**《密码学》课程设计实验报告**

实验序号：01　　　　　　　　　　实验项目名称：分组密码DES

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学　　号 |  | 姓　名 |  | 专业、班 | | 18信安 |
| 实验地点 | B310 | 指导教师 | 王后珍 | 时间 | | 2020.11.23 |
| 1. 实验目的及要求   教学目的：   1. 掌握分组密码的基本概念； 2. 掌握DES（3DES）密码算法； 3. 了解DES（3DES）密码的安全性； 4. 掌握分组密码常用工作模式及其特点； 5. 熟悉分组密码的应用。   实验要求：   1. 复习掌握（古典密码）使用的置换、代替、XOR、迭代等技术； 2. 比较DES中代替技术与古典密码中的联系与区别； 3. 理解S盒、P置换等部件的安全性准则； 4. 实现DES算法的编程与优化。   二、实验设备（环境）及要求  Windows操作系统，高级语言开发环境  三、实验内容与步骤  1. DES 子密钥扩展算法的实现  输入：64位密钥  子过程：  （1）置换选择1（教材 图3-3）  （2）循环左移（教材 表3-1）  （3）置换选择2（教材 图3-4）  输出：16个48位长的子密钥。    2．DES局部加密函数*f*的实现  加密函数是DES的核心部分。它的作用是在第*i*次加密迭代中用子密钥*K*i对*R*i-1进行  加密。  输入：32位*R*i-1和48位子密钥*K*i  子过程：   * 1. 扩展置换E（教材 图3-7）：将32位*R*i-1扩展为48位；   2. 异或操作：步骤（1）的48位结果与子密钥*K*i按位模2相加；   3. 代替S盒（教材 表3-2）：步骤（2）的48位结果分成6位×8组压缩为4位×8组，即32位输出；   4. 置换运算P（教材 图3-8）：32位输入/输出。   输出：32位*f* (*R*i-1*,K*i)    3. DES加密过程完整实现   1. 64位密钥经子密钥产生算法产生出16个子密钥：*K*1，*K*2，...，*K*16 ，分别供第一次，第二次，...，第十六次加密迭代使用。 2. 64位明文首先经过初始置换***IP***（Initial permutation），将数据打乱重新排列并分成左右两半。左边32位构成*L*0，左边32位构成*R*0。 3. 由加密函数*f*实现子密钥*K*1 对*R*0的加密，结果为32位的数据组*f*（*R*0，*K*1)。*f*（*R*0，*K*1)再与*L*0模2相加，又得到一个32位的数据组*L*0⊕*f*（*R*0，*K*1)。以*L*0⊕*f*（*R*0，*K*1)作为第二次加密迭代的*R*1，以*R*0作为第二次加密迭代的*L*1。至此，第一次加密迭代结束。 4. 第二次加密迭代至第十六次加密迭代分别用子密钥*K*2，...，*K*16进行，其过程与第一次加密迭代相同。 5. 第十六次加密迭代结束后，产生一个64位的数据组。以*R*16作为其左边32位，以*L*16作为其右边32位，两者合并再经过逆初始置换***IP*** –1，将数据重新排列，便得到64位密文。至此加密过程全部结束。   综上可将DES的加密过程用如下的数学公式描述：  *L*i = *R*i-1  *R*i=*L*i-1⊕*f* (*R*i-1*,K*i) （3-1）  i =1,2,3,…16  4. DES解密过程实现  由于DES的运算是对和运算，所以解密和加密可共用同一个运算，只是子密钥使用的  顺序不同。  把64位密文当作明文输入，而且第一次解密迭代使用子密钥*K*16，第二次解密迭代使用子密钥*K*15，…，第十六次解密迭代使用子密钥*K*1，最后的输出便是64位明文。  解密过程可用如下的数学公式描述：  *R*i-1= *L*i  *L*i-1= *R*i⊕*f* (*L*i*,K*i) （3-2）  i =16,15,14，...，1  5. DES的S盒密码学特性（**重点**）  通过编程实现或者手工计算，试验证***S***盒的以下准则：  ① 输出不是输入的线性和仿射函数；  ② 任意改变输入中的一位，输出至少有两位发生变化；  ③ 对于任何***S***盒和任何输入*x*，***S***(*x*)和***S***(x⊕001100)至少有两位不同，这里*x*是一个6位的二进制串；  ④ 对于任何***S***盒和任何输入*x*，以及*y*,*z*∈GF(2)，***S***(*x*)≠***S***(*x*⊕11*yz*00)，这里*x*是一个6位的二进制串；  ⑤ 保持输入中的1位不变，其余5位变化，输出中的0和1的个数接近相等。  例如，可通过如下步骤验证②、③两条：  设S盒的输入为X，输出为Y。（X和Y都以二进制表示）  （1）对于已知输入值X1=110010和X2=100010，分别求出对应的输出值Y1和Y2。  （2）比较输出值Y1和Y2各位的异同，即按位计算Y1⊕Y2。  根据上面得出的结果试说明S盒对于DES的安全性影响。  6.扩展思考  （1）Feistel结构为什么可以保证算法的对合性？  （2）第16轮为什么不做左右互换？  （3）如果去掉初始置换和逆初始置换，对算法安全性有影响吗？（提示：算法所有的细节都是公开的）  （4）证明DES解密算法是加密算法的逆，即DES的对合性。  （5）**a.**设X′是对X按位取反的结果。证明如果明文和密钥都取反，则密文取反。即   |  |  | | --- | --- | | 如果 Y ＝ | E(K, X) | | 那么Y′＝ | E(K′，X′) |   提示：首先证明对任意两个相同长度的串A和B，有  (A⊕B)′＝A′⊕B。  **b.**假设对DES的穷举攻击需要搜索256个密钥的密钥空间。a中的结论对此是否有影响？  （6）证明DES中每个子密钥的前24位均来自于初始密钥的同一个子集，该子集有28位，而后24位来自于初始秘密钥的另外28位。  四、实验结果与数据处理  ***代码整体说明***  相关文件(des.cpp/des.hpp)  接口设计如下图(其中用**蓝色方框标出的为公开方法**)    接口设计图  程序执行可以用如下的流程图来表示    DES流程图  ***核心代码说明***  首先是一些**置换表**的定义    置换表定义  由于**置换操作**在DES中经常被使用，并且置换表的长度是不定长的，在这里我们使用**模板方法**简化代码，其定义如下。在这里我们需要注意的是模板类***bitset***中数据的存储方式是颠倒的    Permute定义    另一个比较重要的运算是**S盒**，它是DES中的**唯一非线性部件**，也是为其提供安全性的基础。它一共由8个S盒组成，每个S盒有6为输入，产生4为输出，其运算规则可以用如下的公式来表示  与之相关的代码如下图所示    S盒代码  有了上述**S盒**以及**置换**这两个基本操作，我们可以很容易的进行**子密钥的生成**。其流程图如下所示    生成子密钥流程图  其对应的代码如下图    生成子密钥代码  得到了子密钥之后，我们可以开始加解密了。  其中**加密**函数如下图所示，它接受64比特的明文输入，并产生64比特的密文输出    加密函数  **解密**函数如下图所示，它接受64比特的密文输入，解密出64比特的明文    解密函数  在加解密的过程中，都调用了**轮加密函数**，它主要包括如下几个步骤   1. 选择运算E(32位输入，48位输出) 2. 轮密钥异或 3. S盒 4. 置换运算P     轮加密函数  ***实验结果演示***  这里我们直接采用书上的示例  其中产生的**子密钥**如下图所示    子密钥生成  **加解密**的结果如下图所示    实验结果图  可以发现我们的结果与书中的示例完全一致  五、分析与讨论   1. 由于DES算法需要对比特流进行操作，而现代计算机的存储单元是以字节为单位，因此要寻找一个合适的数据结构以比特单位存储信息。通过与同学的讨论，我了解到了C++中的模板类***bitset***能够完成这样的操作。 2. DES的S盒是唯一的非线性部件，因此它在保证DES的安全性上有至关重要的作用 3. DES的初始IP置换和IP逆置换由于没有用到加密密钥，因此在置换表公开之后它们在密码意义上作用不大 4. 尽管DES由于长度较短而存在被暴力枚举破解的风险，但是其设计思想仍然为后续的密码算法所借鉴 | | | | | | |
| 六、教师评语  签名：  日期： | | | | | 成绩 | |