**密码学原理与实践**

**实验-1-RSA算法与攻击实现**

**学院名称 智能与计算学部**

**专 业 网络空间安全**

**学生姓名 石子跃**

**学 号 3020244294**

**年 班 级 2020级1班**

**2022年 10 月 29 日**

# 实验问题-1 设计一个RSA加密算法

## 1.1实验原理





## 1.2 实验要求

要求：实现加解密功能。

加密

输入：p, q, e, m

输出：c

解密

输入：p, q, d, c

输出：m

(需要对p和q做素性测试；需要对e和d做参数检查；输入参数为unsigned long型，考虑计算中间结果溢出情况的为佳)

## 1.3关键技术的程序实现方法与代码：

#### 算法2 RSA公钥加密 及算法4 从左向右二进制算法：

1. //rsa加密函数
2. uli rsa\_encrypt(**struct** encrypt\_data data)
3. {
4. uli c = rsa\_modExp(data.m,data.e,data.p\*data.q);
6. **return** c;
7. }
9. //模函数 C=M^e mod n
10. uli rsa\_modExp(uli M, uli e, uli n)
11. {
12. uli res = 1;
13. **while**(e > 0) {
14. /\* Need long multiplication else this will overflow...
15. 必须使用长乘法，否则这将溢出\*/
16. **if**(e & 1) {
17. //res = (res \* M) % n;
18. res = mod2(res,M,n);
19. }
20. e = e >> 1;
21. //M = (M \* M) % n; /\* Same deal here \*/
22. M = mod2(M,M,n);
23. }
24. **return** res;
25. }

为了避免一些溢出问题，其中将模乘进行了算法优化，如下：

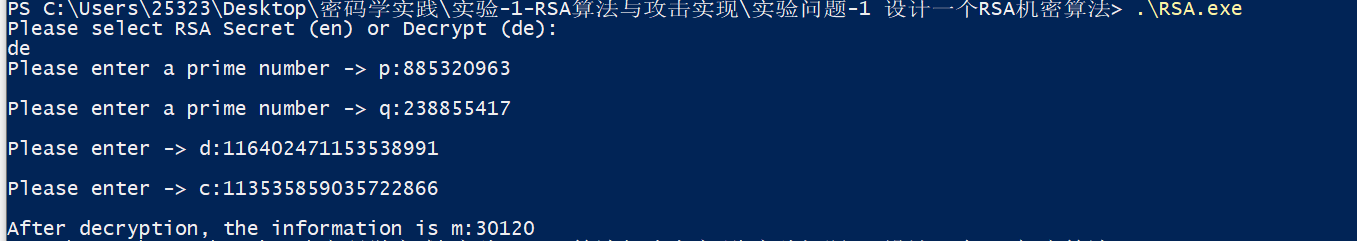
1. uli mod2(uli a, uli b, uli n){
2. uli big1[2]={0, 0};
3. uli d = 0;
4. uli tmp1 = 0, tmp2 = 0;
5. **int** i = 0;
6. **for**(i; i<64; i++){
7. tmp1 = a \* ((b>>(63-i))&1);
8. **if**((2\*d)<d){
9. uli tmp = d;
10. big1[1] = MAX\_64num - tmp;
11. big1[0] = MAX\_64num - n;
12. tmp = tmp - big1[1];
13. tmp = tmp + big1[0];
14. tmp2 = tmp;
15. }
16. **else**{
17. tmp2 = (2 \* d)%n;
18. }
19. **if**((tmp1+tmp2)<tmp1){
20. big1[1] = MAX\_64num - tmp2;
21. big1[0] = MAX\_64num - n;
22. d = tmp1 - big1[1];
23. d = d + big1[0];
24. }
25. **else**{
26. d = (tmp1+tmp2)%n;
27. }
28. }
29. **return** d;
30. }

#### 算法3 Fermat素性测试：

1. //Fermat素性测试
2. **bool** Fermatpritest(uli n,uli ref){
3. srand((unsigned)time(NULL));
4. uli i;
5. **for**(i=0;i<ref;i++){
6. uli a = (rand() % (n-2-2+1)) + 2;//随机选取整数a,2<=a<=n-2
7. uli r = (uli)rsa\_modExp(a,n-1,n);
8. **if**(r!=1){
9. cout<<"---"<<n<<"是合数,不符合规范,请重新进入程序"<<endl;
10. **return** 0;
11. }
12. }
13. cout<<"---"<<n<<"是素数,符合规范,继续执行"<<endl;
14. **return** 1;
15. }

## 1.4数据测试结果





# 实验问题-2 设计一个RSA小加密指数e的攻击算法

## 2.1 实验原理





## 2.2 实验要求

要求：给出RSA的公开参数和密文，可以恢复出群发明文。

输入：e=3, n1, n2, n3, c1, c2, c3

输出：m

(需要对n1、 n2、 n3做互素测试；输入参数为unsigned long型，考虑计算中间结果溢出情况的为佳)

## 2.3 关键技术的程序实现方法与代码

其实就是中国剩余定理的使用

1. **inline** ll mod\_ADD(ll a, ll b, ll m)
2. {
3. ll res=0;
4. **while**(b>0)
5. {
6. **if**(b&1)res=(res+a)%m;
7. a=(a+a)%m;
8. b>>=1;
9. }
10. **return** res;
11. }
12. **inline** **void** exgcd(ll a,ll b,ll &x,ll &y)
13. {
14. **if**(b==0)
15. {
16. x=1,y=0;
17. **return**;
18. }
19. exgcd(b,a%b,x,y);
20. **int** tmp=x;
21. x=y,y=tmp-a/b\*y;
22. **return**;
23. }
24. ll China()
25. {
26. ll ans=0,m=1;
27. **for**(ll i=1;i<=k;i++)
28. m\*=n[i];
29. ll x,y;
30. **for**(ll i=1;i<=k;i++)
31. {
32. ll p=m/n[i];
33. exgcd(p,n[i],x,y);
34. x=(x%n[i]+n[i])%n[i];
35. ans=(ans+mod\_ADD(mod\_ADD(p,x,m),c[i],m))%m;
36. }
37. **return** (ans+m)%m;
38. }

## 1.4数据测试结果

