## 算法原理概述

DES算法是块加密算法，明文和密文的分组长度为64位，加密方和解密方使用相同的密钥，长度也为64位，算法加密主要是通过换位和置换的方法使得信息充分混淆和充分扩散，达到加密的效果，解密与加密使用相同的函数，不过子密钥的输入顺序是相反的，通过与子密钥再次异或来逐次还原信息。

明文分组结构：*M*=*m*1​*m*2​…*m*64​,*mi*​∈{0,1},*i*=1..64.

密文分组结构：*C*=*c*1​*c*2​…*c*64​,*ci*∈{0,1},*i*=1..64.

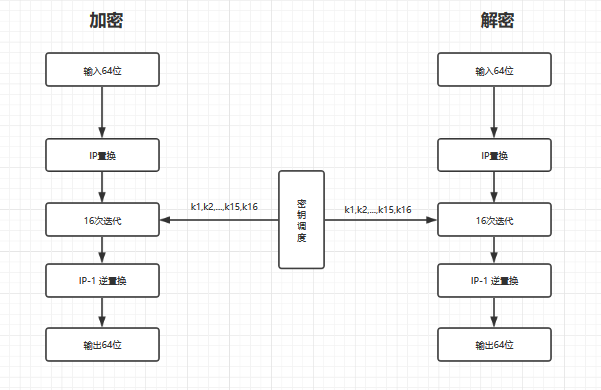
密钥分组结构：*K*=*k*1​*k*2​…*k*64​,*ki*​∈{0,1},*i*=1..64.

但在密钥中，*k*8​,*k*16​,*k*24​,*k*32​,*k*40​,*k*48​,*k*56​,*k*64​

作为奇偶校验位，不参与加密过程，因此仅56位起作用

## 总体结构

加密和解密过程仅密钥调度顺序不一样，加密过程调度顺序为 k1,k2,...,k16,解密过程调度顺序为k16,k15,...,k2,k1,



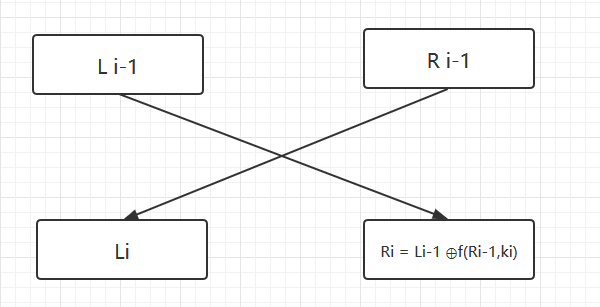
### 加密过程：*C*=*Ek*​(*M*)=*IP*−1⋅*T*16​⋅*T*15​⋅…⋅*T*1​⋅*IP*(*M*).

◌ IP 为初始置换；

◌ IP-1 是IP 的逆置换；

◌ 是一系列的迭代变换。

将IP置换得到的字符串分成。分别为前32位和后32位，进行如下迭代：



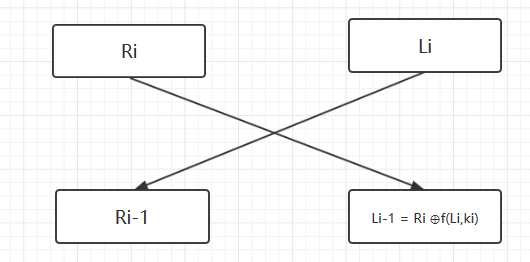
## 解密过程：*C*=*Ek*​(*C*)=*IP*−1⋅*T*1​⋅*T*2​⋅…⋅*T*16​⋅*IP*(*C*).

◌ IP 为初始置换；

◌ 是IP 的逆置换；

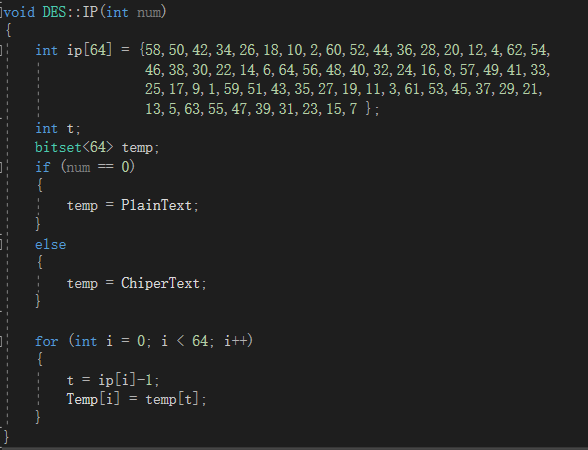
◌ 是一系列的迭代变换。

将IP置换得到的字符串分成。分别为前32位和后32位，进行如下迭代：

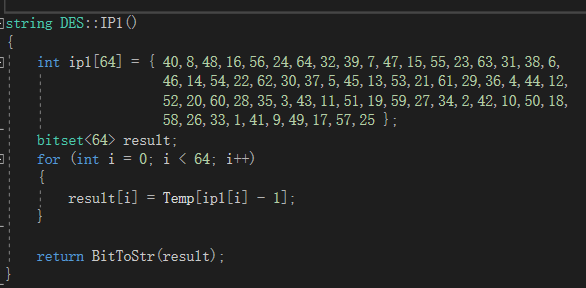


**模块分解：**

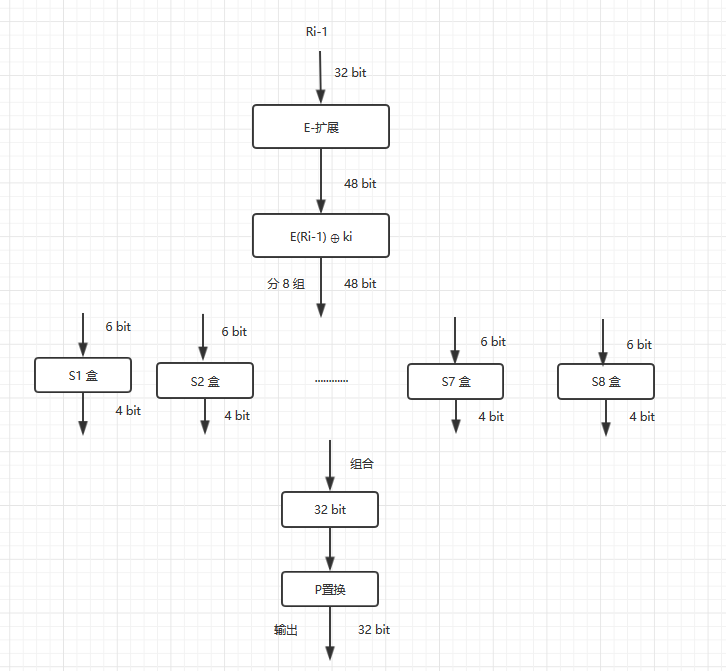
**IP模块:**



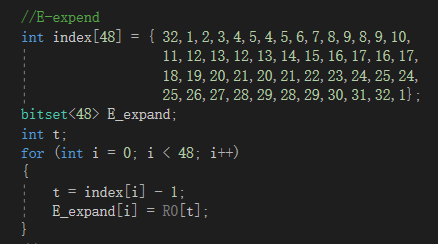
**模块：**



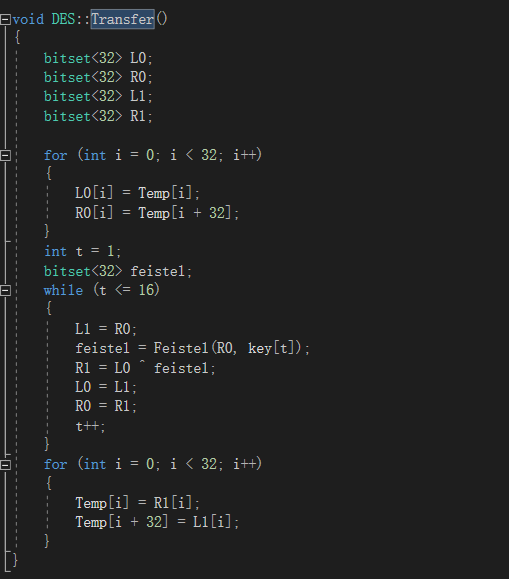
**迭代模块：**

****

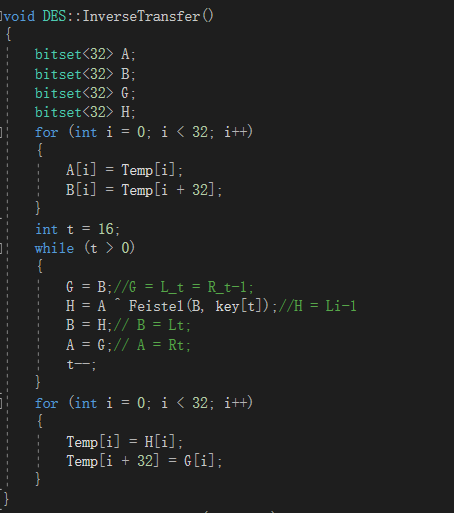
**E-扩展：**



**加密过程的迭代：**



**解密过程的迭代：**

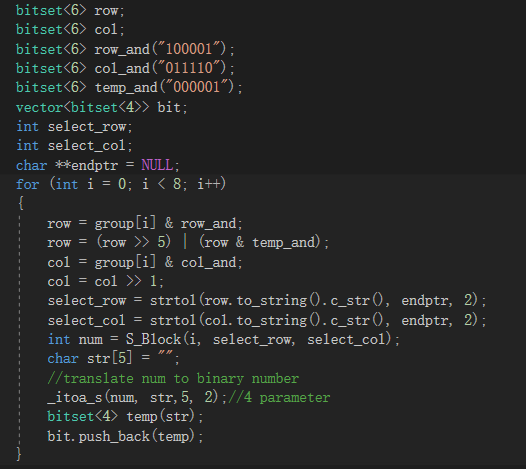


加密过程盒解密过程的子密钥调度顺序是相反的

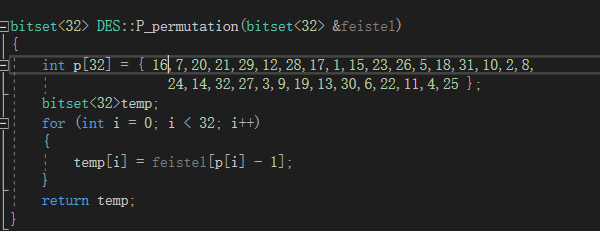
扩展后的二进制串与子密钥异或：



分组后，每组与对应S-盒进行6-4压缩操作，其中s-盒的数据采用文件读取方式来获得，而非初始化一长串数组：

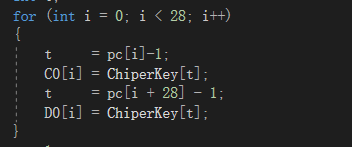


**P置换：**

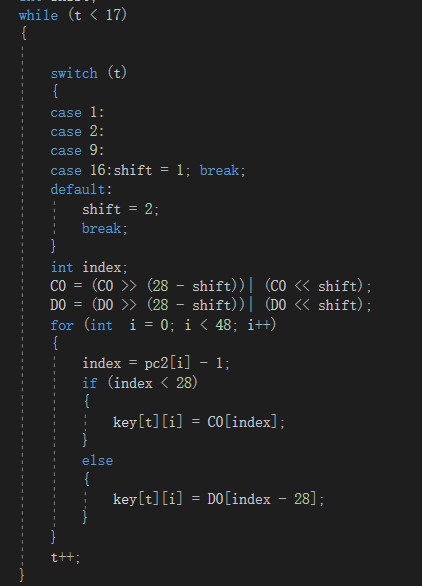


**生成子密钥：**

**Pc1置换：**



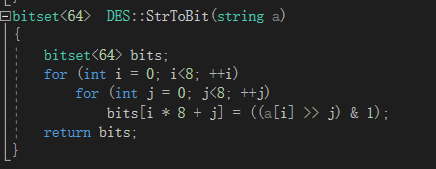
**Pc2置换：**



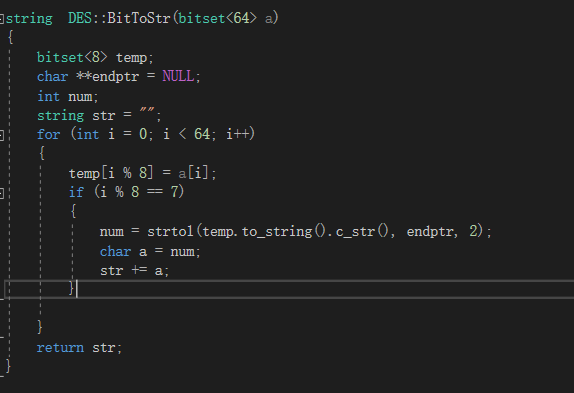
**数据结构：**

本程序使用了bitset数据类型，大多为对单个bit进行操作，实现了两个辅助函数：

将输入的信息转为二进制串



将二进制串转为字符串，用于对输出人类友好的信息：



**运行结果：**

