哈尔滨工业大学

计算机科学与技术学院

《信息安全概论》

实验报告

计算机科学与技术学院

计算机系网络教研室制

|  |  |
| --- | --- |
| 课程名称： | 信息安全概论 |
| 实验名称： | RSA加密实验 |
| 指导教师： | 韩琦 |
| 学生姓名： | 马玉坤 |
| 组 号： | 3 |
| 实验日期： | 2018年3月12日-13日 |
| 实验地点： | 哈尔滨工业大学 |
| 实验成绩： |  |

一、实验目的

通过用RSA算法对实际数据进行加密和解密来深刻了解RSA的运行原理及其特点，能够编译并分析RSA算法，进而加深对非对称加密算法的理解与认识。

掌握基础的数论知识（如同余、互质、欧拉函数、欧拉定理等）以及常用的数论算法（如快速幂、Miller-Rabin素数测试、欧几里得算法等）。

二、实验环境

**操作系统**：Windows 10

**集成开发环境**：Visual Studio Enterprise 2017

**验证软件**：CAP（Cryptographic Analysis Program v4）软件，该软件位于桌面“RSA加解密实验”文件夹中的bmrsall.zip压缩包中。

三、实验内容与实验要求

1. **实验内容**

**任务一：RSA加解密算法的原理**

1. 产生密钥过程：
2. 随机产生两个长度为位的素数 和 ；
3. 计算公钥 ，其中是位的长度。
4. 随机产生一个加密密钥 ，，其中。

注：这是保证解密密钥 有解的充要条件， 称为n的欧拉函数，值为:

1. 求解解密密钥 ，为解密密钥的逆元 ，此公式原方程为
2. 由此，公钥，加密密钥，解密密钥全部产生。
3. 对明文的加密与对密文解密过程

      1) 加密: ，其中表示明文，表示密文

      2) 解密: ，其中表示明文，表示密文

1. 画出算法流程图。

**任务二：使用C语言实现IDEA加密算法**

编译运行下列代码，并对代码进行分析：

1. **原理分析**
2. 欧拉定理

若{\displaystyle n,a}为正整数，且互素（即{\displaystyle \gcd(a,n)=1}），则  
即{\displaystyle a^{\varphi (n)}}与在模下[同余](https://zh.wikipedia.org/wiki/%E5%90%8C%E4%BD%99)；为欧拉函数。

1. RSA算法的正确性

由欧拉定理可以得到：

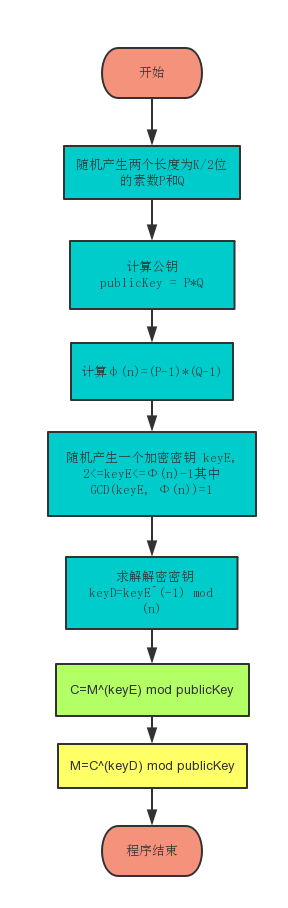
那么解密过程为：

c. RSA算法的安全性

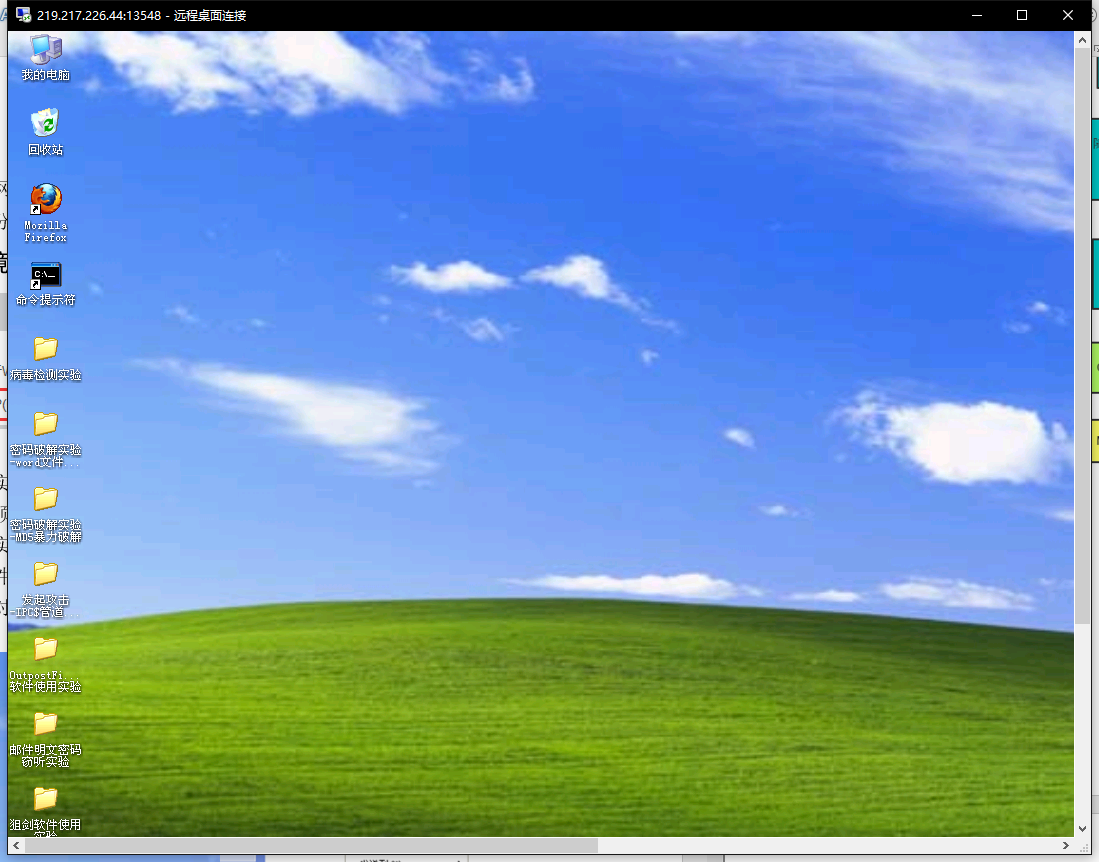
RSA算法的破解需要对n进行质因数分解，而质因数分解在当今是一个无法在多项式时间内解决的难题。因此，当n非常大且是两个大质数的乘积时，将n分解质因数将无法在可以承受的时间内求解。因此RSA算法至今仍未有可靠的攻击方法。

四、实验过程与分析

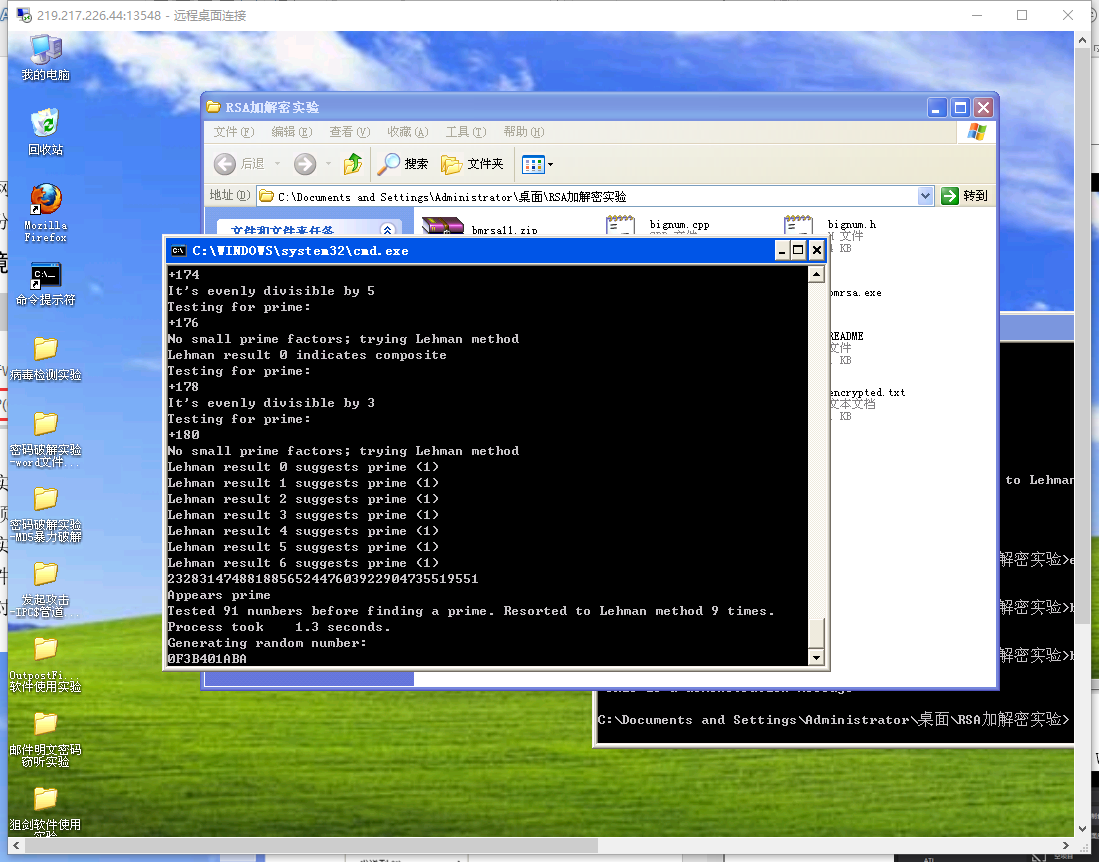
1. **算法流程框图**



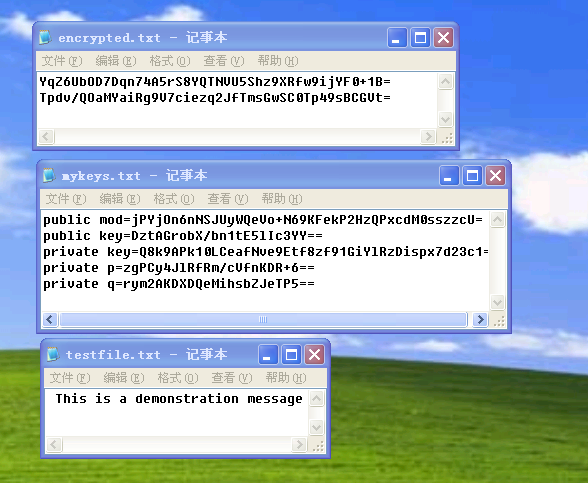
1. **在远程电脑上验证RSA算法**
2. 连接到远程电脑



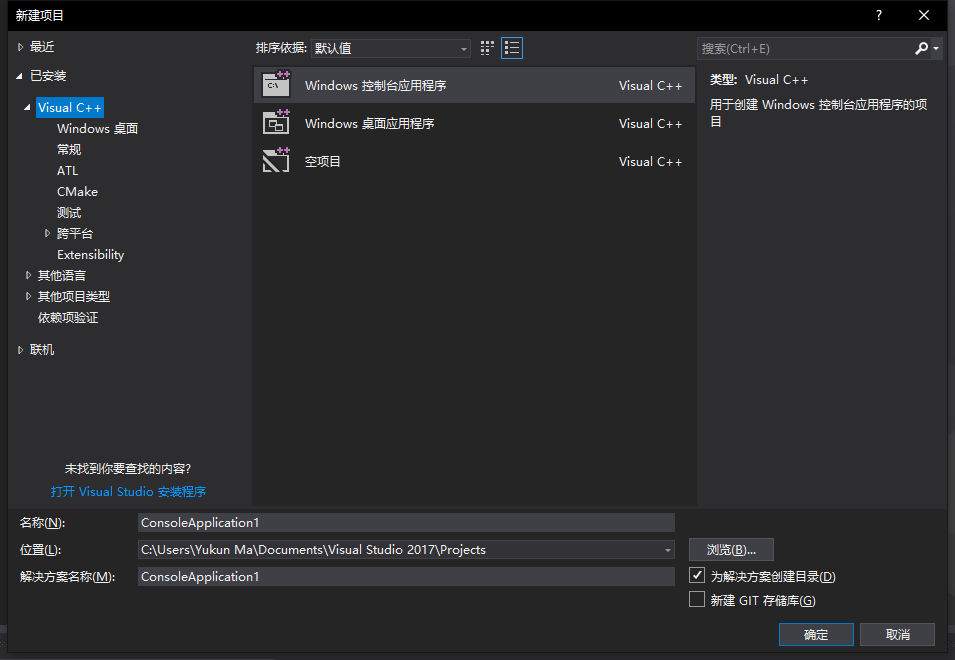
1. 运行bmrsall解压后的demo.bat文件



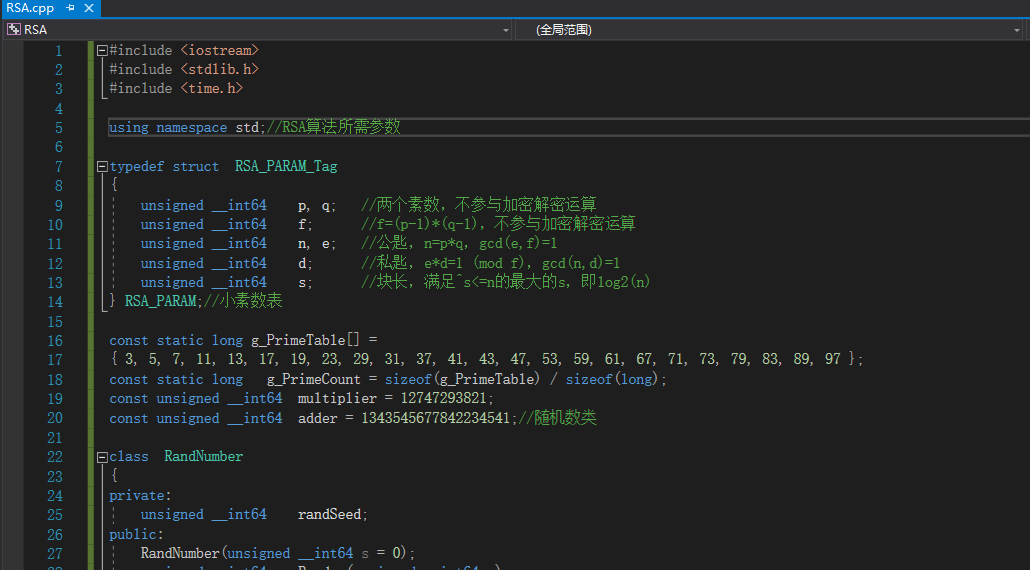
1. 检查encrypted.txt、mykeys.txt、testfile.txt等三个文件



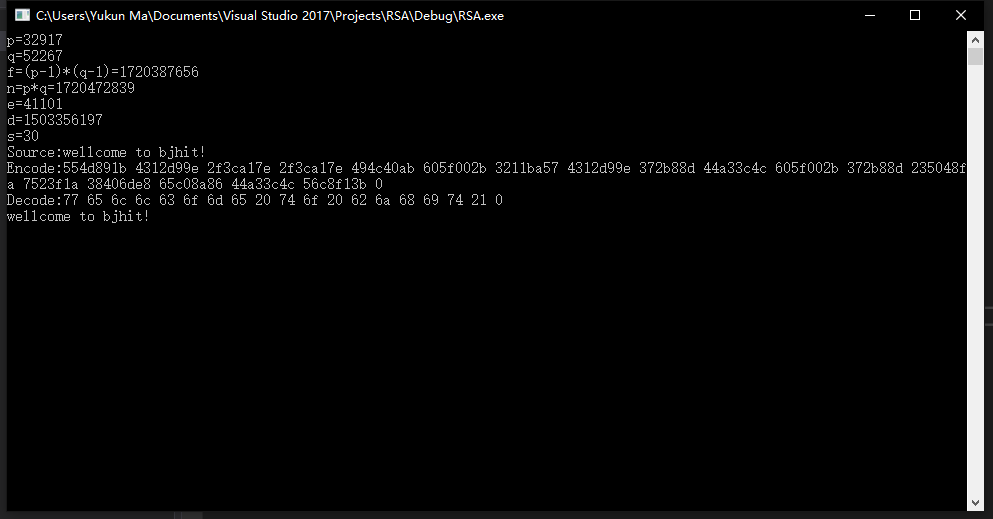
1. **用C语言实现RSA算法**
2. 打开Visual Studio，新建Windows控制台应用程序



1. 编写RSA加密算法



1. 运行结果



1. **RSA算法分析**

首先用RSA\_PARAM RsaGetParam(void)方法随机产生一个RSA加密参数。在此过程中，需要：

1. 使用RandomPrime方法随机产生数，然后用RabinMiller方法进行素数判定。RabinMiller方法使用了Miller-Rabin算法，可在较短时间内实现素数判定。
2. 之后随机产生一个与互质的e
3. 求出在模意义下的逆

之后，就可以使用RSA密钥对信息进行加密和解密。其中在加密和解密过程中，需要使用快速幂算法来快速计算整数的幂。

五、实验结果总结

1. **实验结果分析**

在使用远程计算机验证RSA算法时，三个文件的作用分别是：

* encrypted.txt：加密后的文件
* mykeys.txt：该次加密过程中所用到的公钥、私钥、模数以及两个指数p和q
* testfile.txt：原文件

在使用C语言实现RSA算法时，产生的素数，，,,加密密钥,解密密钥

明文为：wellcome to bjhit!

对应的ASCII码为：77 65 6c 6c 63 6f 6d 65 20 74 6f 20 62 6a 68 69 74 21

密文为：1efbdb72 3332b809 6c8955b4 6c8955b4 14964289 7d075fa 55300bbb 3332b809 123bfb1c 24d4f813 7d075fa 123bfb1c c170db2 4f4fc9a5 6132ce27 4d026386 24d4f813 73b49cb

解密后的明文为：77 65 6c 6c 63 6f 6d 65 20 74 6f 20 62 6a 68 69 74 21

恰为原明文的ASCII码，因此过程有效。

1. **通过本实验，论述RSA算法的加密原理是什么？**

RSA算法基于欧拉定理，而RSA算法的破解需要对n进行质因数分解，而质因数分解在当今是一个无法在多项式时间内解决的难题。因此，当n非常大且是两个大质数的乘积时，将n分解质因数将无法在可以承受的时间内求解。因此RSA算法至今仍未有可靠的攻击方法。

1. **在上述算法中哪些模块是该算法的核心模块？**
2. 素数生成模块：用于随机生成素数（随机产生数并对该数的素性进行判定）
3. 求逆模块：使用拓展欧几里得算法求一个数在某个模数下的逆
4. 求幂模块：使用快速幂算法进行加密解密
5. **对于一个RSA加密算法的密文，要得到明文需要哪些要素？**

以下两种情况均可：

1. 密文、公钥和解密密钥
2. 密文、公钥、加密密钥以及分解公钥所得到的p和q两个素因子
3. **实验心得体会**

通过本次实验，我了解到了RSA算法的原理和特性，对RSA算法的发明者巧妙利用质因数分解问题的困难程度而提高安全性的想法十分佩服。此外，我还实现了RSA算法，对RSA算法的步骤进行了初步的了解。在整个过程中，我对数论也产生了很深的兴趣，对算法每一步用到的数论知识和数论算法也进行了研究，得到了极大的收获。

1. **改进意见**
2. 公式显示不正常，上下标无法正常显示，推荐使用LaTeX+MathJax
3. “keyE-1为解密密钥keyD的逆元”似乎有错，应为“keyE为keyD的逆元”
4. 多人用同一账号的体验十分别扭，希望能给每个人一个独立的账号