Experiment No. 4

Title: Digital Signature using RSA

Batch: B2 **Roll No.: 16010420117** **Experiment No.: 4**

Aim: To implement digital signature using RSA

Resources needed: Windows/Linux

Theory: Pre Lab/ Prior Concepts:

RSA is a public key algorithm named after its inventers Rivest, Shamir and Adleman. The characteristics of public key cryptography (which is also called as Asymmetric Cryptography) are as below:

- It has two keys. One key is called as private key and the other one is called as public key. Everyone who uses this cryptography has to have two keys each.
- Keys used for encryption and decryption should be different. If one is used for encryption, then other must be used for decryption. Any key can be used for encryption and then remaining key can be used for decryption.
- Public Key Cryptography is based on solid foundation of mathematics.
- It has large computational overheads. Hence ciphertext generated is of much larger size than the plaintext. Hence it is normally used for encrypting small size of data blocks. For example, passwords, symmetric keys, etc. It is not preferred to encrypt large files.
- RSA algorithm gets its security from the fact that it is extremely difficult to factorize large prime number.
- Security of the algorithm depends on the size of the key. Greater the size of the key, larger is the security. The key length is variable. The most commonly used key length is 512 bits.

Procedure / Approach /Algorithm / Activity Diagram:

A. Key generation Algorithm:

1. Choose large prime numbers p and q.
2. Calculate product n= pq. The value of n can be revealed publicly, but n is large enough that even supercomputers cannot factor it in a reasonable amount of time (years or even centuries).
3. Calculate phi = (p-1)(q-1)
4. Select e < phi such that it is relatively prime to phi. The public key is (e, n).
5. Determine d such that ed = 1 mod phi. The private key is (d, n). d, p ,q and phi are kept secret, only (e,n) is made public.

B. Digital signature generation:

6. Suppose M is the message.
7. Encrypt this message using private key of the sender to generate digital signature, C = M^d mod n.

C. Digital signature verification:

8. Decrypt the C generated in step 7 to verify the signature using public key of the sender, M = C^e mod n.

.

Results: (Program printout with output / Document printout as per the format)

```
import random
import math

class RSA:

        def _init_(self):self.p=0
                self.q=0 self.n=0
                self.e=0 self.d=0
                print("Digital Signature RSA!")

        def KeyGen(self):
                print("Enter only PRIME NUMBERS:") self.p
                =int(input("Enter p value: ")) self.q = int(input("Enter
                q value: "))while not self.prime(self.p):
                        print("Enter prime numbers only ")
                        self.p = int(input("Enter p value again: "))while not
                self.prime(self.q):
                        print("Enter prime numbers ")
                        self.q =int(input("Enter q value again: "))self.n = self.p*self.q
                print("*****")
                print("Value of n: ",self.n)
                #PHI Function
                self.phi =(self.p-1)*(self.q-1)print("PHI of
                n: ",self.phi)
                #E calculation
                self.e = random.randint(1,self.phi)
                while not math.gcd(self.e,self.phi) == 1:self.e =
                        random.randint(1,self.phi)
                print("Value of e: ",self.e)self.d=0
                for i in range(1,self.phi-1):
                        if (1== (self.e*i)%self.phi ):self.d=i
                print("Value of d: ",self.d)
```

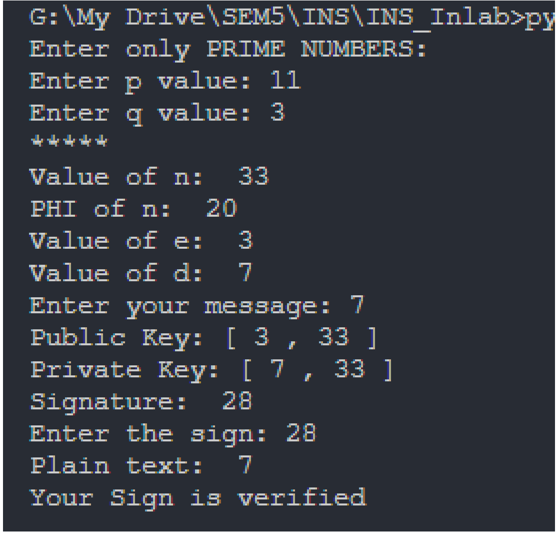
```
def prime(self,num):
        for i in range(2,num):if
            num%i==0:
                return False
        return True

    def CreateSignature(self,msg):
        print("Public Key: [",self.e,",",self.n,"]")
        print("Private Key: [",self.d, "," ,self.n,"]")cipher   =
        (msg**self.d)%self.n print("Signature: ",cipher)
        return cipher

    def VerifySignature(self):
        cipher = int(input("Enter the sign: "))msg =
        (cipher**self.e)%self.n print("Plain text: ",msg)
        return msg


a1=RSA()
a1.KeyGen()
msg = int(input("Enter your message: "))
signature=a1.CreateSignature(msg)
plain=a1.VerifySignature()
# print("Signature: ",signature)
# signature1 = int(input("Enter the sign: "))# plain =
VerifySignature(signature1)
if (plain == msg):
    print("Your Sign is verified")else:
    print("Your Sign is not valid")
```

Output:



```
G:\My Drive\SEM5\INS\INS_Inlab>py
Enter only PRIME NUMBERS:
Enter p value: 11
Enter q value: 3
*****
Value of n:  33
PHI of n:  20
Value of e:  3
Value of d:  7
Enter your message: 7
Public Key: [ 3 , 33 ]
Private Key: [ 7 , 33 ]
Signature:  28
Enter the sign: 28
Plain text:  7
Your Sign is verified
```

.

Questions:

1. In RSA cryptosystem each plaintext character is presented by the number between 00(A) and 25(Z). The number 26 represents the blank character. Bob wants to send Alice the message “Hello World”. So the plaintext is as below,
   07 04 11 11 14 26 22 14 17 11 03 . Suppose p=11, q=3. Find out digital signature.

   Ans: First we calculate n using n = p*q
   n = 11 x 3 = 33
   Then we calculate phi using phi = (p-1)*(q-1)
   phi = 10*2 = 20
   Then we find e &lt; phi such that it is relatively prime to phiWe got e = 3
   Then we find d using 1 = ed mod phiWe got d = 7
   Our private key becomes (d,n) = (7,33)Our public key becomes (e,n) = (3,33)
   Digital Signature is produced by using formula $C = M^d \bmod(n)$

   For 07 :
   $C = 7^7 \mathrm{mod} 33 = 28$

   For 04:
   $C = 4^7 \mathrm{mod} 33 = 16$

   For 11:
   $C = 11^7 \mathrm{mod} 33 = 11$

   For 14:
   $C = 14^7 \mathrm{mod} 33 = 20$

   For 26:
   $C = 26^7 \mathrm{mod} 33 = 5$

   For 22:
   $C = 22^7 \mathrm{mod} 33 = 22$

   For 14:
   $C = 14^7 \mathrm{mod} 33 = 20$

   For 17:
   $C = 17^7 \mathrm{mod} 33 = 8$

   For 11:
   $C = 11^7 \mathrm{mod} 33 = 11$

   For 3:
   $C = 3^7 \mathrm{mod} 33 = 9$
   Plain Text : 07 04 11 11 14 26 22 14 17 11 03

   Digital Signature : 28 16 11 11 20 5 22 20 8 11 9

Outcomes:
CO 2 - Illustrate different cryptographic algorithms for security

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)
Implemented digital signatures using RSA to illustrate different cryptographic algorithms forinformation and network security.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References: Books/ Journals/ Websites:

1. Behrouz A. Forouzan, "Cryptography and Network Security", Tata McGraw Hill
2. William Stalling, "Cryptography and Network Security", Prentice Hall