# 密码算法实验（一）

**【实验目的】**

通过此实验，掌握RSA算法的基本原理及素数判定中的Rabin—Miller测试的原理，Montgomery快速模乘算法，了解公钥加密体制的优缺点以及它的应用方式。

**【实验仪器】**

(1)安装Windows操作系统的PC；

(2)安装VC6.0以上版的编译器。

**【实验原理】**

(1)加密—解密过程

在密码体制中，参与方分为：发信方，收信方及攻击者，有明文，密文，密钥，加密算法，解密算法五大要素。非对称加解密过程如图11-1。

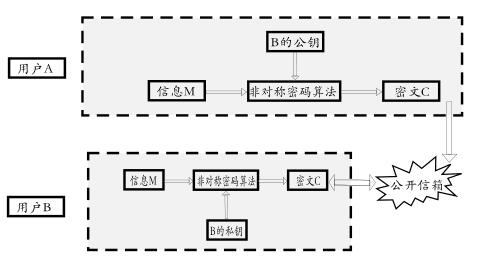


图11-1 非对称密码体制加解密过程

(2)RSA算法介绍

非对称密码算法是指一个加密系统的加密密钥和解密密钥是不相同，或者说不能从其中一个推导出另一个。在非对称密码算法的两个密钥中，一个是用于加密的密钥，它是可以公开的称为公钥；另一个是用于解密的密钥，是保密的，称为私钥。非对称密码算法解决了对称密码体制中密钥管理的难题，并提供了对信息发送人的身份进行验证的手段，是现代密码学的最重要的发明和进展。

RSA密码体制是目前为止最为成功的非对称密码算法，是在1977年由Rivest、Shamir和Adleman提出的第一个比较完善的非对称密码算法。它的安全性是建立在“大数分解和素性检测”这个数论难题的基础上，即将两个大素数相乘在计算上容易实现，而将该乘积分解为两个大素数因子的计算量相当大。虽然的安全性还未能得到理论证明，但经过20多年的密码分析和攻击，迄今仍然被实践证明是安全的。

（3）RSA算法实现

①RSA源代码

见附件的RSA算法源代码。

②Miller-Rabin算法

该检测法基于Gary Miller的部分想法，由Michael Rabin发展。该检测法描述如下：首先选择一个待测的随机数n，计算b，2b是能够整除n-1的2的最大幂数。然后计算m，使得n= 2bm+1。

●随机选取a∈（1，n）；

●设j=0，计算z≡ ammod n；

●若z=1或者z=n-1，则n通过测试，可能是素数；

●如果j﹥0且z=1，则n不是素数；

●令j=j+1。若j﹤b且z≠n-1，令n=z2mod n 然后回到上步。若z=n-1，则n通过测试，可能是素数；

●若j=b且z≠n-1，则n不是素数；

对a选取k个不同的随机值，重复k次这样的测试。如果n能都通过测试，则断定n不是素数的概率不超过4-k。

③Montgomery算法描述

选择与n互素的基数R，为计算方便，它通常是机器字长的倍数；并且选择R-1及n'，满足0< R-1<n,0< n'<R,使得RR-1 -nn' =1。对0≤T＜Rn的任意整数T，Montgomery给出求取模乘法TR-1mod n的快速算法M（T）：

Function M（T）

λ=(mod R)n mod R ; 0≤λ≤R,

t=(T+λn)/R

if t≥n return (t-n)

else return t

从上面的M（T）运算可以看出，因为λn≡Tnn'≡-T mod R，t为整数；因tR≡T mod n，得t≡ mod n。由于0≤T+＜Rn+Rn，M（T）的运算结果范围是0≤t＜2n。

由于用R的剩余系表示整数，Montgomery算法会带来一些附加计算。在计算z=ab modn（其中a，b＜R）之前，预先求出A=aRmodn和B=bRmodn，再求Z=M(A,B)=ABR-1mod n=(aR)(bR)R-1mod n最后的计算结果也要做相应调整

Z=M(Z)=ZR-1mod n=abRR-1mod n =ab mod n

可见这种方法适合于像RSA这样多次取模乘法的取模幂乘运算。对于整数e和任意整数m，加密或解密m即是求解 m6mod n。对输入变换得到M=mR mod n之后，取模幂乘的乘法-平方循环Montgomery乘法来完成，最后调整得到最终的加密或解密信息。把e描述成e=el(n)-1el(n)-2........e1e0，其中l（n）表示e的位数。取模幂乘运算过程可描述成：

M：=mR mod n；

Z：=lR mod n；

for i in l（n）-1 downto 0 loop

Z：=（Z，Z）；

if ei=1 then

Z：=Mn（Z，M）；

end if ；

end loop；

z= ZR-1mod n;

**【实验内容及步骤】**

根据RSA算法，利用已知参数：p=7，q=17，e=5，手工计算公私钥，并对明文进行加密，然后对密文解密。

RSA算法的手工计算方式如下：

1. 选两个互异的大素数p和q

2. 计算n=p×q，φ(n)是n的欧拉函数值

3. 选一个整数e，满足1＜e＜φ(n)，且gcd（φ(n)，e）=1

4. 计算d，满足de≡1 mod φ(n)，即乘法逆元

5. 以{e，n}为公开钥，{d，p，q}为秘密钥

6. 加密时首先将明文比特串分组，使得每个分组对应的十进制数小于n，即分组长度小于，然后对每个明文m，作加密运算：

7. 解密时对密文分组的解密运算为：

例：p=7，q=17，e=5，m=19

则n=119，φ(n)=96，因为77×5=385=4×96+1，故d=77

因此公开钥为{5,119}，秘密钥为{77,7,17}

由明文m=19，则密文；

解密过程为。

**【实验报告】**

  按RSA算法的要求确定两个素数，完成RSA算法的计算，得到加解密过程。整理程序代码文档，记录程序运行结果。

程序随机选取两个大素数p和q，本次运行取p=929，q=823。

计算出公钥：(2665, 764567)，私钥：(42649, 764567)。

运行结果如下图所示：

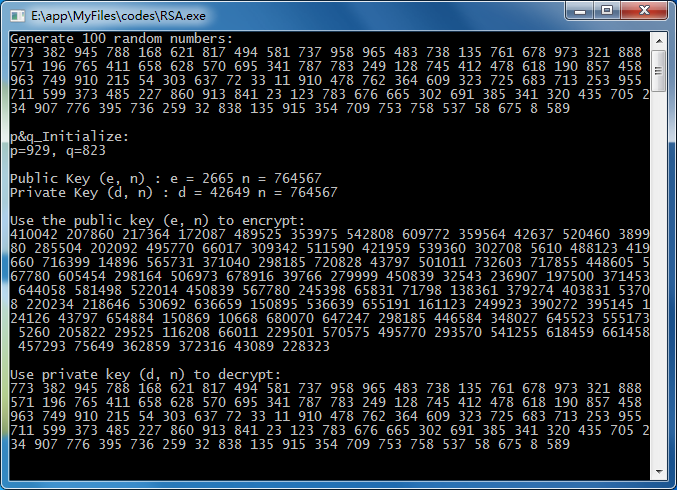


图11-2 程序运行结果