**《密码学》课程设计实验报告**

实验序号：04　　　　　　　　　　实验项目名称：序列密码

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学　　号 |  | 姓　　名 |  | 专业、班 | | 18信安 |
| 实验地点 | 新珈楼B310实验室 | 指导教师 | 王后珍 | 时间 | | 2020.12.14 |
| 1. 实验目的及要求   教学目的：   1. 掌握序列密码的基本概念； 2. 掌握线性移位寄存器的结构及其序列的伪随机性； 3. 熟悉非线性序列的概念与基本产生方法； 4. 了解常用伪随机性评价方法； 5. 掌握一种典型流密码（如RC4或ZUC等）。   实验要求：   1. 掌握序列密码的实现方案； 2. 掌握线性移位寄存器的构造； 3. 熟悉序列伪随机性的基本测试方法； 4. 实现RC4或ZUC算法。   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. 序列密码的实现方案  （1）种子密钥K输入到密钥流发生器；  （2）产生一系列密钥流；  （3）通过与同一时刻的一个字节或者一位明文流进行异或操作产生密文流。  2．线性移位寄存器的构造  （1）选择连接多项式（一般选择本原多项式作为连接多项式）  （2）根据连接多项式的反馈系数得出反馈函数  （3）根据反馈函数得出每个节拍的寄存器状态  （必做题：实验1-2）  **实验（1）**使用本原多项式g1(x)=x4 +x +1为连接多项式组成线性移位寄存器。画出逻辑图，写出输出序列及状态变迁。  **实验（2）**使用本原多项式g2(x)= x4 +x3 +1为连接多项式组成线性移位寄存器。画出逻辑图，写出输出序列及状态变迁。  试对比以上两组输出序列的关系。  提示：（1）与（2）中的多项式是互反多项式，所谓互反多项式是指f(x)与  3. 随机性测试（选作）  随机性假设测试准则  **准则1（频率测试）**：在S的周期SN中，1的个数与0的个数至多相差1。  **准则2（游程测试）**：在S的周期SN中，至少有1/2的游程长度为1，至少有1/4的游程长度为2，至少有1/8的游程长度为3，以此类推，并且0和1游程的个数近似相等。  **准则3（自相关测试）**：自相关函数R(t)是双值的。即对某个整数K，有：  对于实验（1）-（2）产生的随机序列进行上述测试。  4. 编程实现ZUC算法。  5. 思考与扩展阅读（选作）  （1）定理：GF（2）上的 n级移位寄存器有2n个状态，有种不同的反馈函数；  （2）定理：GF（2）上的 n级移位寄存器中线性反馈函数只有 2n-1种；  （3）定理：n次本原多项式的个数是，其中是欧拉函数；（欧拉函数：小于或等于x的数中与x互质的数的数目）  例如对于n=4，，因此对于4元的线性移位寄存器，只有2个LFSR可以生成m序列，即实验（1）和（2）  （4）在实际的序列密码设计中，人们往往选择项数较少的本原多项式构造LFSR，例如三项式或五项式。试分析选择的原因。  （5）本原多项式一定是不可约多项式；反之则不一定成立。  例如GF（2）上的1-4元不可约多项式有（黄色标注的是本原多项式，互反多项式略去）：  n=1时，x、x+1  n=2时，x2+x+1  n=3时，x3+x+1  n=4时，x4+x+1、x4 +x3 +x2 +x +1（T=5）  为了书写简便，5-8元不可约多项式写成系数的形式，如111代表x2+x+1  n=5时，100101、101111、110111  n=6时，1000011、1001001（T=9）、1010111（T=21）、1011011、1100111  n=7时，10000011、10001001、10001111、10011101、10100111、10101011、10111111、11001011、11101111  n=8时，100011011（T=51）、100011101、100101011、100101101。。。。。。  （6）常用的判定GF（2）上的不可约多项式的方法：  （a）如果f(x)的常数项为0，除f(x)=x之外，f(x)一定可约；  （b）如果f(x)的项数为偶数，除f(x)=x+1之外，f(x)一定可约；  （c）如果f(x)中各项的x的幂次都是2的倍数，f(x)一定可约；  （d）如果gcd（f(x), f’(x)）≠1，f(x)一定可约；  （e）如果f(x+1)可约，f(x)一定可约；  （f）如果可约，f(x)一定可约；等等。。。。。。  （7）密码设计的初步尝试：  （a）设计AES类的S盒  选择不同的不可约多项式，按照AES S盒的相同计算过程即可  （b）生成m序列  选择不同的本原多项式，按照LFSR的构造过程即可。  四、实验结果与数据处理  ***代码整体说明***  相关文件(zuc.cpp/zuc.hpp)  接口设计如下图，一共包含三个类，分别为   * ***ZUCPredefinedArrays***：预先定义的表 * ***LFSR***：线性移位寄存器类 * ***ZUC***：ZUC加解密类   其中每个类的接口定义分别如下(**蓝色框标出为公开方法**)    ***ZUCPredefinedArrays***类    ***LFSR***类    ***ZUC***类  程序的执行流程可以用下图来表示    ZUC流程图  ***核心代码说明***  在祖冲之密码算法中一个重要的数据结构就是**线性反馈移位寄存器**，它由16个31比特寄存器单元变量组成，并且  线性反馈移位寄存器以有限域上的16次本原多项式组成  线性反馈移位寄存器的工作模式主要有两种   * 初始化模式 * 工作模式   在初始化模式下，LFSR接受一个31比特字，由非线性函数的32比特输出通过舍弃最低比特位得到。  在工作模式下，LFSR不接受任何输入，依次产生输出。  其对应的代码如下图所示。    LFSR工作代码    注意到这里我们将工作模式和初始化模式合二为一了，并且使用了两个函数***AddUint31***和***RotlUint31***分别进行31位整数的加法以及循环右移  值得一提的是，这里我们使用了双端队列***deque***来保存***LFSR***的状态数组，使其在进行循环移位的时候只需要的时间  **比特重组**也是ZUC中的一个重要算法，其实现如下    比特重组  为了保证密码的可靠性，除了线性移位寄存器之外，我们还需要有非线性部件S盒，这点主要有ZUC中的F运算来保证。其代码如下    F运算  在F运算中***L1***和***L2***分别表示两个32比特线性变换，定义如下    L1和L2  接下来是ZUC的初始化部分，它主要包括如下几个步骤   * 密钥扩展 * LFSR加载扩展密钥 * 32轮初始化     ZUC初始化  在初始化完成后，我们就可以进行密钥流的生成，代码如下    密钥流生成  ***实验结果演示***  在这里我们直接使用书上的例子。  可以发现初始化阶段LFSR的初态如下图，与书上的结果相吻合    LRSR初态  初始化完成后LFSR的状态如下图所示    初始化后LFSR的状态    有限状态自动机的内部状态如下图所示    有限状态自动机    生成的密钥流如下    密钥流  均符合实验结果。  五、分析与讨论   1. 为了在通信的双方能够产生相同的密钥序列，因此我们通常需要伪随机序列，而不是真随机序列 2. ZUC算法通过线性移位寄存器产生随机序列。尽管它产生的随机序列是线性的，但是通过与S盒相结合的方式，我们依然能够获得较好的安全性 3. 当线性移位寄存器的连接多项式为本原多项式时，线性移位寄存器的输出序列周期能够达到最大值。因此我们通常需要为ZUC算法选择合适的连接多项式 | | | | | | |
| 六、教师评语  签名：  日期： | | | | | 成绩 | |