Practical: 8

Aim: Implement RSA encryption-decryption algorithm.

- RSA Algorithm is used to encrypt and decrypt data in modern computer systems and other electronic devices. RSA algorithm is an asymmetric cryptographic algorithm as it creates 2 different keys for the purpose of encryption and decryption.

- RSA makes use of prime numbers (arbitrary large numbers) to function. The public key is made available publicly (means to everyone) and only the person having the private key with them can decrypt the original message.

❖ Working of RSA Algorithm:

- RSA involves use of public and private key for its operation. The keys are generated using the following steps:-

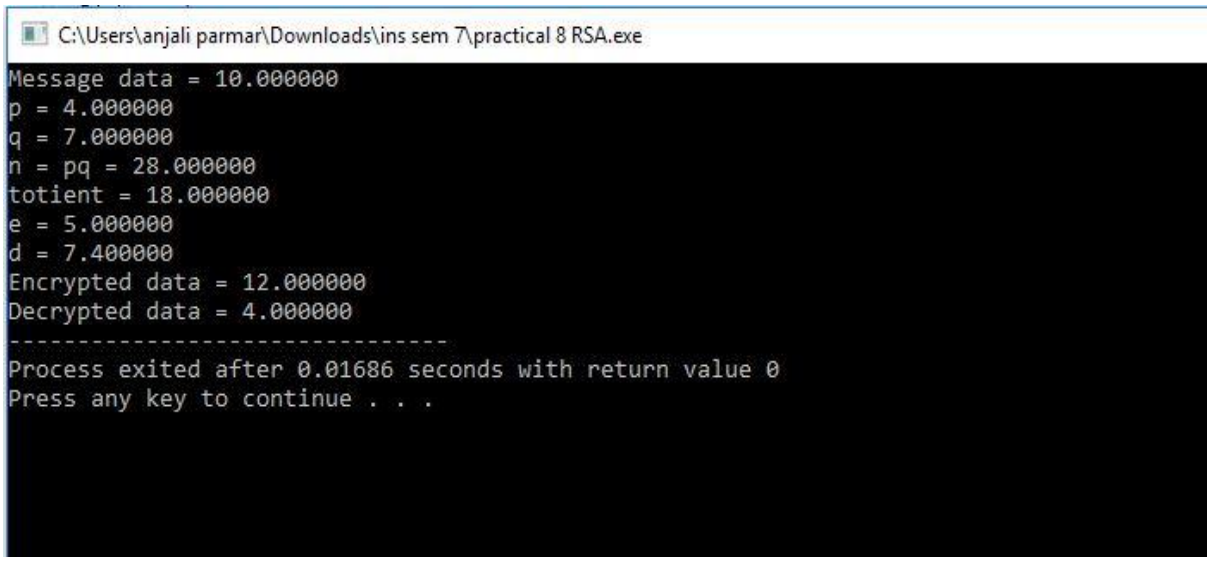
1. Two prime numbers are selected as **p** and **q**
2. **n = pq** which is the modulus of both the keys.
3. Calculate **∅(n) = (p-1)(q-1)**
4. Choose **e** such that **e > 1** and coprime to **∅(n)** which means **gcd (e, ∅(n))** must be equal to **1, e** is the public key
5. Choose **d** such that it satisfies the equation **de = 1 + k (∅(n)), d** is the private key not known to everyone.
6. Cipher text is calculated using the equation **c = m^e mod n** where m is the message.
7. With the help of **c** and **d** we decrypt message using equation **m = c^d mod n** where d is the private key.

❖ program:

```
#include<stdio.h>
#include<math.h>

int gcd(int a, int h)
{
   int temp;
   while(1)
   {
      temp = a%h;
      if(temp==0)
      return h;
      a = h;
      h = temp;
   }
}
```

```
int main()
{
    double p = 4;
    double q = 7;
    double n=p*q;
    double count;
    double totient = (p-1)*(q-1);

     double e=2;

    while(e<totient){
    count = gcd(e,totient);
    if(count==1)
        break;
    else
        e++;
    }

    double d;

    double k = 2;

    d = (1 + (k*totient))/e;
    double msg = 12;
    double c = pow(msg,e);
    double m = pow(c,d);
    c=fmod(c,n);
    m=fmod(m,n);

    printf("Message data = %lf",msg);
    printf("\np = %lf",p);
    printf("\nq = %lf",q);
    printf("\nn = pq = %lf",n);
    printf("\ntotient = %lf",totient);
    printf("\ne = %lf",e);
    printf("\nd = %lf",d);
    printf("\nEncrypted data = %lf",c);
    printf("\nDecrypted data = %lf",m);

    return 0;
}
```

Output:



C:\Users\anjali parmar\Downloads\ins sem 7\practical 8 RSA.exe

```
Message data = 10.000000
p = 4.000000
q = 7.000000
n = pq = 28.000000
totient = 18.000000
e = 5.000000
d = 7.400000
Encrypted data = 12.000000
Decrypted data = 4.000000
--------------------------------
Process exited after 0.01686 seconds with return value 0
Press any key to continue . . .
```