DES算法

　　美国国家标准局1973年开始研究除国防部外的其它部门的计算机系统的数据加密标准，于1973年5月15日和1974年8月27日先后两次向公众发出了征求加密算法的公告。加密算法要达到的目的（通常称为DES 密码算法要求）主要为以下四点： ☆提供高质量的数据保护，防止数据未经授权的泄露和未被察觉的修改；

☆具有相当高的复杂性，使得破译的开销超过可能获得的利益，同时又要便于理解和掌握；

☆DES密码体制的安全性应该不依赖于算法的保密，其安全性仅以加密密钥的保密为基础；

☆实现经济，运行有效，并且适用于多种完全不同的应用。

1977年1月，美国政府颁布：采纳IBM公司设计的方案作为非机密数据的正式数据加密标准（DES?Data Encryption Standard）。

　　目前在国内，随着三金工程尤其是金卡工程的启动，DES算法在POS、ATM、磁卡及智能卡（IC卡）、加油站、高速公路收费站等领域被广泛应用，以此来实现关键数据的保密，如信用卡持卡人的PIN的加密传输，IC卡与POS间的双向认证、金融交易数据包的MAC校验等，均用到DES算法。   
　　DES算法的入口参数有三个：Key、Data、Mode。其中Key为8个字节共64位，是DES算法的工作密钥；Data也为8个字节64位，是要被加密或被解密的数据；Mode为DES的工作方式，有两种：加密或解密。   
　　DES算法是这样工作的：如Mode为加密，则用Key 去把数据Data进行加密， 生成Data的密码形式（64位）作为DES的输出结果；如Mode为解密，则用Key去把密码形式的数据Data解密，还原为Data的明码形式（64位）作为DES的输出结果。在通信网络的两端，双方约定一致的Key，在通信的源点用Key对核心数据进行DES加密，然后以密码形式在公共通信网（如电话网）中传输到通信网络的终点，数据到达目的地后，用同样的Key对密码数据进行解密，便再现了明码形式的核心数据。这样，便保证了核心数据（如PIN、MAC等）在公共通信网中传输的安全性和可靠性。   
　　通过定期在通信网络的源端和目的端同时改用新的Key，便能更进一步提高数据的保密性，这正是现在金融交易网络的流行做法。   
　　DES算法详述   
　　DES算法把64位的明文输入块变为64位的密文输出块，它所使用的密钥也是64位，整个算法的主流程图如下：   
其功能是把输入的64位数据块按位重新组合，并把输出分为L0、R0两部分，每部分各长32位，其置换规则见下表：   
58,50,12,34,26,18,10,2,60,52,44,36,28,20,12,4,   
　　62,54,46,38,30,22,14,6,64,56,48,40,32,24,16,8,   
　　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,   
　　即将输入的第58位换到第一位，第50位换到第2位，...，依此类推，最后一位是原来的第7位。L0、R0则是换位输出后的两部分，L0是输出的左32位，R0 是右32位，例：设置换前的输入值为D1D2D3......D64，则经过初始置换后的结果为：L0=D58D50...D8；R0=D57D49...D7。   
　　经过16次迭代运算后。得到L16、R16，将此作为输入，进行逆置换，即得到密文输出。逆置换正好是初始置的逆运算，例如，第1位经过初始置换后，处于第40位，而通过逆置换，又将第40位换回到第1位，其逆置换规则如下表所示：   
　　40,8,48,16,56,24,64,32,39,7,47,15,55,23,63,31,   
　　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,   
放大换位表   
　　32, 1, 2, 3, 4, 5, 4, 5, 6, 7, 8, 9, 8, 9, 10,11,   
　　12,13,12,13,14,15,16,17,16,17,18,19,20,21,20,21,   
　　22,23,24,25,24,25,26,27,28,29,28,29,30,31,32, 1,   
单纯换位表   
　　16,7,20,21,29,12,28,17, 1,15,23,26, 5,18,31,10,   
　　2,8,24,14,32,27, 3, 9,19,13,30, 6,22,11, 4,25,   
　　在f(Ri,Ki)算法描述图中，S1,S2...S8为选择函数，其功能是把6bit数据变为4bit数据。下面给出选择函数Si(i=1,2......的功能表：   
选择函数Si   
S1:   
　　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,   
S2:   
　　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,   
S3:   
　　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,   
S4:   
　　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,   
S5:   
　　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,   
S6:   
　　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,   
S7:   
　　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,   
S8:   
　　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,   
在此以S1为例说明其功能，我们可以看到：在S1中，共有4行数据，命名为0，1、2、3行；每行有16列，命名为0、1、2、3，......，14、15列。   
　　现设输入为： D＝D1D2D3D4D5D6   
令：列＝D2D3D4D5   
　　行＝D1D6   
　　然后在S1表中查得对应的数，以4位二进制表示，此即为选择函数S1的输出。下面给出子密钥Ki(48bit)的生成算法   
　　从子密钥Ki的生成算法描述图中我们可以看到：初始Key值为64位，但DES算法规定，其中第8、16、......64位是奇偶校验位，不参与DES运算。故Key 实际可用位数便只有56位。即：经过缩小选择换位表1的变换后，Key 的位数由64 位变成了56位，此56位分为C0、D0两部分，各28位，然后分别进行第1次循环左移，得到C1、D1，将C1（28位）、D1（28位）合并得到56位，再经过缩小选择换位2，从而便得到了密钥K0（48位）。依此类推，便可得到K1、K2、......、K15，不过需要注意的是，16次循环左移对应的左移位数要依据下述规则进行：   
循环左移位数   
1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1   
　　以上介绍了DES算法的加密过程。DES算法的解密过程是一样的，区别仅仅在于第一次迭代时用子密钥K15，第二次K14、......，最后一次用K0，算法本身并没有任何变化。

二、DES算法理论图解

DES的算法是对称的，既可用于加密又可用于解密。下图是它的算法粗框图。其具体运算过程有如下七步。   
＜缺：找到补上＞

三、DES算法的应用误区

　　DES算法具有极高安全性，到目前为止，除了用穷举搜索法对DES算法进行攻击外，还没有发现更有效的办法。而56位长的密钥的穷举空间为256，这意味着如果一台计算机的速度是每一秒种检测一百万个密钥，则它搜索完全部密钥就需要将近2285年的时间，可见，这是难以实现的，当然，随着科学技术的发展，当出现超高速计算机后，我们可考虑把DES密钥的长度再增长一些，以此来达到更高的保密程度。   
　　由上述DES算法介绍我们可以看到：DES算法中只用到64位密钥中的其中56位，而第8、16、24、......64位8个位并未参与DES运算，这一点，向我们提出了一个应用上的要求，即DES的安全性是基于除了8，16，24，......64位外的其余56位的组合变化256才得以保证的。因此，在实际应用中，我们应避开使用第8，16，24，......64位作为有效数据位，而使用其它的56位作为有效数据位，才能保证DES算法安全可靠地发挥作用。如果不了解这一点，把密钥Key的8，16，24，..... .64位作为有效数据使用，将不能保证DES加密数据的安全性，对运用DES来达到保密作用的系统产生数据被破译的危险，这正是DES算法在应用上的误区，留下了被人攻击、被人破译的极大隐患。

http://www.iplaysoft.com/encrypt-arithmetic.html