****

现代密码学

**实验报告**

**学生姓名 许可嘉**

**学 号** 0906140102

**专业班级** 信息安全1401班

**指导教师** 段桂华

**学 院** 信息科学与工程学院

**完成时间** 2016年4月

**实验一 密码算法实验**

**[实验目的]**

1.掌握密码学中经典的对称密码算法AES、RC4的算法原理。

2.掌握AES、RC4的算法流程和实现方法。

**[实验预备]**

1. AES算法的基本原理和特点。

基本原理：混乱和密钥扩散。AES加密数据块分组长度必须为128比特，密钥长度可以是128比特、192比特、256比特中的任意一个（如果数据块及密钥长度不足时，会补齐）。AES加密有很多轮的重复和变换。大致步骤如下：1、密钥扩展（KeyExpansion），2、初始轮（Initial Round），3、重复轮（Rounds），每一轮又包括：SubBytes、ShiftRows、MixColumns、AddRoundKey，4、最终轮（Final Round），最终轮没有MixColumns。

特点：a.克服DES算法的弱点，保留了多轮的合理模式

b.分组，密钥长度和轮数可变，支持长度为128位，192位和256位的分组和密钥

c. 安全,易实现,算法灵活与简便

2.流密码RC4的密钥流生成以及S盒初始化过程。

**密钥流生成：**

j=i=0;

i=(i+1) mod 256;

j=(j+Si) mod 256;

交换Si和Sj;

t=(Si+Sj)mod256;

k=St；

k为密钥流

**S盒初始化：**

a.S0=0,S1=1...S255=255;

b.用初始密钥key填充另一个256字节的数组不断重复密钥直至填充整个数组

j=0

对于i=0至255

j=(j+Si+Ki)mod256

交换Si和Sj

C.得到S盒

**[实验内容]**

1. 分析AES、RC4的实现过程。

2. 用程序设计语言将算法过程编程实现。

3. 完成字符串数据的加密运算和解密运算

输入十六进制明文：11223344556677889900AABBCCDDEEFF

输入十六进制密钥：13579BDF02468ACE1234567890ABCDEF

**[实验步骤]**

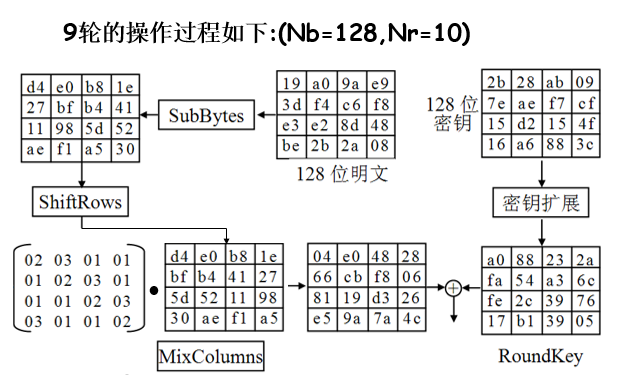
1. 预习AES、RC4算法。

AES: 比利时密码学家Joan Daemen和Vincent Rijmen提出的密码算法方案。算法分析如下：128比特的明文分组被分成4 X 4 的字节矩阵。1.字节替换SubBytes.非线性置换，独立作用于状态中的每一个字节；2.移位行运算ShiftRows.字节的循环移位运算，第1行和第4行字节分别向左移动0~3列；3.混合列运算MixColumns.由一个线性变换对状态的每一列进行变换；4.轮密钥加密AddRoundKey.与对应的密钥进行异或。

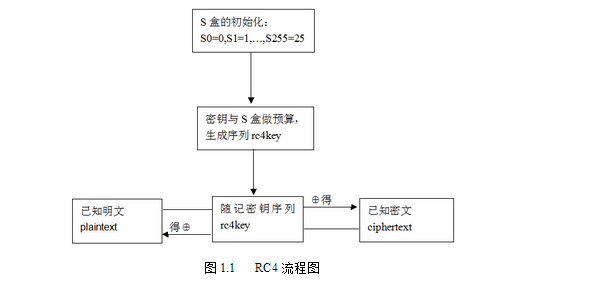
RC4: Ron Rivest在1987年为RSA数据安全公司开发的可变密钥长度的序列密码，广泛应用于商业密码产品中。RC4算法中密钥序列与明文相互独立；有一个256字节的S盒：S0，S1，...S255,其值是0到255之间不重复的值，是一个可变的长度密钥的函数。

1. 写出算法流程，用程序设计语言将算法过程编程实现。

AES算法流程：

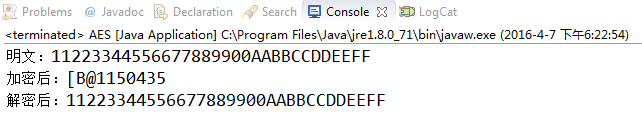


RC4算法流程：

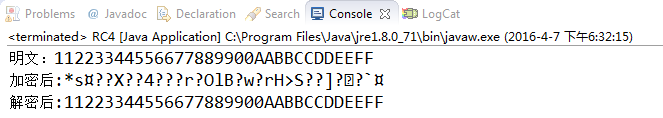


1. 输入指定的明文、密钥进行实验，验证结果。

AES算法结果：



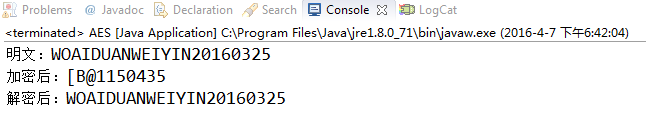
RC4算法结果：

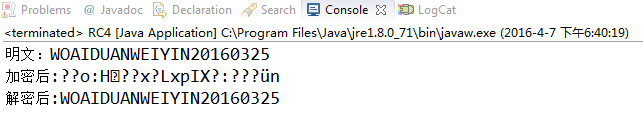


4. 自己选择不同的输入，记录输出结果。

**写出所编写程序的流程图和运行界面、运行结果。**

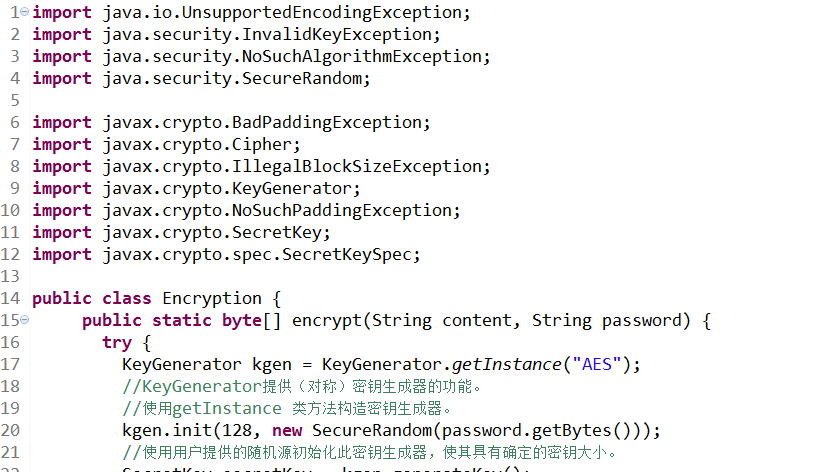
AES算法结果：

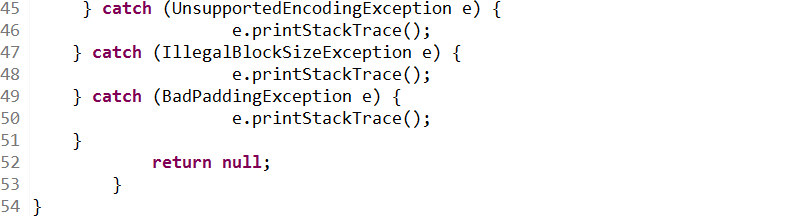
RC4算法结果：

源代码如下：(用java语言实现)

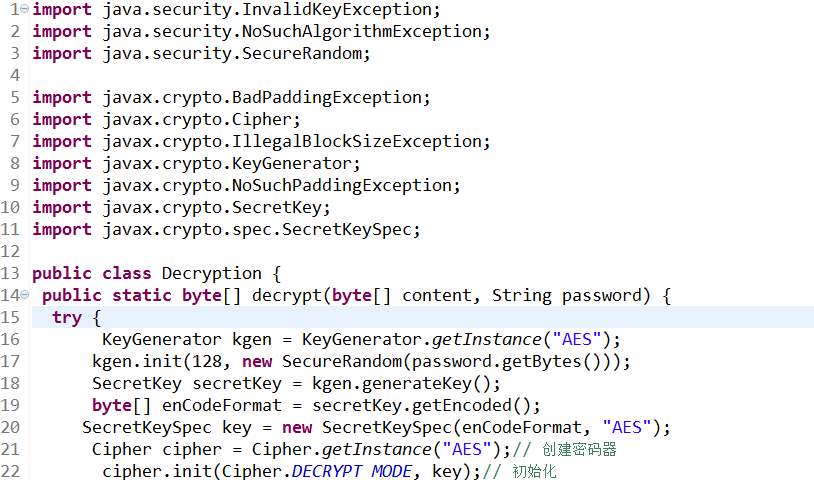
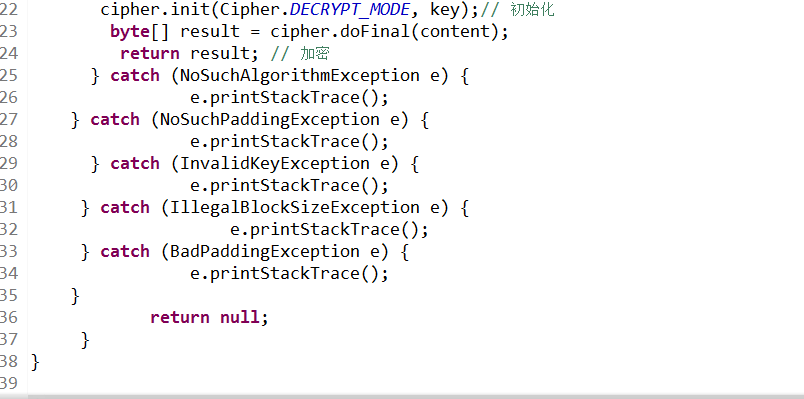
1.AES：

加密类：





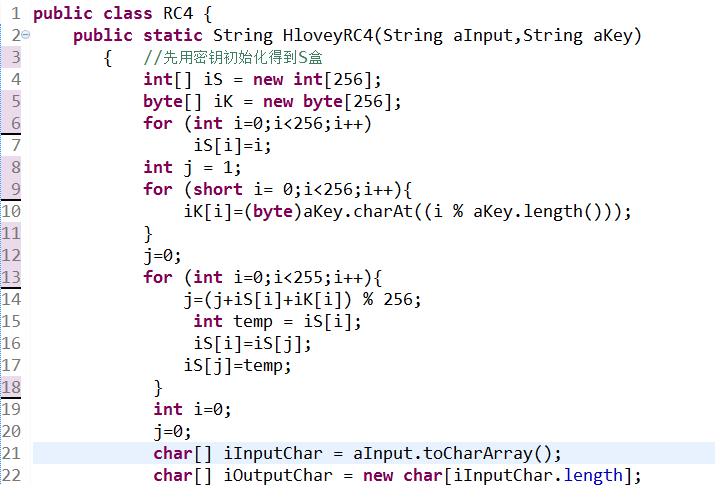
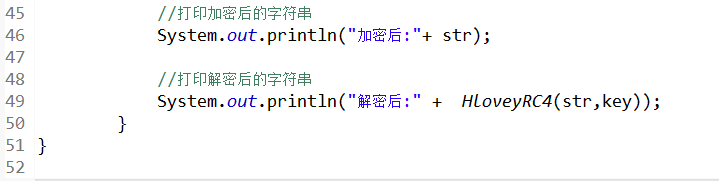
解密类：

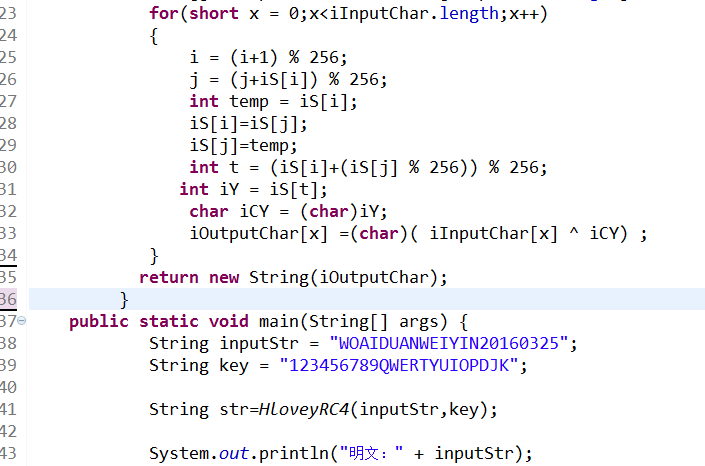
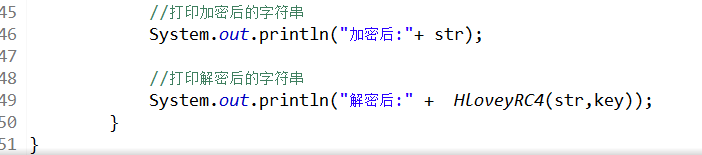


AES主函数：



1. RC4：





**[问题讨论]**

1. 改变明文或密钥中的一个比特值可能影响AES值中的多少比特？

可能影响80比特

1. 在RC4的密钥流生成中，改变初始密钥的一个比特值可能影响输出中的多少比特？

初始密码不同，S盒不同，因此可能影响256比特

3.分析实验中在编辑、编译、运行等各环节中所出现的问题及解决方法。

在AES算法实现过程中,只用一个类实现，代码太过杂乱条理不清，然后我用了java项目常用的思想，分开加密类和解密类和主函数，顿时使代码清晰，处理bug也方便多。Eclipse编写导入类时会遇到各种问题，通过百度解决。

RC4算法在对输入数据的处理上遇到了很多的问题，刚开始的时候习惯性地使用char的形式保存S盒等数据，但是由于AES算法用到了很多的异或和移位操作，最后选择了使用byte类型的数据。

在对这两个算法实现过程中加深对算法思想流程的理解，提高了对密码学学习的兴趣。

**实验二 SKEY协议设计实验**

**[实验目的]**

1.掌握身份认证协议的原理和基本思想。

2.掌握SKEY协议的机制和实现方法。

**[实验预备]**

1. SKEY协议的作用。

(1)由于每个数只用一次，因此对数据库攻击用处不大;

(2)节约储存开销

1. SKEY协议的安全性分析。

依赖于单向函数的安全性

3.SKEY协议的实现过程。

**[实验内容]**

1. 分析SKEY协议的实现过程。

2. 用程序设计语言将算法过程编程实现。

3. 演示SKEY协议的身份鉴别过程。

**[实验步骤]**

1. 预习SKEY协议的机制。

skey一次性口令系统是一个基于MD5的一次性口令生成方案。它可以对访问者的身份与设备进行综合验证。Skey协议的操作时基于[客户端](http://baike.baidu.com/view/930.htm" \t "_blank)/服务器端模式。

1. 选择和实现相应的摘要算法MD5或SHA。

MD5及其算法流程：

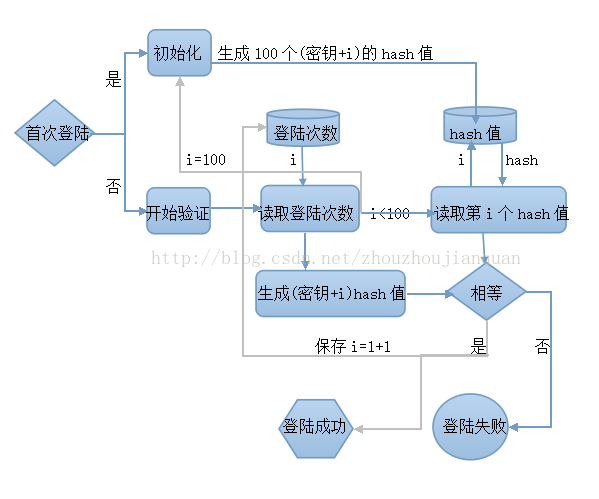
附加填充比特-->附加长度-->初始化MD缓冲区-->按每块16字（512比特）对数据进行处理-->输出

1. 写出算法流程，用程序设计语言将协议过程编程实现。

分析：

为了实现起来简单明了一些，我采用了一个比较简单的方式进行。要模拟客户端保存数据比较不容易实现，对服务器生成的hash值，保存在本地文件中,当然这个hash的值是动态的，每次hash都会加上相应的登陆次数（这个hash方式和一般的skey协议有些不同）,这样可以动态的获取。通过文件读写的方式获得。对于生成的hash值从100到0次访问的问题也是使用一个文件来储存，每次验证完成后，都会把前一次存的数据减1然后重新的写入文件中。

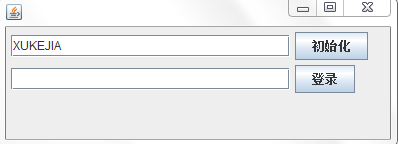
流程图：



具体代码见文档最下面

4. 验证SKEY协议的身份鉴别过程。

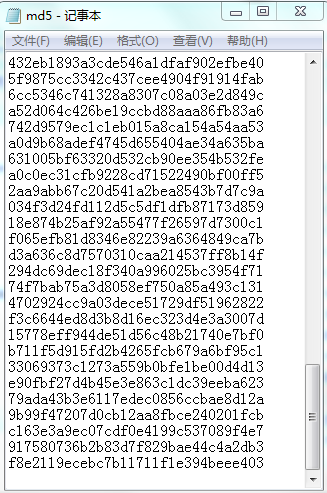
(1)输入用户名进行初始化



生成文件：

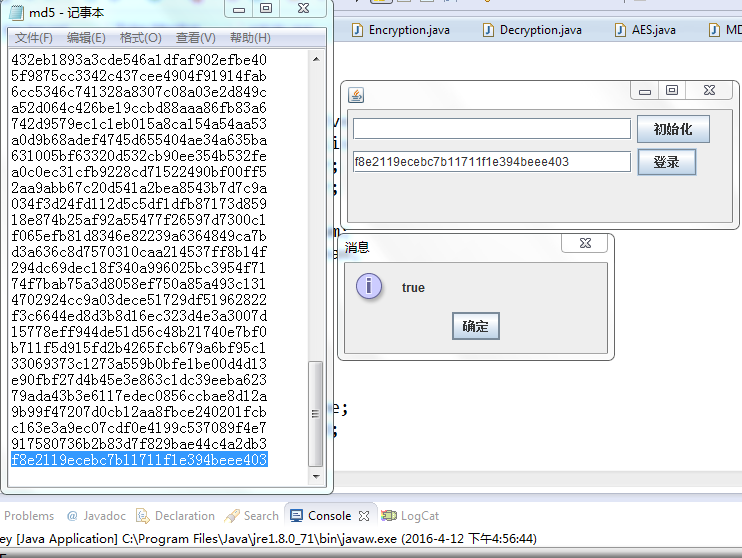


md5存放hash值：

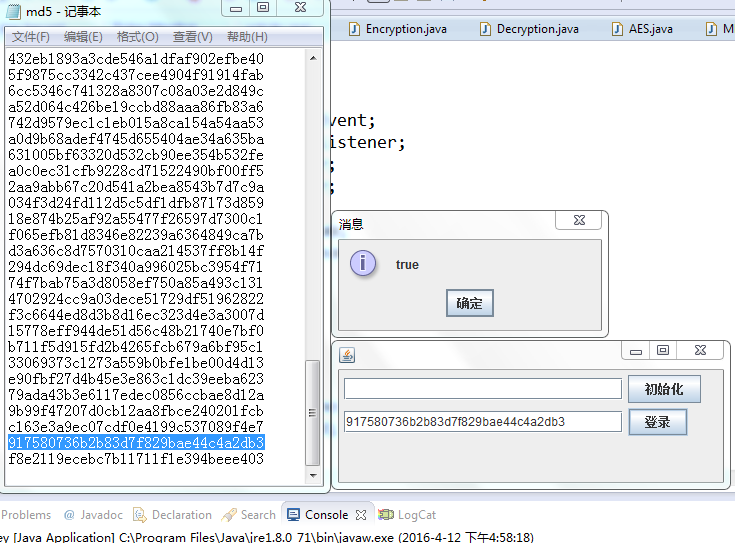


（2）用hash值（每次用一行）登陆验证：

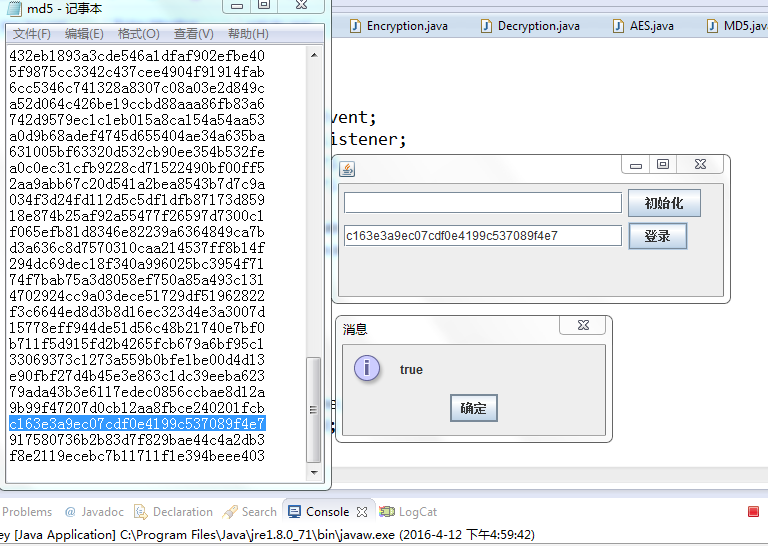
第一次：



第二次：



第三次：



**[问题讨论]**

1. 分析SKEY的安全性；

1.每个hash值只能使用一次，对数据库攻击用处不大

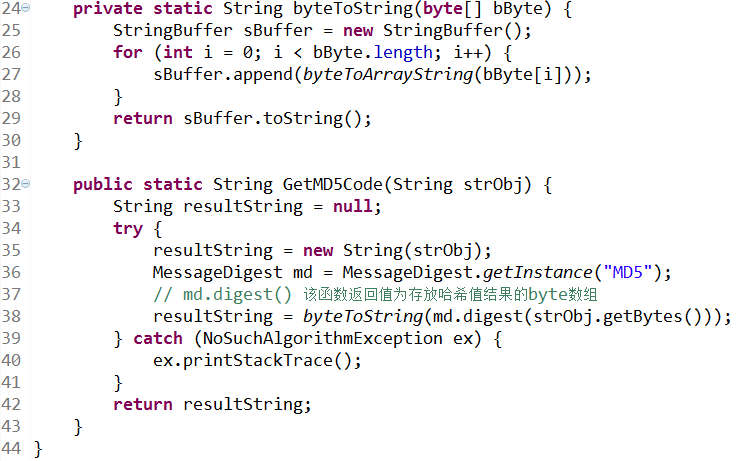
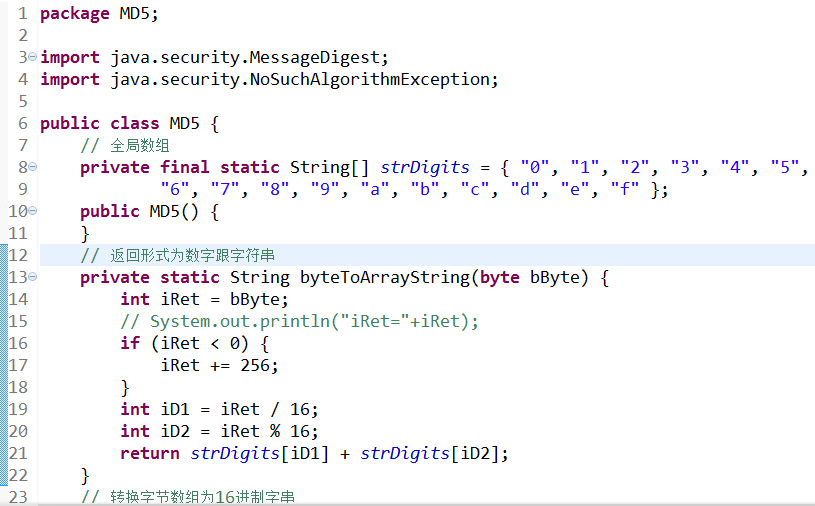
2.避免重放攻击

2. 分析实验中在编辑、编译、运行等各环节中所出现的问题及解决方法。

一开始想上传到服务器端，后来改为用两个文件操作，实现起来更便捷。对java的io流实现要很熟悉，在该算法实现过程中加深对算法思想流程的理解，提高了对密码学学习的兴趣。

代码如下：(用java实现)

MD5.JAVA



Skey.java

