毕业论文中期报告

开题报告过后，主要的工作内容如下：

1.整理论文的写作思路，开题报告做补充。

2.编写RSA算法Java程序代码。

3.编写LLL算法Java程序代码。

4.找到自己的创新点。

5.毕业论文写作规划。

1.论文的写作思路整理：

（1）认真阅读理解RSA算法，对RSA进行安全性分析，安全参数的生成与选取：

a.RSA算法中p,q的选取（大素数生成），找到时间复杂度比较小的大素数生成算法。p,q对RSA算法安全性的影响。什么样的p,q才是最安全的参数。

b.e,d的选取。e,d对RSA算法安全性的影响，以及e,d对RSA算法时间复杂度的影响。

C.进行LLL算法破解RSA算法时，对RSA算法参数的特定要求。

（2）对RSA进行安全分析。

a.选择密文攻击（破译密文和骗取签名）。

破译密文：

Step1攻击者截获密文c(c=mmod n)后，随机选取.

Step2计算c’=mod(n).

Step3发送cc’给原消息接收者。接受者进行解密m’=(cc’).

Step4计算m=m’r.

骗取签名：

Step1产生两份消息m1，m2满足m’=m1m2mod n。

Step2获得A对m1，m 2的签名

Step3计算(m’)d mod n=m1dm2dmod n，获得A对非法消息m’的签名。

b.公共模数攻击。

如果系统中每个人拥有相同的模数n，即使每个人采用不同的公/私钥对，系统安全仍然会收到巨大的威胁。设消息为m，两个加密密钥为分别为e1，e2，且满足gcd(e1，e2)=1，两个密文为c1=me1mod n，c2=me2mod n，由于，攻击者很容易由扩展欧几里德算法得到r，s，满足re1+se2=1，那么c1rc2smod n=mre1+mse2mod n=m。因此，在一组用户之间共享模数是不安全的。

扩展的欧几里德算法：如何a,b是正整数，那么肯定存在u,v满足下面的方程。



,.

c.低加密指数攻击。

RSA密钥体系加密的时候时间复杂度较高，为了能更快的加密，很多时候大家用小指数e加密，小指数加密有很大的安全隐患，当e非常小的时候，<N的时候，我们可以很快的求出明文。

攻击方法如下：

当用户将一个m明文发送给n个用户的时候，并且用相同的小指数e加密。此时攻击者获取不同的密文。









后面就可以根据中国剩余定理求出同余方程组



可以求出唯一解x，即

x= 并且0 (1.1)

又因为m<min{,,,,}，所以

0 (1.2)

由上可知，x=。我们只需计算就可以计算出m。

2.编写RSA算法Java程序代码。

（1）n位p,q生成。主要使用大整数包。

Step1.随机产生一个n位数组p，数组中每个元素0-9里面随机生成。

Step2.若n位数组p中的第一个元素p[0]=0，那么另p[0]=1。

Step3.若p中的最后一个元素p[n-1]为偶数，那么另p[n-1]=p[n-1]+1。

Step4.生成一个数组x，x由小素数组成。

Step5.判断x中的元素能否被p整除。若能被p整除，p=p+2.继续上述判断。知道都不能被整除为止。

使用米勒拉宾素性检测。

Step1.将 n – 1 分解为 2s·d的形式，这里d是奇数。

Step2.将以下步骤(第 step3 到 step10)循环k次。

Step3.在 [2,n-2] 的范围中独立和随机地选择一个正整数a。

Step4.计算该序列的第一个值：x←ad mod n。

Step5.如果该序列的第一个数是 1 或者n-1，符合上述条件，n可能是素数，转到第 03 行进行一下次循环。

Step6.循环执行第 step7 到 step9，顺序遍历该序列剩下的s–1个值。

Step7.计算该序列的下一个值：x←x2modn。

Step8.如果这个值是 1 ，但是前边的值不是n-1,不符合上述条件，因此n肯定是合数，算法结束。

Step9.如果这个值是n-1，因此下一个值一定是 1，符合上述条件，n可能是素数，转到step3进行下一次循环。

Step10.发现该序列不是以 1 结束，不符合上述条件，因此 n 肯定是合数，算法结束。

Step11.已经对 k 个独立和随机地选择的 a 值进行了检验，因此判断 n 非常有可能是素数，算法结束。

1. 公钥e和私钥d的生成。使用扩展的欧几里得算法可以由e,（p-1）(q-1)求出d.具体算法如下：

Step1.另(p-1)(q-1),。

Step2.

Step3.循环上述两个步骤直到。

根据扩展的欧几里得算法求出d.

一个小例子：

P=47,q=59,n=p\*q=47\*59=2773,e=157.=46\*58=2668.d就可以用以下方法求出。



2668=157\*16+156,157=1\*156+1.

所以d=17.

1. 上述算法中所用到的指数运算和加解密运算全部使用快速模幂运算。快速模幂算法如下：

Step1.写出A的二进制表示形式。



Step2.计算(mod N),0ir



Step3.计算

=

=



1. 编写LLL算法的程序代码。

输入：格L中的任意一组基，，…，和规约参数

输出：以参数规约的LLL约化基

Step 1.k=2;

Step 2.while(){

2.1计算Gram-Schmidt正交化系数和，i=1,2,3,...,k-1;

2.2 for(j=1;j<k;j++) {

if>1/2 then = else =0  
 };

2.3 =;

2.4 if(){

2.4.1 交换和；

2.4.2 k=max{k-1,2};

}else k=k+1;  
 };

Step 3.输出，，...，。

1. 论文的重点研究方向

由于RSA密码体制是基本大整数分解困难问题，LLL算法是解决最短向量困难问题。所以我们必须把攻击RSA转化成解决格中的最短向量问题。

最短向量问题：

格中的一个欧式距离最短的向量就是最短向量。

a.RSA解密转化成解决最短向量问题.

mod(N)

M表示泄露的部分明文，x表示未知的部分明文，e是加密指数。

b.将mod(N)转化为求方程的结.

P(x)=

C.由于<X,所以可以把上式转化成求下面格中的最短向量。

M=

运用LLL算法解出最短向量求出x的值。

假设p()=\*N,未知，。

r = (1,,,,,,0)

行向量s如下。

S=r\*M = (1,(),,,,.0)

s的欧拉范式的上界是，如果X<.我们就可以认为s是近似最短向量，我们可以用LLL算法求出s。进而求出x。