|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap Id: | 60004220127 |
| Class: | T. Y. B. Tech (Computer Engineering) |
| Course: | Information Security Laboratory |
| Course Code: | DJ19CEL603 |
| Experiment No.: | 08 |

**AIM:** Study and Implement RSA Digital Signature.

**CODE:**

import hashlib

import random

def gcd(a, b):

    while b != 0:

        a, b = b, a % b

    return a

def mod\_inverse(a, m):

    m0, x0, x1 = m, 0, 1

    while a > 1:

        q = a // m

        m, a = a % m, m

        x0, x1 = x1 - q \* x0, x0

    return x1 + m0 if x1 < 0 else x1

def generate\_key\_pair(p, q):

    n = p \* q

    phi = (p - 1) \* (q - 1)

    e = random.randrange(1, phi)

    g = gcd(e, phi)

    while g != 1:

        e = random.randrange(1, phi)

        g = gcd(e, phi)

    d = mod\_inverse(e, phi)

    return ((e, n), (d, n))

def rsa\_encrypt(message, public\_key):

    e, n = public\_key

    encrypted\_message = [pow(char, e, n) for char in message]

    return encrypted\_message

def rsa\_decrypt(encrypted\_message, private\_key):

    d, n = private\_key

    decrypted\_message = [chr(pow(char, d, n)) for char in encrypted\_message]

    return ''.join(decrypted\_message)

def md5\_hash(message):

    hash\_object = hashlib.md5(message.encode())

    return int.from\_bytes(hash\_object.digest(), byteorder='big')

def sign\_message(message, private\_key):

    hashed\_message = md5\_hash(message)

    signature = pow(hashed\_message, private\_key[1])

    return signature % (p \* q)

def verify\_signature(message, public\_key):

    hashed\_message = md5\_hash(message)

    decrypted\_signature = pow(hashed\_message, public\_key[1])

    return decrypted\_signature % (p \* q)

p = 61

q = 53

public\_key, private\_key = generate\_key\_pair(p, q) # print(public\_key, private\_key)

message = "This is Prerna Jadhav"

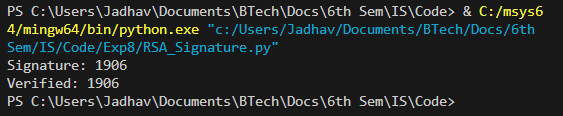
signature = sign\_message(message, private\_key)

print("Signature:", signature)

verified = verify\_signature(message, public\_key)

print("Verified:", verified)

**OUTPUT:**

****