上海理工大学光电信息与计算机工程学院

《信息安全》实验报告

专　　业　计算机科学与技术

学生姓名　许耀永

学　　号　1712480131

年　　级　2017 级

指导教师　刘亚

成　　绩：

教师签字：

报告格式要求

1、 正文字体中文为宋体，五号，行距为固定值 18 磅，西文为 Times New Rome, 五号，行距为固定值 18 磅。

2、 章节标题为加粗宋体，小四号，段前段后各 0.5 行，行距为固定值 18 磅。

3、 打印时需双面打印。

RSA 非对称加密实验

一、 实验目的

实现 RSA 的加密(Encryption)和解密(Decryption)

二、RSA 基础

Operation:

1. Key Generation:

(1) Select two large prime numbers p and q randomly, such that p is not equal to q

(2) Compute n such that n = p * q

(3) Compute f(n) = f(p) * f(q) = (p - 1) * (q - 1) where f is Euler's totient function

(4) Select an number **e** such that 1 < **e** < f(n) and gcd (e, f(n)) == 1

where e and f(n) are co-prime

(5) Compute d as the multiplicative inverse of e(mod(f(n))) i.e. de = 1 mod f(n)

(6) Publish the pair PU = (e, n) as participant's public key

Keep the pair PR = (d, n) as the participant's private key

```
Algorithm 1 : findE(phi_n)
    Begin
        e <- 0
        do
            begin
                choose an integer number e
                (e must be co-prime of phi_n)
                while(!checkCoprime(phi_n, e))
        end do-while
        return e
    End
```

```
Algorithm2 : FindD(phi_n,e)
        Begin
            Local variables: a, b , x , y , u , v , m , n , q , r , gcd
            a <- phi_n
            b <- e
            x <- 0
            y <- 1
            u <- 1
            v <- 0
            gcd <- b
            while(a != 0)
                begin
                    q <- gcd / a
                    r <- gcd % a
                    m <- x – u * q
                    n <- y – v * q
                    gcd <- a
                    a <- r
                    x <- u
                    y <- v
                    u <- m
                    v <- n
            end while
            if y < 1
                begin
                    y <- phi_n + y
            end if
        return y
    End
```

Algorithm 3 : Generate(&n,&e,&d)

Begin

local variables : p , q , phi_n

Enter two prime numbers and stored them in p and q respective

n <- multiply(p , q)

phi_n <- multiply(p – 1 , q - 1)

e <- findE(phi_n)

d <- findD(phi_n,e)

(e,n) => public key

(d,n) => private key

End

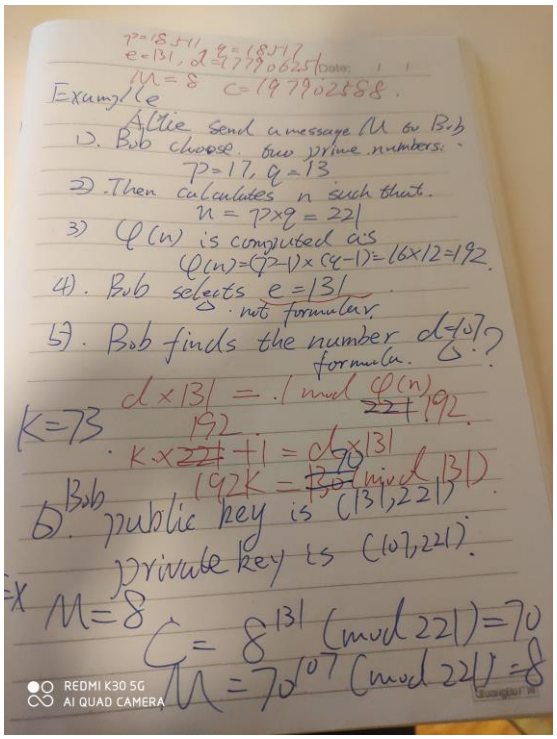
2. Encryption

C = M ^ e (mod n)

3. Decryption

M = C ^ d (mod n)

Example



三、RSA 项目代码

//RSA.h

```
#include<cstdio>
#include<cstdlib>
#include<ctime>
#include<cmath>
#include<cstring>

typedef long long LL;

const int MAX_ROW = 50;

//KEY GENERATION
LL getPrime(){
	bool ifPrime = false;
	LL a = 0;
	int arr[MAX_ROW];

	for(int i = 0; i < MAX_ROW;++i){
		arr[i] = i + 3;
	}

	while(!ifPrime){
		srand((unsigned)time(0));
		ifPrime = true;
		a = (rand() % 1000) * 2 + 3;
```

```
            for(int j = 0;j < MAX_ROW;j++){
                if(a % arr[j] == 0){
                    ifPrime = false;
                    break;
                }
            }
        }

        return a;
}

bool checkIsPrime(LL n){
        if(n < 2) return false;

        LL i = 2;

        while(i <= n /2){
            if(!(n % i)) return false;
            i++;
        }

        return true;
}

bool checkCoPrime(LL n1,LL n2){
        LL lowest;
```

```
    if(n1 > n2) lowest = n1;
    else lowest = n2;

    LL i = 2;
    bool coPrime = true;

    while(i < lowest){
        if(!(n1 % i) && !(n2 % i)) coPrime = false;
        i++;
    }

    return coPrime;
}

LL findE(LL phi_n){
    LL e = 0;

    do{
        printf("Choose an integer number e\n");
        scanf("%lld",&e);
    }while(!checkCoPrime(phi_n,e));

    return e;
}

LL findD(LL phi_n,LL e){
    LL a = phi_n;
```

```
    LL b = e;
    LL x = 0,y = 1;
    LL u = 1,v = 0;
    LL gcd = b;
    LL m,n,q,r;

    while(a != 0){
        q = gcd / a;
        r = gcd % a;
        m = x - u * q;
        n = y - v * q;

        gcd = a;
        a = r;
        x = u;
        y = v;
        u = m;
        v = n;
    }

    if(y < 1){
        y = phi_n + y;
    }

    return y;
}
```

```
LL multiply(LL n1,LL n2){
	return n1 * n2;
}

int primeMenu(){
	int choice = 0;
	do{
		printf("------pq------\n");
		printf("1:auto\n");
		printf("2:myself\n");
		printf("choose:\n");
		scanf("%d",&choice);
		printf("--------------\n");
	}while(choice != 1 && choice != 2);
	return choice;
}

void autoPrime(LL& p,LL& q){
	printf("auto\n");
	do{
		p = getPrime();
		q = getPrime();
	}while(p == q);
	printf("p:%lld q:%lld\n",p,q);
}
```

```
void myselfPrime(LL& p,LL& q){
	printf("myself\n");
	do{
		scanf("%lld",&p);
	}while(!checkIsPrime(p));

	do{
		scanf("%lld",&q);
	}while(!checkIsPrime(q));
	printf("p:%lld q:%lld\n",p,q);
}

void generateKey(LL& n,LL& e,LL& d){
	LL p,q,phi_n;

	int choice = primeMenu();
	switch(choice){
		case 1 :
			autoPrime(p,q);
			break;
		case 2 :
			myselfPrime(p,q);
			break;
	}

	n = multiply(p,q);
```

```
	printf("n:%lld\n",n);

	phi_n = multiply(p-1,q-1);
	printf("phi_n:%lld\n",phi_n);

	e = findE(phi_n);
	printf("e:%lld\n",e);

	d = findD(phi_n,e);
	printf("d:%lld\n",d);
}

//ENCRYPT DECRYPT
LL encDec(LL m,LL ed,LL n){
	LL rem;
	LL x = 1;

	while(ed != 0){
		rem = ed % 2;
		ed = ed / 2;

		if(rem == 1) x = (x * m) % n;

		m = (m * m) % n;
	}

	return x;
```

```
}

void encDecNum(LL ed,LL n){
	LL m;

	printf("Enter a message number:\n");
	scanf("%lld",&m);

	LL c = encDec(m,ed,n);
	printf("M/C:%lld C/M:%lld\n",m,c);
}
```

//main.cpp

```
#include "RSA.h"

int main(){
	LL n,e,d;
	generateKey(n,e,d);

	encDecNum(e,n);

	return 0;
}
```

四、RSA 运行截图

```
myself
17
13
p:17 q:13
n:221
phi_n:192
Choose an integer number e
131
e:131
d:107
Enter a message number:
8
M/C:8 C/M:70
```

```
auto
p:541 q:547
n:295927
phi_n:294840
Choose an integer number e
131
e:131
d:36011
Enter a message number:
8
M/C:8 C/M:197829
```

```
myself
541
547
p:541 q:547
n:295927
phi_n:294840
Choose an integer number e
36011
e:36011
d:131
Enter a message number:
197829
M/C:197829 C/M:8
```