

**Experiment No. 4**

**Title:** Digital Signature using RSA

(Autonomous College Affiliated to University of Mumbai)

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

**Aim:** To implement digital signature using RSA

**Resources needed:** Windows/Linux, C or JAVA language

**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)*

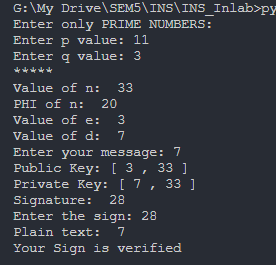
*# signature1 = int(input("Enter the sign: ")) # plain = VerifySignature(signature1)*

if (plain == msg):

print("Your Sign is verified") else:

print("Your Sign is not valid")

**Output:**



**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.

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 phi We got e = 3

Then we find d using 1 = ed mod phi We 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 = Md mod(n)

For 07 :

C = 77mod33 = 28

For 04:

C = 47mod33 = 16

For 11:

C = 117mod33 = 11

For 14:

C = 147mod33 = 20

For 26:

C = 267mod33 = 5

For 22:

C = 227mod33 = 22

For 14:

C = 147mod33 = 20

For 17:

C = 177mod33 = 8

For 11:

C = 117mod33 = 11

For 3:

C = 37mod33 = 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 for information and network security.

**Grade: AA / AB / BB / BC / CC / CD /DD Signature of faculty in-charge with date**

**References: Books/ Journals/ Websites:**

1. Charles P. Pfleeger, “Security in Computing”, Pearson Education
2. Behrouz A. Forouzan, “Cryptography and Network Security”, Tata McGraw Hill
3. William Stalling, “Cryptography and Network Security”, Prentice Hall