Maitreya Kanitkar

BE-IT 8084

ICS Assignment 2: RSA algorithm

Title: Write a program in C++ or JAVA to implement the RSA algorithm for key generation and Cipher verification.

Objective: To study,
1. Public key algorithm.
2. RSA algorithm
3. Concept of Public key and Private Key.

Theory:

Public Key Algorithm:
Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristics:
It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.
In addition, some algorithms, such as RSA, also exhibit the following characteristics:
Either of the two related keys can be used for encryption, with the other used for decryption.
Public key encryption scheme has six ingredients:
Plaintext: This is a readable message or data that is fed into the algorithm as input.
Encryption algorithm: The encryption algorithm performs various transformations on plaintext.
Public and private key: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different cypher texts.
Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.
The essential steps are as the following:
1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or the other accessible file. This is the public key. The companion key is kept private. As the figure suggests, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

The RSA Algorithm:
The scheme developed by Rivest, Shamir and Adleman make use of an expression with exponentials. The plaintext is encrypted in blocks, with each block having a binary value less than some number n. That is the block size must be less than or equal to log2 (n); in practice, the
block size is I bits, where 2i<n<=2i+1. Encryption and decryption are of the following form, for some plaintext block M and cypher text block C:

C = (M^e)mod n
M = (C^d)mod n = ((M^e)^d)mod n = (M^(ed))mod n

Both sender and receiver must know the value of n. The sender knows the value of e,
and only the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of PU = {e, n} and a private key of PR = {d, n}. For this algorithm to be satisfactory for public-key encryption, the following requirements must meet:

1. It is possible to find values of e, d, n such that Med mod n = M for all M<n.
2. It is relatively easy to calculate Me mod n and Cd mod n for all values of M<n.
3. It is feasible to determine d given e and n.

Example:
1. Select two prime numbers, p = 17 and q = 11.
2. Calculate n = pq = 17*11 = 187.
3. Calculate Ø (n) = (p-1)(q-1) = 16*10 = 160.
4. Select e such that relatively prime to Ø (n) = 160 and less than Ø (n); we choose e = 7.
5. Determine d such that de ≡ 1 (mod 160) and d < 160. The correct value is d = 23, because 23*7 = 161 = 10*16+1; d can be calculated using the extended Euclid's algorithm.
The resulting keys are public-key PU = {7, 187} and private key PR = {23, 187}. The example shows the use of these keys for plaintext input of M=88.

Advantages:
1. Easy to implement.

Disadvantages:
1. Anyone can announce the public key.

Algorithm:
1. Start
2. Input two prime numbers p and q.
3. Calculate n=pq.
4. Calculate phi=(p-1)(q-1).

5. Calculate the value of e.
6. Determine d.
7. Determine PU and PR.
8. Take input plaintext.
9. Encrypt the plaintext and show the output.
10. Stop.

Conclusion: We have studied and implemented the RSA - Public key algorithm.

Program:

```
#include<iostream>
using namespace std;

class rsa
{
    int p, q;
    double n, phi, e=2, g, d;
    long msg, encrypted, decrypted, enc;

    public:
    void pipeline()
    {
        input();
        algorithm();
        output();
    }

    void input()
    {
        cout<<"Enter the value of p : ";
        cin>>p;

        cout<<"Enter the value of q : ";
        cin>>q;

        cout<<"Enter the message : ";
        cin>>msg;
```

```
}

int gcd(int a, int b)
{
    if(a==0)
    return b;
    return gcd(b%a, a);
}

int mod(int m, int k, int n)
{
    int result=1;

    m=m%n;

    if(m==0)
    return 0;

    while(k>0)
    {
        if(k%2==1)
        result=(result*m)%n;

        k/=2;
        m=(m*m)%n;
    }
    return result;
}

void algorithm()
{
    n=p*q;
    phi=(p-1)*(q-1);

    while(e<phi)
    {
        if((gcd(e, phi)==1))
        break;

        else
```

```
            e++;
        }

        for(int i=1;i<phi;i++)
        if(((int)e*i)%(int)phi==1)
        d=i;

        encrypted=mod(msg, e, n);
        decrypted=mod(encrypted, d, n);
    }

    void output()
    {
        cout<<endl<<"Original message : "<<msg;
        cout<<endl<<"Encrypted message : "<<encrypted;
        cout<<endl<<"Decrypted message : "<<decrypted;
    }
};

int main()
{
    rsa obj;
    obj.pipeline();
    return 0;
}



/*

OUTPUT 1:-

Enter the value of p : 3
Enter the value of q : 7
Enter the message : 12

Original message : 12
Encrypted message : 3
Decrypted message : 12
```

```
OUTPUT 2:-

Enter the value of p : 91
Enter the value of q : 97
Enter the message : 1000

Original message : 1000
Encrypted message : 5095
Decrypted message : 1000

*/
```