仅仅是 p(i)=1 的情况，化简得：a^(2^i) = a^(2^(i-1) * 2) = (a^(p(i) * 2^(i-1))) ^2

- **bool is_prime (big num, ine times)**

对于奇整数 num 的 **Miller-Rabin 素性测试算法**如下:

(1)令 num-1=2^km，m 是奇数;

(2)对 i=1 to times do

①选择随机整数 a(2≤a≤num-2);

②计算 b≡a^mmodnum

③If b≡1 mod num，then num 是素数 and 返回,else 令 j=1

④If j=k then num 是合数，算法终止

⑤If b≡-1 mod num, then num 是素数 and 返回，

else 计算 b=b^2 mod num, j=j+1,goto ④

- **big extended_Euclid (big a, big b, big &x, big &y)**

扩展欧几里得算法：对于不完全为 0 的非负整数 a，b，gcd（a，b）表示 a，b 的最大公约数，必然存在整数对 x，y，使得 gcd（a，b）=ax+by

用于求 a 模 b 的逆元时调用形式为（a,b,x,y）x 的返回值即为 a 的逆元

其它函数：

- **void create_prime(int len, big p, int times)**

  创建素数的顶层函数，调用 bigdig()，随机产生长度为 len 的二进制大数 p，直到 is_prime()判断 p 为素数

- **void my_sub(big a, big b, big z)**

  定义减法，调用 add()计算 a 加上 b 的相反数(negnify(b))

4） main.c

首先是 DES 需要用到的操作（具体见“8 轮 DES 实现说明”）然后是 **main 函数：**

① 对话界面给出密钥、文件地址等的输入模式规定

② 调用 rsa.h 中实现的函数生成用于 RSA 算法的素数和密钥对，并输出

③ 提示用户选择工作模式：加密 or 解密

④ **加密模式：**

- 用户输入密钥（按照前面的规定应为 64 位二进制序列，为增强容错性，此处立即检查读取的字符串长度，如果不为 64，提示错误，结束此次操作）
- 使用 cinstr()函数把 string 转换成 big 类型，然后调用 pow_mod(key, e, n, enc_key)对初始密钥进行 RSA 加密
- 用户输入明文文件名（绝对路径或相对路径）和希望输出到的密文文件名（绝对路径或相对路径），程序进行检查，若文件不存在则退出
- 打开文件以后先将前面 RSA 加密后的密钥使用 cotstr()转换成 string 类型，输出到密文文件中
- 将明文文件中二进制串读入到一个 string 型变量中，使用 StringToBitset()依次存入 Bitset<64>型变量中（本实验中要求加密时输入的文件由一行二进制数构成）
- 因为采用 CBC 模式，所以需要一个初始向量 IV，本实验中使用时间作为种子随机生成一个 64 位二进制序列作为 IV 并加密输出到密文文件中
- 将明文按照 64 位一组分组，若明文长度不是 64 的整数倍，需要在最后一组添加补充位，本实验中采用零填充，最后 8 位需要空余出来以保存填充的位数（方便解密后的填充位舍弃）。如果出现最后一组位数大于等于 56 位（空余 8 位的情况），则添加新的一组（64 位），以保证最后 8 位用于保存填充位数（本实验中填充位数不包括最后 8 位，count>=0&&count<=56）
- 第一组明文先与 IV 模 2 相加（即异或运算）再加密；后面各组明文先与前一组的密文模 2 相加，再加密
- 各组密文生成后即输出到指定文件中

⑤ **解密模式：**

与解密类似，但互为逆操作

- 用户输入密文文件名（绝对路径或相对路径）和希望输出到的明文文件名（绝对路径或相对路径），程序进行检查，若文件不存在则退出（本实验中要求解密时输入的文件由三行构成，第一行为 RSA 加密后的密钥，第二行为 DES 加密后的 IV，第三行为需要解密的密文）
- 打开文件后先将第一行的 string 读入转换成 big 类型的 big_key4，然后调用 pow_mod(big_key4, d, n, dec)解密，终端输出解密后的密钥
- 将第二行的 IV 读入解密转换，再将第三行的密文读入并转换
- 先把最后一组解密、与前一组密文异或得到原文、从最后 8 位得到填充的位数（二进制转十进制）
- 然后从第一组开始解密、异或、输出原文，注意最后一组舍掉之前的填充位，考虑到之前加密时提到的情况，也可能会舍弃掉整个最后一组，倒数第二组也输出一部分

⑥ 加密/解密完成后，提示成功!

三、测试分析

工作模式选择 0 时**加密**，输入 64 位初始密钥，encrypt.txt 作为输入的明文文件（只包括需要加密的明文），ciper.txt 作为输出的密文文件（包括三行二进制数，分别为 RSA 加密后的密钥，DES 加密后的 IV，DES 加密后的明文）

工作模式选择 1 时**解密**，ciper.txt 作为输入的密文文件，plain.txt 作为输出的明文文件

```
------------------------8轮DES文件加密----------------------------------
工作模式:                                   加密为0/解密为1
密钥:                                       长度为64位的二进制0/1串
输入文件地址:                               绝对地址/相对地址
输出文件地址:                               绝对地址/相对地址
------------------------------------------------------------------------

正在生成用于RSA算法的素数和密钥对...
生成成功!
p=10001010110011101101110000011011110111100011101100111111110110111
q=11111111001110001001110110110000101110011011101101111111100100001
n=10001010011000101100000000000000100110000101111100110011001101101000001000100111010111000101011000011011101011111011111110100111
fn=10001010011000101100000000000000100110000101111100110011001101001111100000100000001101000101111001110101111000010010000001100000
d=11011100000001111110100100010010101011011000100011001111001110101101011001101101111111010011011110001110010001010001101010000010001
e=10101000000101000000111100001010001010011001111001110110001011010110100000010010111010001110000010011100011100110111110001
请选择工作模式: 0
请输入初始密钥: 1100100000111111101010010010011010101110110110111010011111100100
初始密钥: 1100100000111111101010010010011010101110110110111010011111100100
加密后: 1001101110111011101111011100110110111001011100000010011010110110111111111100100001010111010110010010010010001010010110111110
请输入明文文件名(.txt): encrypt.txt
请输入密文文件名(.txt): ciper.txt

已成功打开文件!
开始加密...
encrypt.txt-->ciper.txt
加密成功!

请按任意键继续. . .
```

encrypt.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1001000110110011101111111100010010110100100001011100010010111100111001000110110011110111100111001000
1101100111101111111000100101101001000010111000100101111001110010001101100111101111100010010110100100001011100010010111100111
```

ciper.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1001101110111011101111011100110110111001011100000010011010110110111111111100100001010111010110010010010010001010010110111110
1100101011011011101111001100001010100001110111100110000101110001
0010000100001110111101001111011101100000101111100010101111110001000001100010100100100101110000100110010111000111000110011110111000000111
0011100111111000100000101101110000100010110110111100011101011000100100110001110001100011010001000101000011100101101111000100001111000111
1010101111000110010110101001000010100111100101
```

```
------------------------8轮DES文件加密----------------------------------
工作模式:                                   加密为0/解密为1
密钥:                                       长度为64位的二进制0/1串
输入文件地址:                               绝对地址/相对地址
输出文件地址:                               绝对地址/相对地址
------------------------------------------------------------------------

正在生成用于RSA算法的素数和密钥对...
生成成功!
p=10001010110011101101110000011011110111100011101100111111110110111
q=11111111001110001001110110110000101110011011101101111111100100001
n=10001010011000101100000000000000100110000101111100110011001101101000001000100111010111000101011000011011101011111011111110100111
fn=10001010011000101100000000000000100110000101111100110011001101001111100000100000001101000101111001110101111000010010000001100000
d=11011100000001111110100100010010101011011000100011001111001110101101011001101101111111010011011110001110010001010001101010000010001
e=10101000000101000000111100001010001010011001111001110110001011010110100000010010111010001110000010011100011100110111110001
请选择工作模式: 1
请输入密文文件名(.txt): ciper.txt
请输入明文文件名(.txt): plain.txt

已成功打开文件!
解密后的密钥: 1100100000111111101010010010011010101110110110111010011111100100
开始解密...
ciper.txt-->plain.txt
解密成功!

请按任意键继续. . .
```

encrypt.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1001000110110011101111111100010010110100100001011100010010111100111001000110110011110111100111001000
1101100111101111111000100101101001000010111000100101111001110010001101100111101111100010010110100100001011100010010111100111
```

ciper.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1001101110111011101111011100110110111001011100000010011010110110111111111100100001010111010110010010010010001010010110111110
1100101011011011101111001100001010100001110111100110000101110001
0010000100001110111101001111011101100000101111100010101111110001000001100010100100100101110000100110010111000111000110011110111000000111
0011100111111000100000101101110000100010110110111100011101011000100100110001110001100011010001000101000011100101101111000100001111000111
1010101111000110010110101001000010100111100101
```

plain.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1001000110110011101111111100010010110100100001011100010010111100111001000110110011110111100111001000
1101100111101111111000100101101001000010111000100101111001110010001101100111101111100010010110100100001011100010010111100111
```