8 轮 DES 实现说明

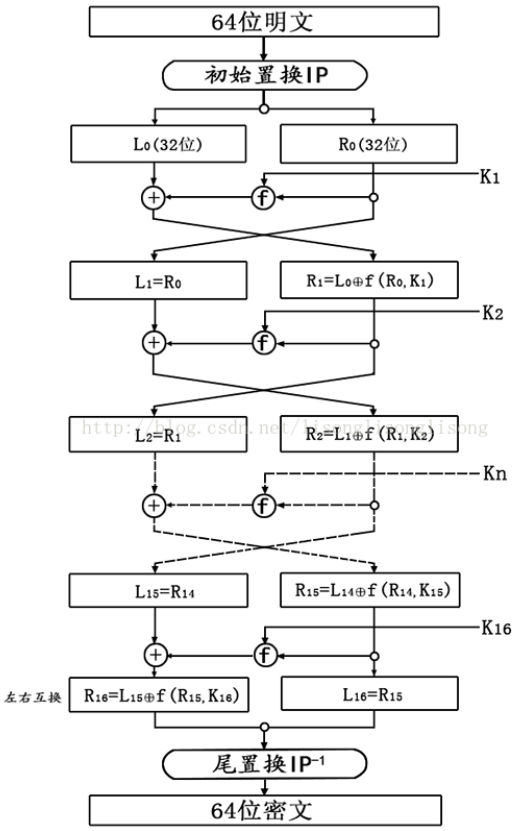
一、DES 算法概要

DES 是一种典型的**分组加密**算法：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。DES 采用对称密码体制，加密和解密使用相同的密钥，其所有保密性均依赖于密钥。

分组密码有多种工作方式，CBC 模式通过预制一初始向量 IV（IV 同样加密传输），每一组明文加密之前先与前一组的密文模 2 相加后再加密的方法，获得反馈使得输出的密文组与以前的明文组相关，较好实现了隐蔽明文、扩散明文效应的作用。

二、实验思路

DES 算法的**主流程**如下图所示



从图中可看出，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}

其中第二、三部分是算法的核心。

数据结构及数据类型

- **一维数组：**各置换表（置换 IP、置换 IP^{-1}、扩展置换 E、置换 P、压缩置换 IPC、循环左移位数表 LS、压缩置换 PC），线性表便于使用和管理
- **两维数组：**group[8][6]存储进入 S 盒的八个 6 比特矩阵，八个盒子内部实现 6 位到 4 位的压缩置换，便于分组处理
- **三维数组：**S 盒[8][4][16]存储 8 个矩阵，便于分组批量处理
- **bitset<n>**是一种类似数组的结构，每一个元素只能是 0 或 1，仅占用 1bit 的空间，相比使用 int 型数组或者 char 型数组空间复杂度低很多，并且异或等操作方便很多。但使用时需注意与 string 之间的相互转化，以及由于小端存储方式带来的下标顺序问题（本次实验中选择 bitset 主要考虑空间节约，没有用到其成员函数，所以采用正序存储，即读入的 64 位 string 存入 bitset<64>时，采用 bits[i]=string[i]赋值）

文件及函数

整个程序分两个文件:

1) **const.h**

   仅用于各变换矩阵的定义

2) **des.cpp**

   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

- **leftshift()**

  循环左移根据置换表移位即可，需要注意左移时要用下标小（位于左侧的）的替换下标较大的（位于右侧的）

- **generate_keys()**

  首先将输入的 64 位密钥去掉奇偶校验位得到 56 位密钥，然后分为高低 28 位 C、D 两组，分别左移后合并成 56 位，压缩置换得 48 位子密钥。因为本实验只进行 8 轮迭代，所以只需要生成 8 个子密钥

- **S_func()**

  48 位分成 8*6 的矩阵, 由 6 位二进制序列决定要查找元素在 S 盒中的行数和列数，找到的十进制数转换成四位二进制数即为该盒的输出，8 组 4 位数合并得到 48 位输出

- **F_func()**

  轮函数的输入为 32 位矩阵 R 和 48 位子密钥 K，首先进行扩展置换变为 48 位 expandedR，与 K 进行异或运算，将 48 位结果输入到 S 盒中，运算合并后得到 32 位矩阵，P 置换后得到 F 函数返回值

- **enc_dec()**

  加解密函数，输入为 64 位明文（或密文），首先进行 IP 下标置换，按高低 32 位分为 L 和 R，然后进行 8 轮迭代，每次迭代 newL = R； newR = L ^ F(R, K)；最后合并成 64 位再置换输出

- **StringToBitset()**, **BitsetToString()**

  bitset<64>和 64 位 string 之间的转换

- **main()**

  主函数完成该加密解密器的最后封装：

  ① 对话界面给出密钥、文件地址等的输入模式规定

  ② 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）

  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
  - 打开文件以后先将二进制串读入到一个 string 型变量中，然后使用 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整行二进制数构成）
  - 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两行构成，第一行为指定的 IV，第二行为需要解密的密文）
  - 打开文件以后先将第一行的 IV 读入解密转换，再将第二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

注意：本实验中密文文件格式为第一行加密后的 IV，第二行密文，需要换行！

下面分三种情况测试，包括<=56 位，>56&&<=64 位，>64 位，为了检验程序正确性，先加密后解密，对比解密出来的明文和原明文，若一致，则正确。（decrypt.txt 和 ciper 其实是相同的文件）

1） 36 位明文

需要填充 28 位

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

请输入密钥: 1011001101010011001010011010101000110011001010100101010101010000
请选择工作模式: 0
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)

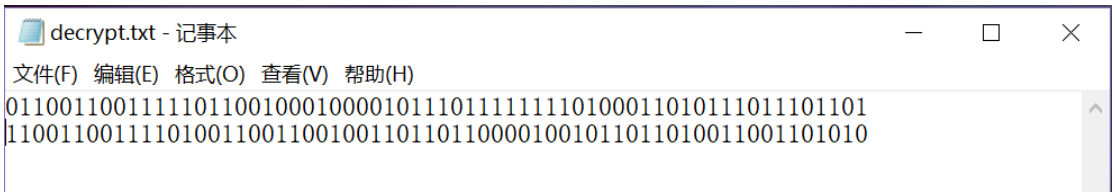
```
01100101011101001010010101010010101O
```

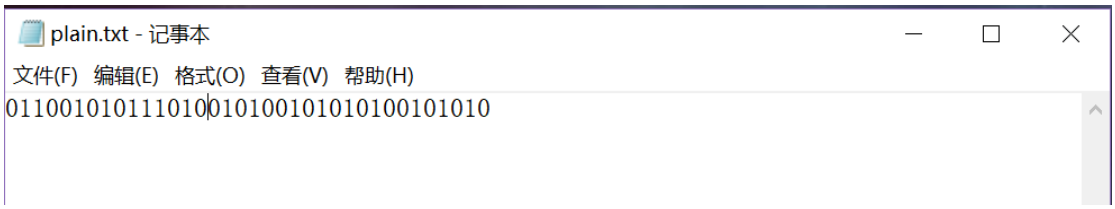
ciper.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
0110011001111101100100010000101110111111111010001101011101110110 1
1100110011110100110011001001101101100001001011011010011001101010
```

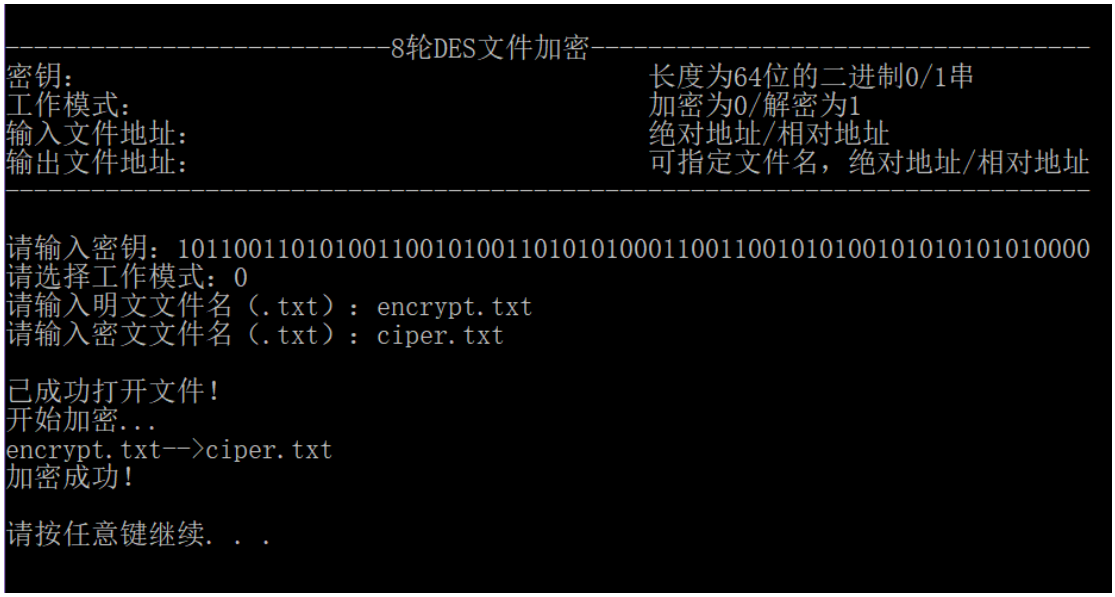
```
--------------------------8轮DES文件加密-----------------------------------
密钥:                                           长度为64位的二进制0/1串
工作模式:                                       加密为0/解密为1
输入文件地址:                                   绝对地址/相对地址
输出文件地址:                                   可指定文件名，绝对地址/相对地址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

请输入密钥: 1011001101010011001010011010101000110011001010100101010101010000
请选择工作模式: 1
请输入密文文件名（.txt）: decrypt.txt
请输入明文文件名（.txt）: plain.txt

已成功打开文件!
开始解密...
decrypt.txt-->plain.txt
解密成功!

请按任意键继续. . .
```

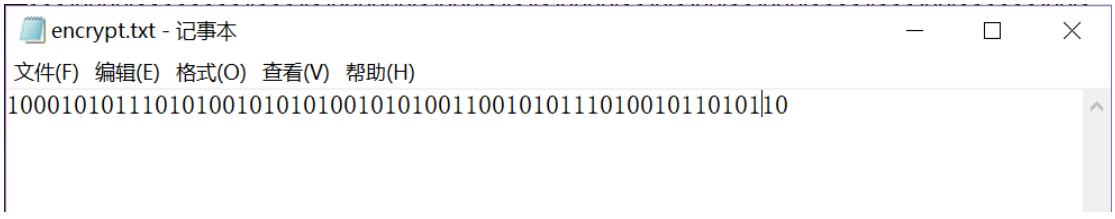


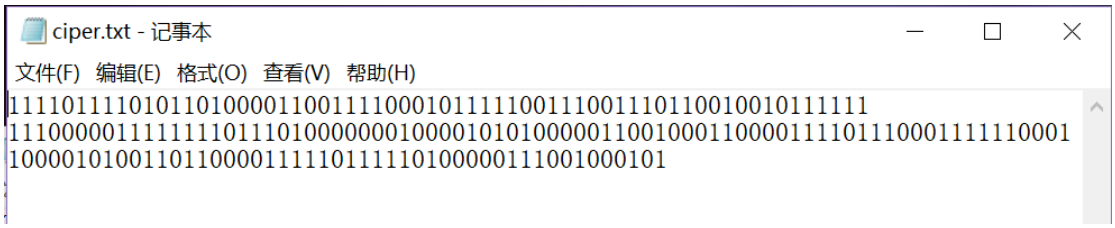


2） 58 位明文

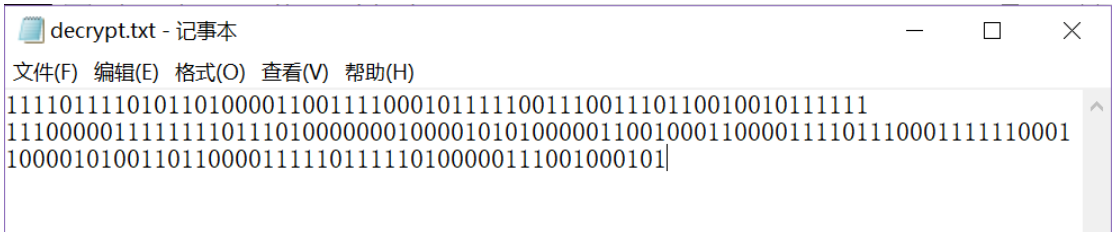
需要填充 62 位（多出 64 位）

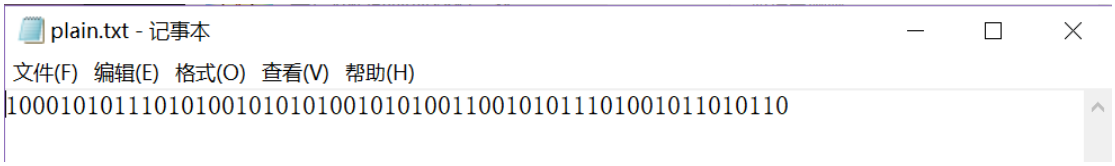






```
--------------------------8轮DES文件加密-----------------------------------
密钥:                                                长度为64位的二进制0/1串
工作模式:                                            加密为0/解密为1
输入文件地址:                                        绝对地址/相对地址
输出文件地址:                                    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

请输入密钥：1011001101010011001010011010101000110011001010100101010101010000
请选择工作模式：1
请输入密文文件名（.txt）：decrypt.txt
请输入明文文件名（.txt）：plain.txt

已成功打开文件！
开始解密...
decrypt.txt-->plain.txt
解密成功！

请按任意键继续. . .
```





3） 260 位明文

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

请输入密钥：1100100000111111101010010010011010101110110110111010011111100100
请选择工作模式：0
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)

```
1001000110110011110111111000100101101001000010111000100101111001110010001101100
1111011111100010010110100100001011100010010111100111001000110110011110111111000
1001011010010000101110001001011110011100100011011001111011111100010010110100100
0010111000100101111 0011
```

ciper.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1000010101100110010101100001010010010101001010110110011010011110
0110111010110011000111100010000101010100010010110010110000011110001010111101001
1110100010010110110010000010100001101100001010000100100111100111101100011001010
0011111101000111111010011110101011001101000010100101010101100001001100010100101
0011011001110000011010010010101010101100101010011111111101101111011110101111111
1110
```

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

请输入密钥: 1100100000111111101010010010011010101110110110111010011111100100
请选择工作模式: 1
请输入密文文件名（.txt）: decrypt.txt
请输入明文文件名（.txt）: plain.txt

已成功打开文件!
开始解密...
decrypt.txt-->plain.txt
解密成功!

请按任意键继续. . .
```

decrypt.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1000010101100110010101100001010010010101001010110110011010011110
0110111010110011000111100010000101010100010010110010110000011110001010111101001
1110100010010110110010000010100001101100001010000100100111100111101100011001010
0011111101000111111010011110101011001101000010100101010101100001001100010100101
0011011001110000011010010010101010101100101010011111111101101111011110101111111
1110
```

plain.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1001000110110011110111111000100101101001000010111000100101111001110010001101100
1111011111100010010110100100001011100010010111100111001000110110011110111111000
1001011010010000101110001001011110011100100011011001111011111100010010110100100
0010111000100101111 0011
```