Pandit Deendayal Energy University

School of Technology

Information Security Lab

B.Tech-Computer Science & Engineering (Sem-V)

PATEL VEDANT H.

19BCP138

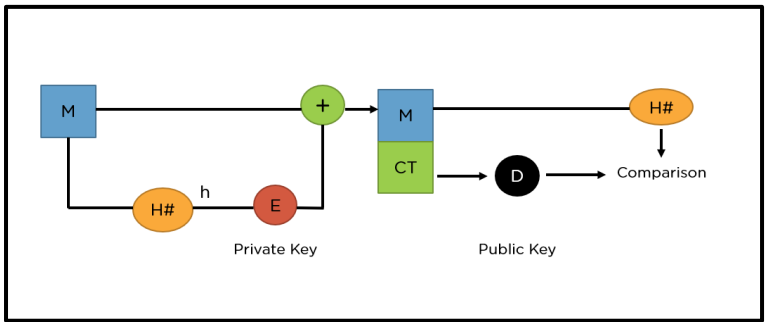
DIVISION – 2

Lab 10 Assignment

❖ **Aim:** Study and Implement of Digital Signature using RSA.

❖ **Introduction:**

RSA is an asymmetric algorithm to encrypt plaintext or to decrypt the ciphertext. It is the first algorithm known to be suitable for signing as well as encryption and was one of the first great advances in public-key cryptography. RSA is widely used in electronic commerce protocols and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser. Digital signatures are used to verify the authenticity of the message sent electronically. A digital signature algorithm uses a public key system. The intended transmitter signs his/her message with his/her private key and the intended receiver verifies it with the transmitter's public key. A digital signature can provide message authentication, message integrity, and non-repudiation services. The image below shows the entire procedure of the RSA algorithm.



❖ Program:

➢ Encrypt and Decrypt Python Program:

D:\Sem5\Information Security\Lab\Lab 10\IS-lab-10-Encrypt-Decrypt.py

temp.py × | IS-lab-10-Encrypt-Decrypt.py ×

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 15 22:53:46 2021

@author: vedpa
"""

from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import binascii

keyPair = RSA.generate(3072)
pubKey = keyPair.publickey()
pubKeyPEM = pubKey.exportKey()
privKeyPEM = keyPair.exportKey()

msg = b'Vedant Signature'
encryptor = PKCS1_OAEP.new(pubKey)
encrypted = encryptor.encrypt(msg)
print("Encrypted:", binascii.hexlify(encrypted))

decryptor = PKCS1_OAEP.new(keyPair)
decrypted = decryptor.decrypt(encrypted)
print('Decrypted:', decrypted)

```

➢ Encrypt and Decrypt Output:

Console 1/A ×

```
In [1]: runfile('D:/Sem5/Information Security/Lab/Lab10/IS-lab-10-Encrypt-
Decrypt.py', wdir='D:/Sem5/Information Security/Lab/Lab10')
Encrypted:
b'6f26b7542186cad7a1210346a43219976b7a131543f0c338ac4b77e070c424556feb9caceee
d84bee10fedf741d6cd0498920e177722b27ce8393d6928266564a994b8a078cf5c9569ec4c0
9fd8d8fc5bc579f5dc21a7628f2f5bbaa1b82414c8908635ffc1c7acdbde33ac2d41400037c0
e23c07d376e85279dca43f04a5685f7d7e4e86b2a9dab974adad111ef35db876d881215a0a31
becf473126ffdb9435965263b4014a7dcf46193644081fdc241ce23612d5c2472264e69ce1bc
c9d1aec5fd150713464f8a8dff46f06040ecf060227c2420f56e962a8907ea545c32cb3f23bb
4bea2f650e3a99114b159d2ccf92fee0b2fe85743197ac624c73460eabb9a0ae731c606c5758
e554e6ed35e87fda76f794bbb5621b78a038d0142342224a14c9d8ae6ab021d9682c1506f7a5
f7567d28ced4d421e65dba2ab96240cc019cca280464925c69b4d620c6d75bbc442e3e841341
4e5917501f13f9dcdffce96027d29624bc3bdcd622eb9c4977ab80e3f02ca9218f3349f6af95
4322405c7a'
Decrypted: b'Vedant Signature'
```

❖ Cryptanalysis:

Due to the principle, a quantum computer with a sufficient number of entangled quantum bits (qubits) can quickly perform a factorization because it can simultaneously test every possible factor simultaneously. So far, however, there is no known quantum computer, which has just an approximately large computing capacity. Thus, effective quantum computers are currently a myth that will probably not be ready for production in the next few years. However, factoring may be over in 20 years and RSA loses its security. The larger the prime factors are, the longer actual algorithms will take and the more qubits will be needed in future quantum computers.

❖ Applications:

- RSA encryption is often used in combination with other encryption schemes, or for digital signatures which can prove the authenticity and integrity of a message.
- RSA encryption can be used in several different systems. It can be implemented in OpenSSL, wolfCrypt, cryptlib, and several other cryptographic libraries.
- As one of the first widely used public-key encryption schemes, RSA laid the foundations for much of our secure communications. It was traditionally used in TLS and was also the original algorithm used in PGP encryption. RSA is still seen in a range of web browsers, email, VPNs, chat, and other communication channels.
- RSA is also often used to make secure connections between VPN clients and VPN servers. Under protocols like OpenVPN, TLS handshakes can use the RSA algorithm to exchange keys and establish a secure channel.

❖ Reference:

- https://www.geeksforgeeks.org/rsa-digital-signature-scheme-using-python/
- https://www.simplilearn.com/tutorials/cryptography-tutorial/rsa-algorithm/
- https://www.comparitech.com/blog/information-security/rsa-encryption/