实验报告

57119101 王晨阳

目录

实验原理

- 公钥(加密)：(e,n)
- 私钥(解密)：(d,n)

$n=p\cdot q$，p和q为两个大素数

$(e,\varphi(n))=1$

$e\cdot d\equiv 1\ (\mod \varphi(n))$

- **加密**
  $E(P)=C\equiv P^e\ (\mod n)$
- 准备好p,q，计算$n=p\cdot q,\varphi(n)$
  - 假设一个与$\varphi(n)$互质的e，求出d
  - 使用公钥加密信息m：$m^e\equiv c\ (\mod n)$
- **解密**
  $D(C)=C^d\equiv (P^e)^d\equiv P^{e\cdot d}$
  $\equiv P^{k\cdot\varphi(n)+1}\equiv \left(P^{\varphi(n)}\right)P\equiv P\ (\mod n)$
  - 求解$c^d\ (\mod n)$

编程实现

扩展欧几里得

求解贝祖等式

```
void ex_gcd(long long a,long long b,long long &x,long long &y)
{
    if (b==0)
    {
        x=1;
        y=0;
        return;
    }
    ex_gcd(b,a%b,x,y);
```

```
    long long t=x;
    x=y;
    y=t-a/b*y;
    return;
}
```

快速乘

防止乘法溢出`long long`

```
long long qMultiply(long long x,long long y,long long c)
{
    return (x*y-(long long)((long double)x/c*y)*c+c)%c;
}
```

快速幂

加快幂运算

为什么不用中国剩余定理：因为我能想到的算法复杂度更高

```
long long qPower(long long base,long long power,long long c)
{
    long long ans=1,res=base;
    while(power)
    {
        if(power&1)//power为奇数
          ans=qMultiply(ans,res,c)%c;
        res=qMultiply(res,res,c)%c;
        power>>=1;//power/2
    }
    return ans;
}
```

验证互质

```
bool is_coprime(long long a,long long b)
{
    if (a==1 || b==1)     // 两个正整数中，只有其中一个数值为1，两个正整数为互质数
        return true;
    while (true)
    {          // 求出两个正整数的最大公约数
        long long t=a%b;
        if (t==0) break;
        else
        {
            a=b;
            b=t;
        }
    }
    return (b==1);
}
```

RSA加密

```
long long RSA_encode(long long m,long long &n,long long &d,long long &e,long long &phi_n)
{
    long long temp;
    n=p*q;
    phi_n=n-p-q+1;
    srand(time(0));
    e=rand()%phi_n+1;
    while (!is_coprime(e,phi_n)) e=rand()%phi_n+1;
    ex_gcd(e,phi_n,d,temp);
    while (d<=0)
        d+=phi_n;
    return (qPower(m,e,n));
}
```

RSA解密

```
long long RSA_decode(long long n,long long d,long long c)
{
    return (qPower(c,d,n));
}
```

代码

```
/*************************************************************************
**********
author:王晨阳
description:RSA加密解密
*************************************************************************
*********/

#include<bits/stdc++.h>
using namespace std;

const long long p=10191161;
const long long q=10191133;

//扩展欧几里得
void ex_gcd(long long a,long long b,long long &x,long long &y)
{
    if (b==0)
    {
        x=1;
        y=0;
        return;
    }
    ex_gcd(b,a%b,x,y);
    long long t=x;
    x=y;
    y=t-a/b*y;
    return;
}

```

```cpp
//快速乘
long long qMultiply(long long x,long long y,long long c)
{
    return (x*y-(long long)((long double)x/c*y)*c+c)%c;
}

//快速幂
long long qPower(long long base,long long power,long long c)
{
    long long ans=1,res=base;
    while(power)
    {
        if(power&1)//power为奇数
          ans=qMultiply(ans,res,c)%c;
        res=qMultiply(res,res,c)%c;
        power>>=1;//power/2
    }
    return ans;
}

//验证互质
bool is_coprime(long long a,long long b)
{
    if (a==1 || b==1)     // 两个正整数中，只有其中一个数值为1，两个正整数为互质数
        return true;
    while (true)
    {          // 求出两个正整数的最大公约数
        long long t=a%b;
        if (t==0) break;
        else
        {
            a=b;
            b=t;
        }
    }
    return (b==1);
}

//RSA加密
long long RSA_encode(long long m,long long &n,long long &d,long long &e,long long &phi_n)
{
    long long temp;
    n=p*q;
    phi_n=n-p-q+1;
    srand(time(0));
    e=rand()%phi_n+1;
    while (!is_coprime(e,phi_n)) e=rand()%phi_n+1;
    ex_gcd(e,phi_n,d,temp);
    while (d<=0)
        d+=phi_n;
    return (qPower(m,e,n));
}

```

```
//RSA解密
long long RSA_decode(long long n,long long d,long long c)
{
    return (qPower(c,d,n));
}

//主程序
int main()
{
    long long m,n,d,e,phi_n,c,Message;
    printf("明文为 ");
    scanf("%lld",&m);
    c=RSA_encode(m,n,d,e,phi_n);
    printf("公钥为 (%lld,%lld)\n私钥为 (%lld,%lld)\n",e,n,d,n);
    printf("密文为 %lld\n",c);
    Message=RSA_decode(n,d,c);
    printf("明文为 %lld\n",Message);
}
```

实验结果

由上次作业得到两个素数分别为

```
p=10191161
q=10191133
```

程序运行结果为

```
明文为 57119101
公钥为 (6619,103859477175413)
私钥为 (96296341794739,103859477175413)
密文为 84116618419953
明文为 57119101
```

注：由于`e`为随机产生，每次运行的结果不完全相同！