密码学大作业解释文档

实现DES和RSA加密算法

姓名：包逸豪

学号：3119105322

1. 代码运行环境

本次实现运行在Java 1.8环境中，使用IntelliJ IDEA软件编写运行代码。

1. DES算法

1.算法原理

DES是一种对称加密算法，它使用同一个密钥来加密和解密数据，它还是一种分组加密算法，每次处理固定长度的数据段，DES分组的大小是64位。

首先将设定的64位密钥根据PC-1进行变换为56位

57,49,41,33,25,17,9,  
 1,58,50,42,34,26,18,  
 10,2,59,51,43,35,27,  
 19,11,3,60,52,44,36,  
 63,55,47,39,31,23,15,  
 7,62,54,46,38,30,22,  
 14,6,61,53,45,37,29,  
 21,13,5,28,20,12,4

PC-1变换表

将56位密钥分为2组每组28位记为c0,d0,由SLS左移表将c0,d0一步步进行左移产生cN和dN，1=<N<=16，将拼合的cNdN按照PC-2进行置换得到16个子密钥。

14,17,11,24,1,5,3,28,  
 15,6,21,10,23,19,12,4,  
 26,8,16,7,27,20,13,2,  
 41,52,31,37,47,55,30,40,  
 51,45,33,48,44,49,39,56,  
 34,53,46,42,50,36,29,32

PC-2变换表

其次加密数据的每个64位区块，首先将64位数据进行IP变换，接着讲变换IP分为32位的左半边L0和32位的右半边R0，我们接着执行16个迭代，对1<=n<=16，使用F函数。F函数输入两个区块——一个32位的数据区块和一个48位的密钥区块Kn ——输出一个32位的区块。定义+表示异或XOR。那么让n从1循环到16，我们计算

Ln = Rn-1

Rn = Ln-1 + f(Rn-1,Kn)

计算16轮之后就得到了L16R16。

58,50,42,34,26,18,10,2,  
 60,52,44,36,28,20,12,4,  
 62,54,46,38,30,22,14,6,  
 64,56,48,40,32,24,16,8,  
 57,49,41,33,25,17,9,1,  
 59,51,43,35,27,19,11,3,  
 61,53,45,37,29,21,13,5,  
 63,55,47,39,31,23,15,7

IP置换表

在加密过程中，F函数的工作原理如下，我们首先根据函数E拓展每个Rn-1，将其从32位拓展到48位，将48位输出与密钥执行异或运算在通过S盒最后得到32位输出。F函数的最后一步就是对S盒的输出进行P变换来产生最终输出32位。

32,1,2,3,4,5,  
 4,5,6,7,8,9,  
 8,9,10,11,12,13,  
 12,13,14,15,16,17,  
 16,17,18,19,20,21,  
 20,21,22,23,24,25,  
 24,25,26,27,28,29,  
 28,29,30,31,32,1

E置换

//S1  
 { 14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7 },  
 { 0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8 },  
 { 4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0 },  
 { 15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13 } },  
{//S2  
 { 15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10 },  
 { 3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5 },  
 { 0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15 },  
 { 13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9 } },  
{//S3  
 { 10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8 },  
 { 13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1 },  
 { 13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7 },  
 { 1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12 } },  
{//S4  
 { 7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15 },  
 { 13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9 },  
 { 10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4 },  
 { 3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14 } },  
{//S5  
 { 2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9 },  
 { 14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6 },  
 { 4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14 },  
 { 11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3 } },  
{//S6  
 { 12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11 },  
 { 10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8 },  
 { 9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6 },  
 { 4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13 } },  
{//S7  
 { 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1 },  
 { 13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6 },  
 { 1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2 },  
 { 6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12 } },  
{//S8  
 { 13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7 },  
 { 1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2 },  
 { 7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8 },  
 { 2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11 }

S盒

16,7,20,21,29,12,28,17,  
 1,15,23,26,5,18,31,10,  
 2,8,24,14,32,27,3,9,  
 19,13,30,6,22,11,4,25

P置换

最后我们将得到的L16R16逆转顺序得到R16L16，最后进行一个最终的变换IP-1，就得到了最终的密文。

40,8,48,16,56,24,64,32,  
 39,7,47,15,55,23,63,31,  
 38,6,46,14,54,22,62,30,  
 37,5,45,13,53,21,61,29,  
 36,4,44,12,52,20,60,28,  
 35,3,43,11,51,19,59,27,  
 34,2,42,10,50,18,58,26,  
 33,1,41,9,49,17,57,25

IP-1变换

解密就是加密的反过程，执行上述步骤，只不过在那16轮迭代中，调转左右子秘钥的位置而已。

2.算法实现

本次程序分为两部分：分别为通过原理实现DES和通过Java提供的DES工具包实现。

第一种首先创建ReplaceTable类存放DES算法中所有的置换表，如IP表、PC1、PC2等，其次创建DES类为DES主要实现过程，主要deal算法进行将输入明文进行64位分组，在将每个分组输入到descryUnit函数中进行加密或解密，其余实现了分组函数separate，F函数fFunction，生成子密钥函数generateKeys。

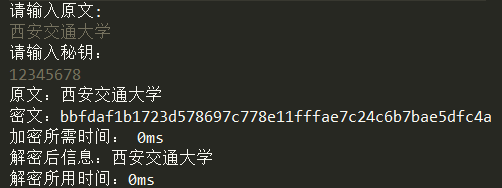
最后创建测试类，进行DES测试。

第二种实现使用DES工具包，要求输入密钥位数要求必须为8的倍数

3.DES测试结果

第一种

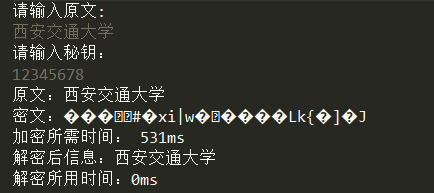
输入明文：西安交通大学 加密密钥：12345678 结果如下：



其加密时间和解密时间都在1ms以内

第二种

1. 输入明文：西安交通大学 加密密钥：12345678 结果如下：



1. RSA算法

1.算法原理

RSA采用非对称密码体制，其加密算法和解密算法使用不同的密钥，一个是公钥，一个是私钥，两个密钥都可以用于加密，解密时需要使用另一个密钥。通常用公钥加密私钥解密，因为公钥是近乎完全公开的，对于私钥加密的数据，有太多的人可以解密了。理论上 A 和 B 之间要通过 RSA 实现保密通信，需要 A 和 B 各自生成一组密钥，同时保管好自己的私钥；用对方的公钥加密要发送的消息，用自己的私钥解密对方发送过来的消息。

首先生成公钥和私钥：随意选择两个大的质数p和q，计算N=pq。根据欧拉函数，求得r=φ(N)=φ(p)φ(q)=(p−1)(q−1)。选择一个小于r的整数e，使e 与r互质。并求得e关于r的模反元素，命名为d（求d令 ed≡1(mod r)）。（模反元素存在，当且仅当e与r互质）。(N,e)是公钥，(N,d)是私钥。公钥发送给所有的通信对象（对服务器来说就是所有的客户端），私钥则必须保管好，防止泄露。将明文信息数字化得到m，使用c≡m^e(mod n)得到加密的密文，若要解密，使用私钥经过公式m≡c^d(mod n)可以得到原文的数字化再讲数字转换为明文即能得到明文。

RSA 加密时，对要加密数据的大小有限制，最大不大于密钥长度。

2.算法实现

本次程序分为两部分：分别为通过原理实现RSA和通过Java提供的RSA工具包实现。

第一种实现通过创建GCD类包括静态方法extGcd实现了扩展欧几里得算法求乘法逆元，实现Exponentiation类完成超大整数超大次幂然后对超大的整数取模。实现RSA类包括生成密钥和加密解密算法，本次实现我将输入的明文转换为二进制数据进行加密，最后将解密后的二进制数转换为字符串表示，最后实现RSATest类完成测试。

第二种实现使用RSA工具包：主要使用

*KeyPairGenerator* keyPairGen = *KeyPairGenerator*.getInstance("RSA");  
keyPairGen.initialize(1024);  
*KeyPair* key = keyPairGen.generateKeyPair();  
*PublicKey* publicKey = key.getPublic();  
*PrivateKey* privateKey = key.getPrivate();

以上代码生成密钥，从而完成加密

3.RSA测试结果

第一种：

p初始设置为：

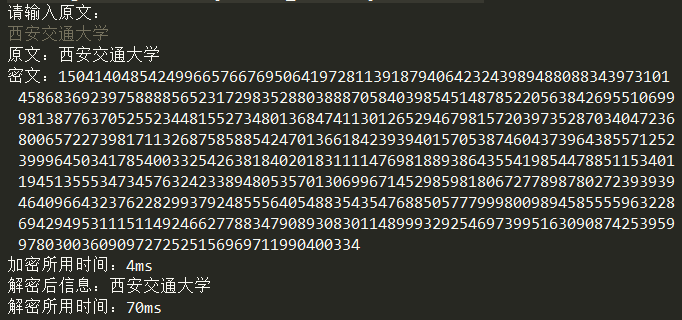
106697219132480173106064317148705638676529121742557567770857687729397446898790451577487723991083173010242416863238099716044775658681981821407922722052778958942891831033512463262741053961681512908218003840408526915629689432111480588966800949428079015682624591636010678691927285321708935076221951173426894836169

q初始设置为：

144819424465842307806353672547344125290716753535239658417883828941232509622838692761917211806963011168822281666033695157426515864265527046213326145174398018859056439431422867957079149967592078894410082695714160599647180947207504108618794637872261572262805565517756922288320779308895819726074229154002310375209

e初始设置为：4007

输入明文：西安交通大学 结果如下



第二种：输入明文西安交通大学 结果如下

