

**实 验 报 告**

（ 2021/2022 学年 第 2 学期）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 算法分析与设计 | | | | | |
| 实验名称 | 密码算法 | | | | | |
| 实验时间 | 2022 | 年 | 5 | 月 | 30 | 日 |
| 指导单位 | 南京邮电大学计算机学院、软件学院 | | | | | |
| 指导教师 | 陈春玲 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 张宸冉 | 学 号 | B20060527 |
| 学院(系) | 计算机学院 | 专 业 | 计算机科学与技术 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | | 密码算法 | | | | 指导教师 | | | 陈春玲 |
| 实验类型 | | 验证 | 实验学时 | | 2 | 实验时间 | | | 2022.5.30 |
| 1. **实验任务和目的** 2. 目的：   了解现代密码学的基本原理和数论的基础知识，掌握非对称密码体制的著名代表RSA加密算法的工作原理和流程，并设计实现一个简单的密钥系统。   1. 任务：   了解加/解密的基本原理和工作过程，用公开密钥对明文进行加密，并用私人密钥对密文进行解密，构造一个简单的RSA公开密钥系统。 | | | | | | | | | |
| 1. **实验环境**   硬件：AMD Ryzen 7 4800H with Radeon Graphics  软件：Windows操作系统，Visual C++ 6.0。 | | | | | | | | | |
| **三、核心算法设计及分析**  **1、核心算法设计**  RSA密码算法的设计流程如下：   1. 选择两个较大的素数p和q 2. 计算p和q的乘积n，求取n的欧拉函数，由于p和q是两个素数，所以n的欧拉函数： 3. 随机选取一个整数e，使得e和互质，即，其中gcd是求取两个数最大公约数的函数 4. 计算e的关于模的乘法逆元d， 5. 这样就得到了公钥k1=｛e，n｝，私钥k2=｛d，n｝ 6. 加密解密过程，设明文分组为M，密文分组为C，则       对于计算e关于的乘法逆元，有以下方法  e\*0+φ\*1=φ (1)  e\*1+φ\*1=e (2)  则：e\*(0-1\*(φ/e))+.....=φ%e  由于e和φ互质，因此一定有某一次运算后，等式右侧的φ%e==1。  此时左侧等式中e所乘的系数就是所要求的d，即e^-1  加密解密时需要计算的幂表达式远大于计算机可以表示的范围，可以使用按模计算原理，将计算拆解开：  这样对于单独的e求解模数就比较容易实现。  在程序中可以用如下方法展示：  C = 1;  for (int i = 1; i <= e; i++){  C = C \* M % n; //使用按模计算公式  }  RSA算法实现了加密通信，用户可以根据自己生成的p和q产生公钥和私钥，用户将公钥发布在网络上，这样其他用户就可以使用公钥对信息进行加密，只有拥有私钥的用户才可以正确解密密文。这实现了**保密通信**。  同时拥有私钥的人可以使用私钥对信息进行加密，所有拥有公钥的用户都可以解密密文，并且知道消息来源于拥有私钥的用户。这实现了**数字签名**。  **2、核心算法分析**  RSA算法的安全性是基于大整数因式分解的难度。目前最好的因素算法的时间复杂度是。  算法中辗转相除法的时间复杂度是O（log n）  计算逆元使用扩展欧几里得算法，时间复杂度是O（log n）  按模计算定理对信息加密和解密的时间复杂度是O（n） | | | | | | | | | |
| **四、实验及结果分析**  **1、实验**    **2、实验结果分析**  对于输入的内容，如测试用例9726，算法可以将明文转换成对应的密文，这里是3969使得信息在通信过程中得到加密，只有拥有密钥的人，才可以通过密钥解密密文，还原输入明文。可以看出算法输出的最后一行得到的明文和用户输入的明文一致。 | | | | | | | | | |
| **五、实验小结（包括问题和解决办法、心得体会、建议和意见等）**  通过本次实验，我了解了RSA算法对信息进行加密的具体过程，了解了在计算机硬件限制的情况下，如何使用按模计算原理等方法在有限计算范围内对较大数字进行计算。通过实验测试，了解了RSA算法在保密通信和数字签名上的实现原理。 | | | | | | | | | | |
| **六、指导教师评语** | | | | | | | | | | |
| 成 绩 |  | | 批阅人 |  | | | 日 期 |  | | |

**附件：**

#include <iostream>

using namespace std;

void swap(int& a, int& b) {

int c = a;

a = b;

b = c;

}

int gcd(int m, int n) {

if (m == 0) {

return n;

}

if (n == 0) {

return m;

}

if (m > n) {

swap(m, n);

}

while (m>0)

{

int c = n % m;

n = m;

m = c;

}

return n;

}

int choiceE(int f) {

int e = rand() % f + 1;

while (gcd(e,f)!=1)

{

e = rand() % f + 1;

}

return e;

}

int ext\_euclid(int a, int b, int f, int e){

int m, n, t;

if (e == 1) return b;

m = f / e;

n = f % e;

t = a - b \* m;

ext\_euclid(b, t, e, n);

}

void main()

{

//输入质数p和q

int p, q;

cout << "输入一个质数p(如101):";

cin >> p;

cout << "输入一个质数q(如113):";

cin >> q;//求得n=p\*q的值

int n = p \* q;

cout << "分组加密时，每个分组的大小不能超过n=p\*q="<<n<<endl;

//求得φ(n)=(p-1)\*(q-1)的值

int f = (p - 1) \* (q - 1);

cout << "模φ(n)=(p-1)\*(q-1)=" << f << endl << endl;

//选取与φ(n)互质的公钥e

int e = choiceE(f);

cout << "产生与φ(n)互质的公钥e:"<<e<<endl;

//由e和φ(n)生成私钥d

int d = ext\_euclid(0, 1, f, e);

while (d <= 0) d += f;

cout << "通过调用扩展欧几里德算法，求得密钥d为：" << d << endl;

//利用生成的公钥{e,n}对明文M进行加密

int M, C;

cout << "现在公钥{e,n}、私钥{d,n}均已生成完毕。\n\n请输入需要传输的明文内容进行加密(如9726)：";

cin >> M;

C = 1;

for (int i = 1; i <= e; i++)

{

C = C \* M % n; //使用按模计算公式

}

cout << "明文M=" << M << "经加密后得到密文C=M^e(mod n)：" << C << endl;

//利用生成的私钥私钥{e,n}对密文C进行解密

M = 1;

for (int i = 1; i <= d; i++)

{

M = M \* C % n;

}

cout << "密文C=" << C << "经解密后得到明文M=C^d(mod n)：" << M << endl;

}