RSA实验报告

1811430 王瀚威

一、实验目的

通过实际编程了解公钥密码算法RSA的加密和解密过程，加深对公钥密码算法的了解和使用。

二、实验原理

序列密码和分组密码算法都要求通信双方通过交换密钥实现使用同一个密钥，这在密钥的管理、发布和安全性方面存在很多问题，而公钥密码算法解决了这个问题。

公钥密码算法是指一个加密系统的加密密钥和解密密钥是不同的，或者说不能用其中一个推导出另一个。在公钥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。公钥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，是现代密码学最重要的发明。

RSA密码体制是目前为止最成功的公钥密码算法，它是在1977年由Rivest、Shamir和Adleman提出的第一个比较完善的公钥密码算法。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。虽然它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。

RSA算法的基本内容

RSA算法简介

RSA公开密钥密码体制是一种使用不同的加密密钥与解密密钥，“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制 。在公开密钥密码体制中，加密密钥（即公开密钥）PK是公开信息，而解密密钥（即秘密密钥）SK是需要保密的。加密算法E和解密算法D也都是公开的。虽然解密密钥SK是由公开密钥PK决定的，但却不能根据PK计算出SK 。

正是基于这种理论，1978年出现了著名的RSA算法，它通常是先生成一对RSA密钥，其中之一是保密密钥，由用户保存；另一个为公开密钥，可对外公开，甚至可在网络服务器中注册。为提高保密强度，RSA密钥至少为500位长，一般推荐使用1024位。这就使加密的计算量很大。为减少计算量，在传送信息时，常采用传统加密方法与公开密钥加密方法相结合的方式，即信息采用改进的DES或IDEA对话密钥加密，然后使用RSA密钥加密对话密钥和信息摘要。对方收到信息后，用不同的密钥解密并可核对信息摘要。

RSA是被研究得最广泛的公钥算法，从提出到现在已近三十年，经历了各种攻击的考验，逐渐为人们接受，普遍认为是目前最优秀的公钥方案之一。1983年麻省理工学院在美国为RSA算法申请了专利 [3] 。

RSA密码体制是目前为止最成功的公钥密码算法，它是在1977年由Rivest、Shamir和Adleman提出的第一个比较完善的公钥密码算法。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。虽然它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。

RSA算法描述

1.选择两个不同的大素数p和q计算乘积$n=pq,\varphi(n)=(p-1)(q-1)$,$\varphi(n)$为欧拉函数。

2.任意选取一个大整数e，使其满足$gcd\left(e,\varphi\left(n\right)\right)=1$，即e和$\varphi\left(n\right)$互素，整数e用做加密钥（注意：e的选取是很容易的，例如，所有大于p和q的素数都可用）

3.确定解密密钥d，满足$\left(de\right)mod\varphi\left(n\right)=1$,即$de=k\varphi\left(n\right)+1,k\geq1$是任意的整数，所以，若知道e和$\varphi\left(n\right)$，则很容易计算出d

4.公开整数n和e，秘密保存d

5.将明文m加密成密文c，加密算法为 $c=E\left(m\right)=m^emodn$

6.将密文c解密为明文m，解密算法为 $m=D\left(c\right)=c^dmodn$

如果攻击者获得了n、e和密文c，为了破解密文必须计算出私钥d，为此需要先分解n。当n的长度为1024比特时，在目前还是安全的，但从因式分解技术的发展来看，1024比特并不能保证长期的安全性。为了保证安全性，要求在一般的商业应用中使用1024比特的长度，在更高级别的使用场合，要求使用2048比特长度。

RSA算法所依赖的数学难题

根据数论，寻求两个大素数比较简单，而将它们的乘积进行因式分解却极其困难，因此可以将乘积公开作为加密密钥；换句话说，RSA的安全性依赖于大数分解，RSA的破解难度与大数分解难度等价。

大整数因数分解问题可以描述为：

给定两个大素数p,q,计算乘积$ p∗q=n$很容易；
给定大整数n，求n的素因素p,q使得 $n=p∗q$则非常困难.

RSA算法所涉及到的本课程所学的数学原理

互素：如果两个正整数没有除了1以外的其它公因子，则为互素

剩余类与m互素：在模m的一个剩余类中，若有一个数与m互素，则该剩余类中所有数都与m互素，此时称该剩余类与m互素

欧拉函数：设m是正整数，在m的所有剩余类中，与m互素的剩余类的个数成为m的欧拉函数

欧拉定理：欧拉定理表明，若n,a为正整数，且n,a互质，则:$a^{\varphi(n)}\equiv1(\mathrm{mod}\,n)$

对RSA的攻击方法

RSA 算法的安全性依赖于大整数分解的困难性。 最直接的攻击方法是分解 n 得到 p,q，进而基于 e 计算 d，随着计算机运算能力的不断提高，通过二次筛法已能分解 180 多位的十进制素数，增加 p,q 的长度已成为许多安全应用系统的加密要求。 另一方面，利用系统设计和实现的缺陷， 人们也提出了一些基于非因子分解方式破解 RSA 算法的方案。 目前，对 RSA 算法的攻击主要有以下几种：

（1）对模数n的因子分解

分解模数 n 是最直接的攻击方法，也是最困难的方法。 攻击者可以获得公钥 e 和模数 n；如果 n=p*q 被成功分解，则攻击者可以计算出$φ(n)=(p-1)(q-1)$，进而从 $ed≡1modφ(n)$解得私钥 d。

（2）对RSA的公共模数攻击

若一个多用户系统中只采用一个模数 n，不同的用户拥有不同的e 和 d，系统将是危险的。 在此系统中，若有同一消息用不同的公钥加密，这些公钥共模且互质，那该信息无需私钥就可解密。 举例来说，设P 为信息明文，两个加密公钥为 e1和 e2，公共模数是 n，有：$C1=Pe1(modn) $和 $C2=Pe2(modn)$。如果攻击者获得 n、e1、e2、C1和 C2，就能得到 P。 因为 e1和 e2互质，故用欧几里德(Euclid)算法能找到 r 和 s，满足：$r*e1*+*s*e2=1$，设 r 为负数，则$(C1^-1)^-r*C2^s*=*(Pe1(modn))r*(Pe2(modn))s=(Pr*e1*+*s*e2)modn=Pmodn$，如果 P<n ，则p被获取。

（3）对RSA的小指数攻击

如果RSA系统的公钥e选取较小的值，可以使得加密和验证签名的速度有所提高，但是如果e的选取太小，就容易受到攻击。

例如：使用同一个系统的三个用户，分别使用不同的模数n1,n2,n3，但是都选取e=3；另有一个用户欲将同一明文消息m发送给以上三人，使用个人的公钥加密得到：

（4）RSA选择密文攻击

选择密文攻击是指攻击者能选择不同的密文，并拥有对应的明文，由此推出想要的信息。一般攻击者会伪装若干信息，让拥有四亚欧的用户签名，由此获得有用的明文-密文对，然后推算想要的信息，这一个工程量的大小要视情况而定。

$C1≡m^3mod n1$

$C2≡m^3mod n2$

$C3≡m^3mod n3$

一般情况下，n1,n2,n3互素，否则会比较容易求出公因子，从而安全性大幅度的减低，根据中国剩余定理，可以通过C1、C2、C3计算：$C≡m^3mod (n1n2n3)$

三、实验环境

运行Windows操作系统的PC机，具有VC等语言编译环境

四、实验内容和步骤

1、为了加深对RSA算法的了解，根据已知参数：p = 3，q = 11，m = 2，

手工计算公钥和私钥，并对明文m进行加密，然后对密文进行解密。

2、编写一个程序，用于生成512比特的素数。

3．利用2中程序生成的素数，构建一个n的长度为1024比特的RSA算法，

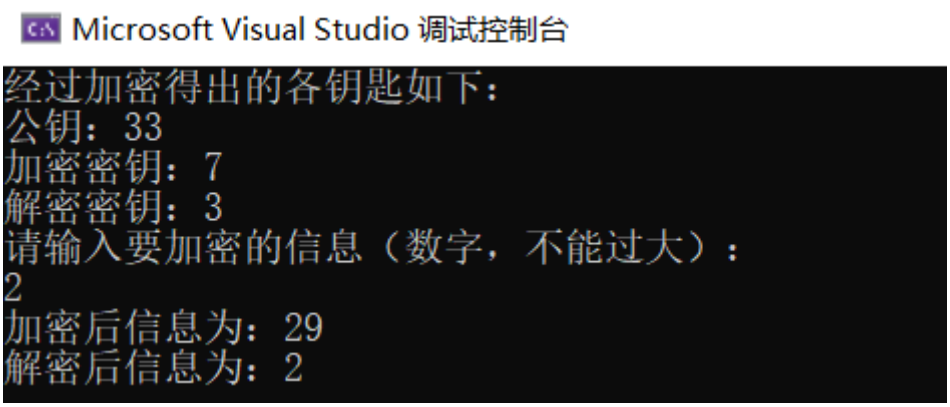
利用该算法实现对明文的加密和解密。

4、在附件中还给出了一个可以进行RSA加密和解密的对话框程序RSATool，运行这个程序加密一段文字，了解RSA算法原理。

五、实验报告

1.实验步骤一的实现

代码见附件，实现结果如图



2.实验步骤二的实现

如何生成一个随机的大素数?

- 伪素数
  伪素数生成过程如下：
  ① 随机选取一个大奇数n
  ②将从2开始的53个素数排列成数组，作为工具a[i]
  ③令i=0，计算x=n%a[i]
  ④ 判断，若x=0，说明n显然是合数，回到步骤1。若不等于0，说明暂且可以 认为n是素性的，进行步骤5。
  ⑤检测n%其他的a[i]． 当i=52，则将n视为一个伪素数，然后作为素数生成部分的结果。
  以上是生成过程，举例为前53个素数。其实在真正的实际应用之中，应当将所有2000以内的素数都纳入工具。

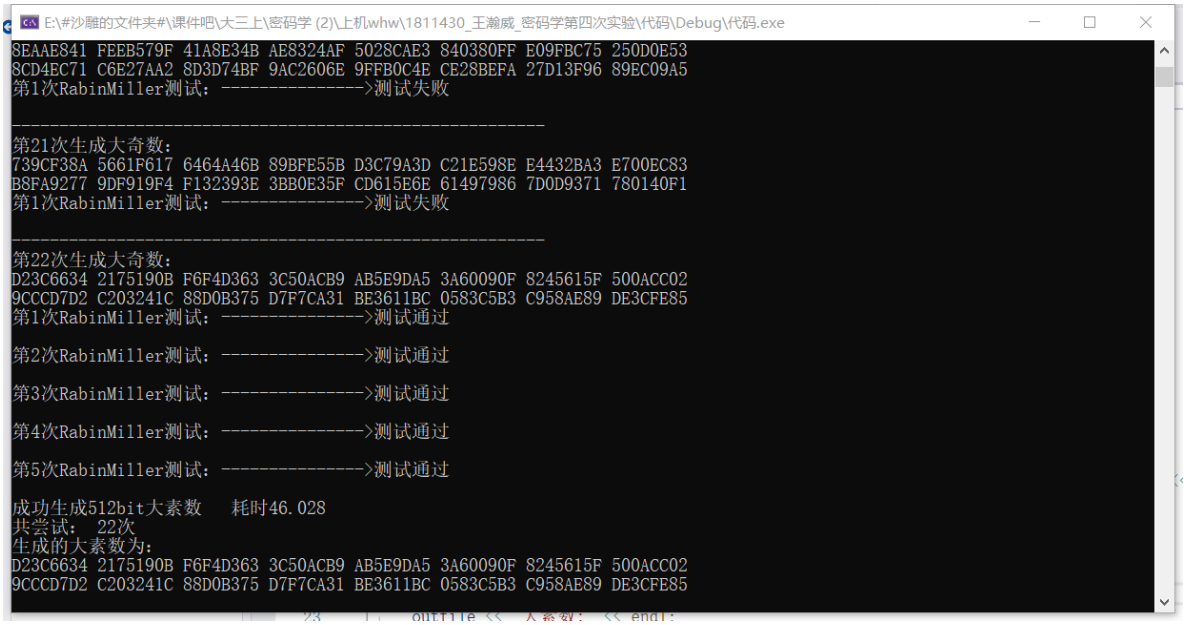
- 素性检测
- Miller-Rabin(n,t)
  输入：一个大于3的奇整数n和一个大于等于1的安全参 数t(用于确定测试轮数)。
  输出：返回n是否是素数(概率意义上的，一般误判概率小于(1/2)80即可) 。
  1、将n-1表示成2sr，(其 中 r是奇数)
  2、 对i从1到 t 循环作下面的操作：
  2.1选择一个随机整数a(2≤a ≤n-2)
  2.2计算y ←ar mod n
  2.3如果y≠1并且y ≠n-1作下面的操作,否则转3：
  2.3.1 j←1;
  2.3.2 当j≤s-1 并且y≠n-1循环作下面操作,否则跳到 2.3.3：
  {计算y ←y2 mod n;
  如果 y=1返回 合数；
  否则 j←j+1; }
  2.3.3如果y ≠n-1 则返回 合数；
  3、返回素数。

实现代码见附件

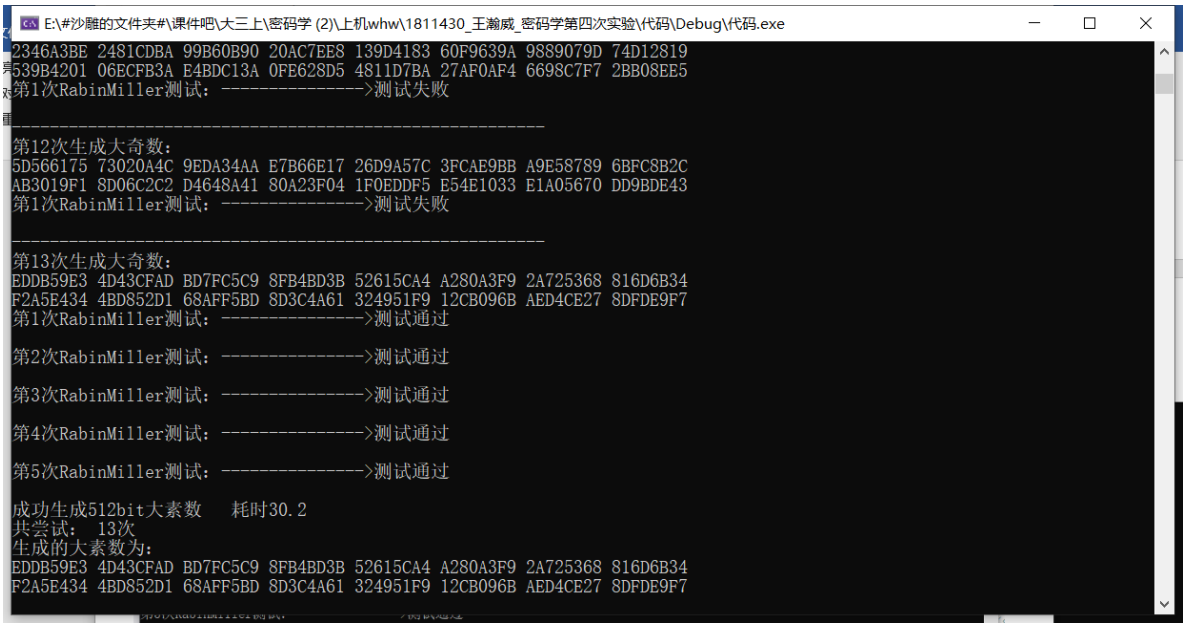
3.实现加密解密

实现结果如图

生成大素数p：



生成大素数q:



生成公钥私钥:

```
公钥n = p * q
/*********************************************************/
公钥n为:
C3560E1B 5C1CA251 D43101EA 72C2AD89 226776B2 F1415317 78CDC94E 3A089455 721E80C4 82F119C0 ADB4FD73 EA4C61F4 54A8AF73 A2A
C36C1 FBBC8F80 7EF3A7C9
53380417 69B617D9 3D739C85 15137C8F 9E80708D 260A3127 14457642 CE8756B9 E751FE87 BB1F2954 3C5EDC43 C9CCA9A2 01010818 27B
F4D6D 8551E904 BAF19F53

/*************************************************************************/
计算公钥e 和秘钥d
/*************************************************************************/
公钥e:
2018FFFF 540BA86B A8BCAF82 00A2EEE0 2F834794 9078B582 AAB08CA5 57D893B9
B217B515 795A65BD 13084381 6D0B0B2A 1797FD45 BD2AF490 09334CD3 7F41D743

秘钥d:
390C5E94 ABEF14D0 FB5104FB BA22513D 0FB2D11B F2C0F385 95E25D11 FD198ECF 6B96F9C1 2DAC4DCB 8C96E2B0 45734636 4CFE4338 606
6E2BD B28AB933 67DAAA1B
7B135A89 3CAC5A1D ACA3DFB6 8FC785C8 AA309CE1 C07CDD4A D8C6BC31 B5B45B8A C8A41B42 7603013A D54BBB3D D19A93C6 000963A2 253
D1E9F 8BCBC3E6 A46ADACB
```

加解密:

```
/********************************************************************************/
随机生成明文m
/********************************************************************************/
明文m为：
ABA5FF13 07899EB7 0D9582BE 5F6A8E11 557B34CA 264CCCAF 49D72642 99B810C1
42E9E8EF 6D75553F 229491EA 47F35969 491E1F88 FFB7AA5B 6AA36A0F 4C92857C

/********************************************************************************/
用秘钥e加密m,得到密文c
/********************************************************************************/
密文c为：
85F21AC1 A114A1CD F43487EF 1F2DD404 6B3B6D30 BA102E6D 3A32B45A 64980F65 4FDA44A3 C1DCFE55 2219B626 A2B29F0C A0FF85B8 F83
69C4F E388248F D1552A34
D38A3D35 AB4CA471 FB3AD0F9 2429709E C6842D98 4B36E36D 290AD14C 53EE61D8 0814AF26 455095A0 F59AC531 7D0A6FCE 213B54F5 9CF
DD7CE 035F7227 1DD23D0E

/********************************************************************************/
用公钥d对c解密，得到明文m2
/********************************************************************************/
明文m2为：
ABA5FF13 07899EB7 0D9582BE 5F6A8E11 557B34CA 264CCCAF 49D72642 99B810C1
42E9E8EF 6D75553F 229491EA 47F35969 491E1F88 FFB7AA5B 6AA36A0F 4C92857C

请按任意键继续. . .
```

输出到output.txt中结果如下：

output.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
大素数:
D23C66342175190BF6F4D3633C50ACB9AB5E9DA53A60090F8245615F500ACC029CCCD7D2C203241C88D0B375D7F7CA31BE3611BC0583C5B3C958AE89DE3CFE85
大素数:
EDDB59E34D43CFADBD7FC5C98FB4BD3B52615CA4A280A3F92A725368816D6B34F2A5E4344BD852D168AFF5BD8D3C4A61324951F912CB096BAED4CE278DFDE9F7
公钥n为:
C3560E1B5C1CA251D43101EA72C2AD89226776B2F141531778CDC94E3A089455721E80C482F119C0ADB4FD73EA4C61F454A8AF73A2AC36C1FBBC8F807EF3A7C95338041769B617D93D739C851
5137C8F9E80708D260A312714457642CE8756B9E751FE87BB1F29543C5EDC43C9CCA9A20101081827BF4D6D8551E904BAF19F53
公钥e:
2018FFFF540BA86BA8BCAF8200A2EEE02F8347949078B582AAB08CA557D893B9B217B515795A65BD130843816D0B0B2A1797FD45BD2AF49009334CD37F41D743
秘钥d:
390C5E94ABEF14D0FB5104FBBA22513D0FB2D11BF2C0F38595E25D11FD198ECF6B96F9C12DAC4DCB8C96E2B0457346364CFE43386066E2BDB28AB93367DAAA1B7B135A893CAC5A1DACA3DFB6
8FC785C8AA309CE1C07CDD4AD8C6BC31B5B45B8AC8A41B427603013AD54BBB3DD19A93C6000963A2253D1E9F8BCBC3E6A46ADACB
明文m为:
ABA5FF1307899EB70D9582BE5F6A8E11557B34CA264CCCAF49D7264299B810C142E9E8EF6D75553F229491EA47F35969491E1F88FFB7AA5B6AA36A0F4C92857C
密文c为:
85F21AC1A114A1CDF43487EF1F2DD4046B3B6D30BA102E6D3A32B45A64980F654FDA44A3C1DCFE552219B626A2B29F0CA0FF85B8F8369C4FE388248FD1552A34D38A3D35AB4CA471FB3AD0F9
2429709EC6842D984B36E36D290AD14C53EE61D80814AF26455095A0F59AC5317D0A6FCE213B54F59CFDD7CE035F72271DD23D0E
明文m2为:
ABA5FF1307899EB70D9582BE5F6A8E11557B34CA264CCCAF49D7264299B810C142E9E8EF6D75553F229491EA47F35969491E1F88FFB7AA5B6AA36A0F4C92857C
```