Experiment 5

Shashwat Shah
60004220126
TY BTech Comps B

Aim: Study and implement RSA algorithm.

Theory: RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e public key and private key. As the name describes, the public key is given to everyone and the private key is kept private.

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private keys are also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponential. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024 bit keys could be broken in the near future. But till now it seems to be an infeasible task.

Public key generation - $p = 53$ and $q = 59$

$\therefore n = p \times q = 3127$

$\therefore \phi(n)$ needs to be found

we need e

$1 < e < \phi(n)$

Private Key generation

$\phi(n) = (p-1)(q-1)$

$\phi(n) = 3016$

$\therefore$

$d = (k * \phi(n)) + 1) / e$

$k = 2$

$\therefore$

$d = 2011$

Encryption - $H = 8$ $I = 9$

$\therefore$

$C = 89^e \bmod n$

$= 1394$

Decryption

$= cd \bmod n$

$\therefore = 89$

$8 = H$ & $9 = I$

$\therefore$ 'HI'

Conclusion : With the increase in amount of data being generated, it is very important that confidential information does not get leaked and is red by the ~~inted~~ intended recipient. We learnt about asymmetric key ciphers and the RSA algorithm.

EXPERIMENT 5

Shashwat Shah
TYBtech Comps B
C22
60004220126

AIM: Study and Implement RSA Algorithm.

CODE:

```
import math
def enc(plain,e,n):
    return (plain**e)%n
def dec(cipher,d,n):
    return (cipher**d)%n
def get_public_key(phi):
    e = 2
    while e < phi:
        if math.gcd(e,phi) == 1:
            break
        else:
            e += 1
    return e
def get_private_key(e,phi):
    d = 2
    while d < phi:
        if (d*e)%phi == 1:
            break
        else:
            d += 1
    return d
if __name__=='__main__':
    p,q = input('Enter two prime numbers: ').split()
    plain = int(input('Enter the plain text: '))
    p,q = int(p),int(q)
    n = p*q
    phi = (p-1)*(q-1)
    e = get_public_key(phi)
    d = get_private_key(e,phi)
    print('Public key(e,n): ',e,n)
    print('Private key(d,n): ',d,n)
    cipher = enc(plain,e,n)
    print('Cipher text: ',cipher)
    print('Plain text: ',dec(cipher,d,n))
```

OUTPUT:

```
uments/BTech/Docs/6th Sem/IS/Code/Exp5/RSA.py"
Enter two prime numbers: 1291 607
Enter the plain text: 909
Public key(e,n):  7 783637
Private key(d,n):  670063 783637
Cipher text:  359730
Plain text:  909
```