

CSS Experiment 3

Deep Kothari
TE COMP B
Roll no 23

Aim:

To Analyze and implement RSA cryptosystem and Digital signature scheme using RSA/El Gamal.

Theory:

RSA Algorithm:

RSA algorithm is a public key encryption technique and is considered as the most secure way of encryption. It was invented by Rivest, Shamir and Adleman in year 1978 and hence name RSA algorithm.

The RSA algorithm holds the following features –

- RSA algorithm is a popular exponentiation in a finite field over integers including prime numbers.
- The integers used by this method are sufficiently large making it difficult to solve.
- There are two sets of keys in this algorithm: private key and public key.

RSA algorithm uses the following procedure to generate public and private keys:

- Select two large prime numbers, p and q .
- Multiply these numbers to find $n = p \times q$, where n is called the modulus for encryption and decryption.
- Choose a number e less than n , such that n is relatively prime to $(p - 1) \times (q - 1)$. It means that e and $(p - 1) \times (q - 1)$ have no common factor except 1. Choose "e" such that $1 < e < \phi(n)$, e is prime to $\phi(n)$,
 $\text{gcd}(e, \phi(n)) = 1$
- If $n = p \times q$, then the public key is $\langle e, n \rangle$. A plaintext message m is encrypted using public key $\langle e, n \rangle$. To find ciphertext from the plain text following formula is used to get ciphertext C .
 $C = m^e \bmod n$
Here, m must be less than n . A larger message ($>n$) is treated as a concatenation of messages, each of which is encrypted separately.
- To determine the private key, we use the following formula to calculate the d such that:
 $D_e \bmod \{(p - 1) \times (q - 1)\} = 1$
Or
 $D_e \bmod \phi(n) = 1$
- The private key is $\langle d, n \rangle$. A ciphertext message c is decrypted using private key $\langle d, n \rangle$. To calculate plain text m from the ciphertext c following formula is used to get plain text m .
 $m = c^d \bmod n$

Example:

This example shows how we can encrypt plaintext 9 using the RSA public-key encryption algorithm. This example uses prime numbers 7 and 11 to generate the public and private keys.

Explanation:

Step 1: Select two large prime numbers, p , and q .

$$p = 7$$

$$q = 11$$

Step 2: Multiply these numbers to find $n = p \times q$, where n is called the modulus for encryption and decryption.

First, we calculate

$$n = p \times q$$

$$n = 7 \times 11$$

$$n = 77$$

Step 3: Choose a number e less than n , such that n is relatively prime to $(p - 1) \times (q - 1)$. It means that e and $(p - 1) \times (q - 1)$ have no common factor except 1. Choose " e " such that $1 < e < \phi(n)$, e is prime to $\phi(n)$, $\gcd(e, \phi(n)) = 1$.

Second, we calculate

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

$$\phi(n) = (7 - 1) \times (11 - 1)$$

$$\phi(n) = 6 \times 10$$

$$\phi(n) = 60$$

Let us now choose relative prime e of 60 as 7.

Thus, the public key is $\langle e, n \rangle = (7, 77)$

Step 4: A plaintext message m is encrypted using public key $\langle e, n \rangle$. To find ciphertext from the plain text following formula is used to get ciphertext C .

To find ciphertext from the plain text following formula is used to get ciphertext C .

$$C = m^e \bmod n$$

$$C = 9^7 \bmod 77$$

$$C = 37$$

Step 5: The private key is $\langle d, n \rangle$. To determine the private key, we use the following formula d such that:

$$D_e \bmod \{(p - 1) \times (q - 1)\} = 1$$

$$7d \bmod 60 = 1, \text{ which gives } d = 43$$

The private key is $\langle d, n \rangle = (43, 77)$

Step 6: A ciphertext message c is decrypted using private key $\langle d, n \rangle$. To calculate plain text m from the ciphertext c following formula is used to get plain text m .

$$m = c^d \bmod n$$

$$m = 37^{43} \bmod 77$$

$$m = 9$$

In this example, Plain text = 9 and the ciphertext = 37

Program for RSA Algorithm:

```
from decimal import Decimal
def gcd (a, b):
    temp = 0
    while (1):
        temp = a % b
        if temp == 0:
            return b
        a = b
        b = temp

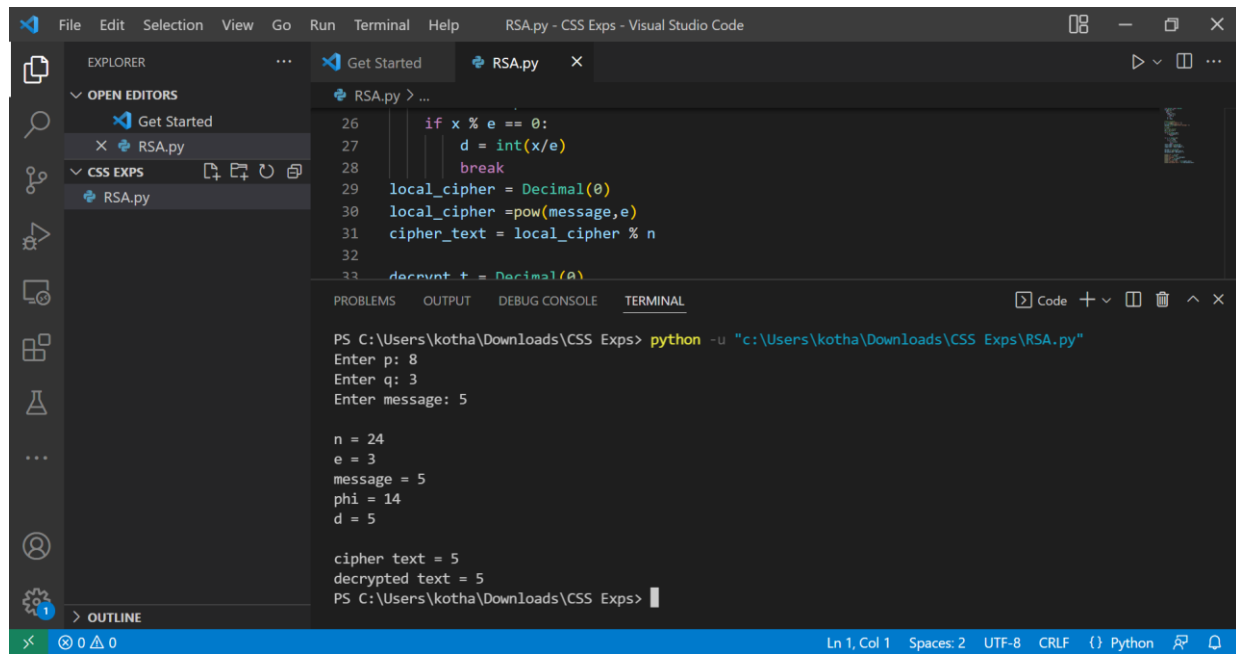
#input variables
p = int(input("Enter p: "))
q = int(input("Enter q: "))
message = int(input("Enter message: "))
#calculate n
n = p*q
#calculate totient
phi = (p-1)*(q-1)
#calculate e
for e in range(2,phi):
    if gcd(e,phi)== 1:
        break

for i in range(1,10):
    x = 1 + i*phi
    if x % e == 0:
        d = int(x/e)
        break
local_cipher = Decimal(0)
local_cipher =pow(message,e)
cipher_text = local_cipher % n

decrypt_t = Decimal(0)
decrypt_t= pow(cipher_text,d)
decrypted_text = decrypt_t % n

print('\nn = '+str(n))
print('e = '+str(e))
print('message = '+str(message))
print('phi = '+str(phi))
print('d = '+str(d))
print('\ncipher text = '+str(cipher_text))
print('decrypted text = '+str(decrypted_text))
```

Output for RSA Algorithm:



```
26     if x % e == 0:
27         d = int(x/e)
28         break
29 local_cipher = Decimal(0)
30 local_cipher = pow(message,e)
31 cipher_text = local_cipher % n
32
33 decrypt_t = Decimal(0)
```

```
PS C:\Users\kotha\Downloads\CSS Exps> python -u "c:\Users\kotha\Downloads\CSS Exps\RSA.py"
Enter p: 8
Enter q: 3
Enter message: 5

n = 24
e = 3
message = 5
phi = 14
d = 5

cipher text = 5
decrypted text = 5
PS C:\Users\kotha\Downloads\CSS Exps>
```

Conclusion:

Thus, we have seen how RSA algorithm works by using both public and the private key. We have successfully designed and implemented a code for RSA algorithm in python programming language and understood the concept of this asymmetric algorithm.