**DES算法的程序设计和实现——实验报告**

1. **算法原理概述**

DES 是一种典型的块加密方法。它以64位为分组长度，64位一组的明文作为算法的输入，通过一系列复杂的操作，输出同样64位长度的密文。DES 使用加密密钥定义变换过程，因此算法认为只有持有加密所用的密钥的用户才能解密密文。DES 采用64位密钥，但由于每8位中的最后1位用于奇偶校验，实际有效密钥长度为56位。密钥可以是任意的56位的数，且可随时改变。DES 算法的基本过程是换位和置换。

加密过程：

 C = Ek(M) = IP-1 · W · T16 · T15 ·… · T1 · IP(M).

解密过程：

 M = Dk(C) = IP-1 · W · T1 · T2 · … · T16 · IP (C) .

变量说明：

①M 为算法输入的64位明文块；

②Ek 描述以 K 为密钥的加密函数，由连续的过程复合构成；

③IP 为64位初始置换；

④T1, T2 , …, T16 是一系列的迭代变换；

⑤W 为64位置换，将输入的高32位和低32位交换后输出；

⑥IP-1 是 IP 的逆置换；

⑦C 为算法输出的64位密文块

1. **算法总体结构**

DES 算法的总体结构为Feistel 结构。

输入64位

初始置换IP

16轮迭代T

交换置换W

逆置换IP-1

输出64位

密钥调度

1. **模块分解**

DES算法模块分为两个大模块，一个是子密钥产生，一个是加解密过程。

子密钥产生部分分为3个模块：

①**PC-1置换**：对K的56个非校验位实行置换 PC-1，得到 C0D0，其中C0和D0分别由PC-1置换后的前28位和后28位组成。

②**16次LS置换**：计算 Ci = LSi(Ci-1) 和 Di = LSi(Di-1)。当i=1, 2, 9, 16时，LSi(A)表示将二进制串A循环左移一个位置；否则循环左移两个位置。

③**PC-2压缩置换**：对 56位的CiDi实行 PC-2压缩置换，得到48位的Ki

加密和解密过程分为6个模块：

①**获取子密钥**：使用子密钥产生模块

②**原始明文消息按PKCS#5 (RFC 8018)规范进行字节填充**：原始明文消息最后的分组不够8个字节 (64位) 时，在末尾以字节填满，填入的字节取值相同，都是填充的字节数目；原始明文消息刚好分组完全时，在末尾填充8个字节 (即增加一个完整分组)，字节取值都是08。

③**初始置换IP**：给定64位明文块 M，通过一个固定的初始置换 IP 来重排 M 中的二进制位，得到二进制串 M0 = IP(M) = L0R0，这里L0和R0分别是M0的前32位和后32位。

④**16轮迭代T**：根据 L0R0 按下述规则进行16次迭代，即Li = Ri-1, Ri = Li-1⊕f(Ri-1, Ki), i = 1 .. 16。输入64位明文M时，子密钥按 (K1K2 … K16)的次序调度，是加密过程。输入64位密文C时子密钥按 (K16K15 … K1)的次序调度。

⑤**交换置换W**：16次迭代后得到 L16R16；左右交换输出R16L16。

⑥**逆置换IP-1**：对迭代 T 输出的二进制串 R16L16使用初始置换的逆置换IP-1得到密文C。

注：实际编程过程中为了编程方便，实际编写的函数模块可能与上述模块的分类有所出入，但总体结构和算法步骤与上述DES算法相同。

1. **数据结构**

本程序设计使用C++ STL中的string作为存储数据结构，利用一维和二维数组保存DES算法所需要的置换矩阵，用于进行明文和密文的比特位的置换。

**五、编译运行结果**

注：由于C语言中一个char为一个字节（8个bit）。因此为了测试方便，64位明文我使用8个char来表示，密钥同样使用8个char来表示，但密文使用64位二进制数表示。

每个测例中含有下面两个方面的测试（将代码宏定义DEBUG设为1可观察每一步的结果）：

1.DES加密测试

输入：明文字符串、密钥

输出：密文字符串

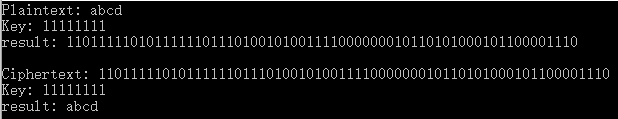
2.DES解密测试

输入：密文字符串、密钥

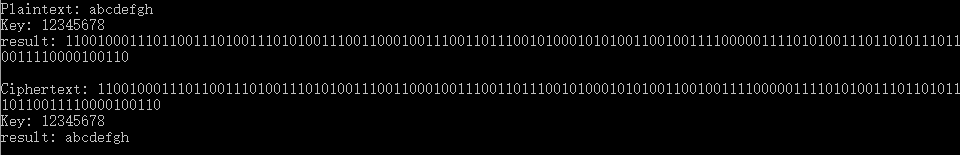
输出：明文字符串

测试顺序：将明文输入，得到密文；再将密文输入，得到明文。比较前后两段明文是否一致。

①明文不足8个字节



②明文为8个字节



③明文大于8个字节

