**《密码学》课程设计实验报告**

实验序号：09　　　　　　　　　　实验项目名称：DES四种分组模式文件加密

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学　　号 | 2019302050244 | 姓　　名 | 庞远心 | 专业、班 | | 19信安2班 |
| 实验地点 | 网安基地新珈楼B308 | 指导教师 | 王张宜 | 时间 | | 2021年12月 |
| 1. 实验目的及要求   教学目的：   1. 熟悉适合密码应用的领域及应用方法； 2. 掌握一种密码应用技术。   实验要求：   1. 掌握对称算法的实现； 2. 掌握非对称算法的实现； 3. 实现下列密码学的一种（或多种）应用：  * 计算机文件加密 * 通信加密 * 认证 * 签名 * 密钥管理  1. 掌握加密、签名、认证、密钥管理等技术的综合运用。   二、实验设备（环境）及要求  Python 3.7  Windows 10  三、实验内容与步骤  1. 实验内容   1. 四种分块加密方式实现DES对文本加密解密并验证 2. DES：   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。  DES的结构为Feistel结构，结构如下：  图1： DES加密算法的一般描述  算法框架可分为轮秘钥产生和16轮加密(解密)，对于原理的阐述此处略。   1. 电码本模式（ECB）   电码本模式是直接利用分组密码对明文的各分组进行加密的方式。  明文,密钥为K，密文C＝ ( C1 ，C2 ，…，Cn ),其中 Ci ＝E（Mi，K）,i=1,2,…,n  图2：ECB加密解密模式原理示意图  ECB（电子密本方式）是将数据按照8个字节一段进行DES加密或解密得到一段8个字节的密文或者明文，最后一段不足8个字节，按照需求补足8个字节进行计算，之后按照顺序将计算所得的数据连在一起即可，各段数据之间互不影响。   * 特点：  1. 简单，有利于并行计算，误差不会被传送； 2. 不能隐藏明文的模式，在密文中出现明文消息的重复 3. 可能对明文进行主动攻击，加密消息块相互独立成为被攻击的弱点 4. 密文链接模式（CBC）   明密文链接方式具有加解密错误传播无界的特性，而磁盘文件加密和通信加密通常希望解密错误传播有界，这时可采用密文链接方式。  **设明文**,密钥为K，**密文**C=(C1 ，C2 ，…，Cn ),  **其中**       * 原理   加密步骤如下：   * 1. 首先将数据按照8个字节一组进行分组得到D1D2......Dn（若数据不是8的整数倍，用指定的PADDING数据补位）   2. 第一组数据D1与初始化向量I异或后的结果进行DES加密得到第一组密文C1（初始化向量I为全零）   3. 第二组数据D2与第一组的加密结果C1异或以后的结果进行DES加密，得到第二组密文C2   4. 之后的数据以此类推，得到Cn   5. 按顺序连为C1C2C3......Cn即为加密结果。   解密是加密的逆过程，步骤如下：   * 1. 首先将数据按照8个字节一组进行分组得到C1C2C3......Cn   2. 将第一组数据进行解密后与初始化向量I进行异或得到第一组明文D1（注意：一定是先解密再异或）   3. 将第二组数据C2进行解密后与第一组密文数据进行异或得到第二组数据D2   4. 之后依此类推，得到Dn   5. 按顺序连为D1D2D3......Dn即为解密结果。   这里注意一点，解密的结果并不一定是我们原来的加密数据，可能还含有你补得位，一定要把补位去掉才是你的原来的数据。   * 特点：  1. 不容易主动攻击,安全性好于ECB,适合传输长度长的报文,是SSL、IPSec的标准。每个密文块依赖于所有的信息块，明文消息中一个改变会影响所有密文块 2. 发送方和接收方都需要知道初始化向量 3. 加密过程是串行的，无法被并行化(在解密时，从两个邻接的密文块中即可得到一个平文块。因此，解密过程可以被并行化)。   图3：CBC加密解密模式原理示意图   1. 密文反馈模式（CFB）   密文反馈模式类似于CBC，可以将块密码变为自同步的流密码；工作过程亦非常相似，CFB的解密过程几乎就是颠倒的CBC的加密过程。   * 原理   需要使用一个与块的大小相同的移位寄存器，并用IV将寄存器初始化。然后，将寄存器内容使用块密码加密，然后将结果的最高x位与平文的x进行异或，以产生密文的x位。下一步将生成的x位密文移入寄存器中，并对下面的x位平文重复这一过程。解密过程与加密过程相似，以IV开始，对寄存器加密，将结果的高x与密文异或，产生x位平文，再将密文的下面x位移入寄存器。 与CBC相似，平文的改变会影响接下来所有的密文，因此加密过程不能并行化；而同样的，与CBC类似，解密过程是可以并行化的。  图4：CFB加密解密模式原理示意图   1. 输出反馈模式（OFB）   输出反馈模式（Output feedback, OFB）可以将块密码变成同步的流密码。它产生密钥流的块，然后将其与平文块进行异或，得到密文。与其它流密码一样，密文中一个位的翻转会使平文中同样位置的位也产生翻转。这种特性使得许多错误校正码，例如奇偶校验位，即使在加密前计算而在加密后进行校验也可以得出正确结果。每个使用OFB的输出块与其前面所有的输出块相关，因此不能并行化处理。然而，由于平文和密文只在最终的异或过程中使用，因此可以事先对IV进行加密，最后并行的将平文或密文进行并行的异或处理。 可以利用输入全0的CBC模式产生OFB模式的密钥流。这种方法十分实用，因为可以利用快速的CBC硬件实现来加速OFB模式的加密过程。  图5：OFB加密解密模式原理示意图   1. 四种分块加密方式实现DES对文件加密解密并测速   在实现了（1）中的功能之后笔者开始着手于将加密解密的文本换成大的文件，这个步骤采用随机生成一个大的文本文件，该文件的字节数至少为8\*1024个，保证足够大，可以测速。  2. 实验步骤   1. 实验主要文件功能说明     图6：实验主要文件说明   1. 实验主要函数流程     图7：主要函数调用图   1. test1为加密验证实验，从des\_plain.txt中读取32位表示16进制的字符串作为明文，将实验结果写入des\_cipher.txt中 2. test2为解密验证实验，从des\_cipher.txt中分别读取不同模式下的32位表示16进制的字符串作为密文，将实验结果写入plain.txt中 3. test3为不同模式下的加密速度实验，从random\_data.txt中读取随机生成的表示16进制的字符串作为明文，将实验结果写入test3\_result.txt中 4. test4为不同模式下的解密速度实验，从random\_data.txt中读取随机生成的表示16进制的字符串作为密文，将实验结果写入test4\_result.txt中 5. 注意：本实验所有输入的数据的类型是表示16进制的字符串类型，如：23684267384FEAB90。如果输入的是其他形式的数据，请先将文件转换为二进制类型再输入，通过operations.py中的bin2string()函数转换为16进制的字符串类型。 6. 主要功能实现    1. 页面  * **页面实现**   **页面实现采用python提供的tkinter库，** tkinter 模块(tk接口)是 Python 的标准 Tk GUI 工具包的接口 .Tk 和 Tkinter 可以在大多数的 Unix 平台下使用,同样可以应用在 Windows 和 Macintosh 系统里。Tk8.0 的后续版本可以实现本地窗口风格,并良好地运行在绝大多数平台中。   * 代码     图8：页面实现代码   * 页面效果     图9：页面效果  注：在笔者的电脑上的字体格式是“猫啃网文明宋-H”，可能在老师的电脑上不是那么明显。   * 使用方法     图10：页面使用流程  在运行后等待大概10分钟，在运行代码的目录下查看结果文档的内容   * 1. 调用函数action（） * 功能   在点击运行按钮后，python将读取输入的两个文件名并进行文本的加解密，   * 代码     图11：action（）函数代码  其中des\_vi.txt存储的是CBC\CFB\OFB的初始iv向量   * 1. Test1（）和test2（）   这一步是根据operations.py中已经定义好的操作（S盒、生成轮函数那些），由于篇幅过长且之前实现过，不在这里将每个操作的定义和功能再写一遍，笔者解释一下如何实现四种分组模式。  首先定义两个类，分别是DES加密器和解密器类，每个类包含一些初始化操作和四种分组加解密的方法。以下以加密器类的实现为例：   * 初始化      * Cipher定义     这一步就是我们之前实现的DES的结构，包括生成秘钥、初始的置换、S盒的实现等。四种加密方式实际的区别在于**秘钥的生成和加密方式**，所以在调用cipher的地方修改即可。代码分别如下：   * ECB      * CBC      * CFB      * OFB     根据实验效果可得加密解密过程正确。   * 1. Test3（）和test4（）   Test3（）和test4（）的实现基于test1（）和test2（）改进，区别在于数据量更大，所以需要在最初的时候随机生成一个大文件，生成大文件的代码如下：  **注：此时test3对random\_data.txt中的内容进行加密，test4也对新生成random\_data.txt解密，所以解密内容和加密内容部内容不一致，只是为了测速而已每种方式加解密各迭代10次，得到平均速度。**    四、实验结果与数据处理  1.  des\_plain.txt:    des\_key.txt:  des\_vi.txt:  2. 结果  des\_cipher.txt  Plain.txt:  Plain.txt已经验证了该des算法的正确性，则只需要测速  Random\_data.txt:  Test3\_result.txt:  Test4\_result.txt:  五、分析与讨论  1.分析   * 安全性   本次实验实现了四种分组加密的模式，可根据实验结果中看到ECB的加解密最快，其次是CBC,CFB,最慢的是OFB，但根据前面的分析可得CFB、OFB的安全性更高，可以用来验证数据的完整性和保密性等，ECB可用来加密解密短的秘钥。   * 算法可拓展性   在加密解密器类可以再添加XCBC等分组模式。   * 程序文件大小 * 执行速度统计和数据吞吐率统计功能   已在实验结果中列出 | | | | | | |
| 六、教师评语  签名：  日期： | | | | | 成绩 | |