**《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7班 | | 21信安 |
| 实验地点 | C202 | 指导教师 | 余荣威 | 时间 | | 2023 .12.1 |
| 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. 验证教材P64页实例（**重点**）  7.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。  四、实验结果与数据处理  1. DES 子密钥扩展算法的实现  （1）变换矩阵：  置换选择1    对应代码PC\_1    置换选择2    对应代码PC\_2    循环左移位数：    对应代码SHIFT    （2）变换操作  PC\_1置换：    PC\_2置换：    左移变换：    （3）子密钥的产生过程：    对应代码generate\_key    2．DES局部加密函数f的实现  （1）变换矩阵  选择运算E，采用重用某些数据位将32位输入扩展为48位    对应代码E    S盒  S盒共8个，对应教材表3-2，转化成矩阵形式即可，由于数量过多，不在这里展示了，具体的操作详见下面的（2）变换操作  置换运算P：将S盒的输出数据打乱重排，将混淆作用扩散开来    对应代码P    （2）变换操作  选择运算E    异或操作  将32位输入经过上一步选择运算E扩展后得到的48位中间结果与48位子密钥进行异或    对应代码xor    S盒变换：  S盒有6位输入，4位输出，代替规则为：    对应代码s    由于共8个S盒，我们要执行8次上面的s盒变换，我们将其整合为一个函数S\_change，将输入分成8分，并将输出整合    置换运算P    （3）加密函数f  执行流程图如下：    各个部分的实现均在(1)(2)中，我们直接在加密函数的过程中调用上述操作，对应代码F    3. DES加密过程完整实现  （1）置换矩阵  初始置换IP可以将64位明文打乱重排    对应代码IP    逆初始置换是初始置换的逆置换    对应代码IP\_RE    （2）变换操作：  初始置换IP    逆初始置换    （3）DES加密过程  DES的流程框架如下    使用到的函数均在1.2和本小节中实现，对应的代码如下    4. DES解密过程实现  DES加密算法是对合运算，因此加解密公用一个算法，解密是加密的逆过程，且子密钥使用的顺序也要逆置，对应的代码如下    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；    输出结果如下：    可以看到，对于输入，我们仅改变了最高位，输出却有3位变化    S盒的非线性性主要体现在改变输入的一个比特位时，输出中多个比特位发生变化，而且这些变化不具有规律性。在你的输出结果中，每次翻转输入的一个比特位，输出的Hamming距离（即输出中不同的比特位数）都发生了变化。这表明 S 盒的输出对于输入的变化是不可预测的，而不是线性的或者仿射的。  ② 任意改变输入中的一位，输出至少有两位发生变化；  由①中结果可得:    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；  我改动了上述测试代码，在这里随机生成一个长度为6的二进制比特串作为input1,并且使用001100作为掩码，将input1与掩码异或的结果作为input2    测试进行了三次，得到的结果如下：        从上述结果可以得到对于任何S盒和任何输入x，S(x)和S(x⊕001100)至少有两位不同  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；  修改test程序如下：    得到结果如下：    更换一个较为不一样的input1=100101    由上述结果可以得到对于任何S盒和任何输入x，以及y,z∈GF(2)，S(x)≠S(x⊕11yz00)  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。  修改test如下    得到结果    可以证明：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。  根据上面得出的结果试说明S盒对于DES的安全性影响。  S盒是DES中唯一一种非线性变换，其输入任意改变1位，输出至少变化2位，并且进行16次迭代，从而有如果改变明文的1位，密文都会大致发生32位的变化，大大提高了保密性。  6. 验证教材P64页实例（**重点**）  1.密钥扩展：  在密钥扩展的代码中添加打印指令如下：    得到结果    与3.1.8（书上64页）的示例相对比是一致的（出于篇幅限制就不全部截图贴上来了）    2.加密过程  对加密过程添加必要的打印指令得到的结果如下：              可以看到，我们输出的结果与书中P64页示例一致，说明加密过程有效  3.解密过程：  与加密过程类似，添加必要的打印指令              可以看到，我们输出的结果与书中P64页示例一致，说明解密过程有效  7.扩展思考  （1）Feistel结构为什么可以保证算法的对合性？  Feistel 密码结构是用于分组密码中的一种对称结构，其加解密流程如下图所示：    其中，加密的过程如下所示：    解密的过程如下所示：    猜测，Feistel结构保证算法的对合性的原因：  ①Feistel是一种对称结构  ②在加解密中，将明/密文分为长度相等的两块，进行的操作（如置换、异或）也是可逆的  ③加解密的算法是可逆的  对合性、可逆性的证明详见（4）  （2）第16轮为什么不做左右互换？  最后一轮不做互换可以保证交换次数是偶数，保持整个Feistel结构的可逆性。且最后一轮没有对右半部分进行变换的操作，如果在最后一轮中对左右半部分进行互换，那么整个结构将不再是可逆的，因为无法逆向恢复原始数据。  （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）  有影响。  初始置换和逆初始置换在算法的开始和结束阶段对数据进行了乱序操作。这有助于将输入数据的统计特征分散到整个加密过程中，降低了密文与明文之间的关联性，提高了算法的安全性。  （4）证明DES解密算法是加密算法的逆，即DES的对合性。    五、分析与讨论  1.DES的优点和缺点是什么？  优点：  （1）安全性（在过去）：DES曾经是一个安全可靠的加密算法，经过广泛测试和密码学分析，并在实际应用中得到验证。然而，随着计算能力的增强和密码分析技术的进步，DES的安全性逐渐弱化。  （2）速度较快：DES是一种具有较高加密和解密速度的算法，对于大多数应用而言，其执行速度是可接受的。  （3）算法简单：DES的算法结构相对简单，易于实现和部署。这使得DES在早期广泛应用于各种系统和设备。  缺点：  （1）安全性较弱：DES的密钥长度为64位（有效位为56位），相对较短。随着计算能力的提升和密码分析技术的发展，DES的密钥空间已经变得较小，容易受到穷举搜索和其他攻击方式的威胁。  （2）密钥管理复杂：由于DES密钥长度较短，密钥管理变得复杂。在大规模系统中，安全地分发和管理DES密钥是一个挑战。  （3）不支持现代的安全要求：DES的密钥长度和算法设计已经不足以满足现代密码学的安全要求。DES已经被更安全的加密算法，如AES等所取代。  （4）不支持数据完整性和身份验证：DES仅提供加密和解密功能，没有内置的数据完整性验证和身份验证机制。在某些应用场景下，这可能需要额外的安全措施。  2.如何提升DES算法性能？  （1）使用硬件加速：DES算法可以在专门的硬件加速器中实现，这样可以显著提高加密和解密的速度。硬件加速器专门针对DES算法进行优化，提供高效的硬件并行处理能力。  （2）使用分组模式：DES算法通常以分组模式（如ECB、CBC、CTR等）的形式应用在实际系统中。选择合适的分组模式可以在不降低安全性的前提下提高算法的性能。例如，CTR模式可以充分利用并行处理，提高加密和解密的速度。  （3）密钥预计算和缓存：针对多次加密和解密操作，可以预先计算子密钥并进行缓存，避免重复计算，提高算法的性能。这样可以降低运行时的计算开销。  （4）并行计算：DES算法可以通过并行计算的方式提高性能。可以将数据分成多个块，并使用多个处理单元同时进行加密或解密操作。这在多核处理器或分布式系统中特别有效。  5. 使用更高效的替代算法：由于DES的安全性已经较弱，可以考虑使用更高效和安全的替代算法，如AES（Advanced Encryption Standard）。AES算法具有更大的密钥空间和更高的安全性，同时也具备较高的性能。 | | | | | | |
| 六、教师评语  签名：  日期： | | | | | 成绩 | |