

Cryptography And Network Security Lab

Assignment submission

PRN No: 2019BTECS00017

Full name: Muskan Raju Attar

Batch: B5

Assignment: 12

Title of assignment: Implementation of RSA (Rivest–Shamir–Adleman)

Title:

Implementation of RSA (Rivest–Shamir–Adleman)

Aim:

To develop and implement the RSA (Rivest–Shamir–Adleman)

Theory:

- RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem that is widely used for secure data transmission. It is also one of the oldest.
- The acronym "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977. An equivalent system was developed secretly in 1973 at GCHQ (the British signals intelligence agency) by the English mathematician Clifford Cocks.
- An RSA user creates and publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers are kept secret. Messages can be encrypted by anyone, via the public key, but can only be decoded by someone who knows the prime numbers.

- The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem".

Implementation of RSA (Rivest–Shamir–Adleman)

Code:

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

int modu(int b, unsigned int exp, unsigned int m)
{
    int x = 1;
    int i;
    int power = b % m;

    for (i = 0; i < sizeof(int) * 8; i++) {
        int least_bit = 0x00000001 & (exp >> i);
        if (least_bit)
            x = (x * power) % m;
        power = (power * power) % m;
    } return x;
}

int modl(int a, int m)
{
    int temp = m;
    int y = 0, x = 1;
    if (m == 1)
        return 0;
    while (a > 1)
    {
        int q = a / m;
        int t = m;
        m = a % m, a = t;
    }
```

```

        t = y;
        y = x - q * y;
        x = t;
    } if (x < 0)
        x += temp;
    return x;
}

```

```

int gcd(int a, int b)
{

```

```

    if (a == 0 || b == 0)
        return 0;

```

```

    if (a == b)
        return a;

```

```

    if (a > b)
        return gcd(a-b, b);
    return gcd(a, b-a);
}

```

```

int Prime(int num){
    int flag = 1;
    for(int i=2;i<=sqrt(num);i++)
    {
        if(num%i==0)
        {
            flag = 0;
            return flag;
        }
    }
    return flag;
}

```

```
}
```

```
int lcm(int a, int b)
{
    return (a*b)/gcd(a, b);
}
```

```
int main(){
    int msg; char m;
    cout<<"\n Enter the character to be encrypted: ";
    cin>>m;
    msg = (int)m;
    cout<<"\n The corresponding ASCII value of the character is"<<msg;
    int p,q, random; int i=0; int a[2];
    srand (time(NULL));
    generate:
    while(i<2){
        random = rand() % 40 + 3;
        if(Prime(random)){
            a[i]=random;
            i++;
        }
    }
    i=0;p=a[0];q=a[1];
    if(p==q){
        goto generate;
    }
```

```
    cout<<"\n The Random Prime Numbers are: "<<p<<" and "<<q;
    int n; n = p*q;
    int phi = (p-1)*(q-1);
    int lambda = lcm(p-1,q-1);
    int e;
    vector<int> tot;
    for(int i=3;i<lambda;i++)
```

```

{ if(gcd(i,lambda) == 1){
    tot.push_back(i);
}
}
int size = tot.size();
int ran = rand() % size;
e = tot[ran];

cout<<"\n The modulus is: "<<n;
cout<<"\n The phi(n) is: "<<phi;
cout<<"\n The lambda(n) is:"<<lambda;
cout<<"\n The toitent is: "<<e;
cout<<"\n The public key is: ("<<n<<","<<e<<");
long long int encrypt;
encrypt = modu(msg,e,n);
cout<<"\n The Cipher text is: "<<(char)encrypt;
cout<<"\n The ASCII value of Cipher Text is: "<<encrypt;
long long int d = modl(e,lambda);
if(d==e){
    cout<<"\n";
    goto generate;
}
cout<<"\n The private key is: "<<d;
long long int decrypted;
decrypted = modu(encrypt,d,n);
cout<<"\n The Decrypted/Plain text is : "<<(char)decrypted;
cout<<"\n The ASCII value of Plain text is: "<<decrypted;
return 0;
}

```

Output:

C:\Users\Muskan Raju Attar\Desktop\CNS Assignment new\Experiment - 12\RSA.exe

```
Enter the character to be encrypted: B

The corresponding ASCII value of the character is 66
The Random Prime Numbers are: 41 and 11
The modulus is: 451
The phi(n) is: 400
The lambda(n) is: 40
The totient is: 21
The public key is: (451,21)
The Cipher text is: B
The ASCII value of Cipher Text is: 66

The Random Prime Numbers are: 13 and 29
The modulus is: 377
The phi(n) is: 336
The lambda(n) is: 84
The totient is: 23
The public key is: (377,23)
The Cipher text is: B
The ASCII value of Cipher Text is: 14
The private key is: 11
The Decrypted/Plain text is : B
The ASCII value of Plain text is: 66
-----
Process exited after 1.452 seconds with return value 0
Press any key to continue . . .
```

Conclusion:

The RSA can be used to do data transmission