**Practical 8**

**AIM:** **A digital signature is a mathematical scheme for presenting the authenticity of digital messages or documents. A valid digital signature gives a recipient reason to believe that • The message was created by a claimed sender (authentication), • The sender cannot deny having sent the message (non-repudiation), • The message was not altered in transit**

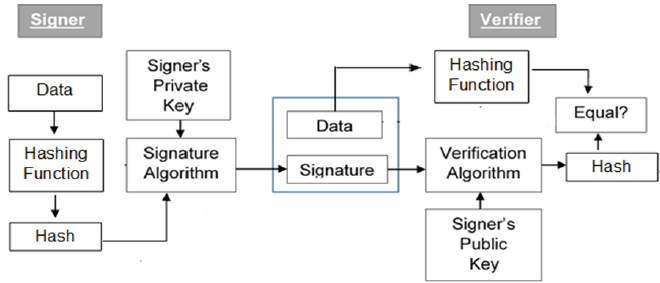
**Software Requirements:** PyCharm for Python

**Theory:** Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.

They provide the three following features:

1. **Authentication** — Digital signatures are bound to a specific user via their private key. Thus, they identify the owner of the private key used to sign the source data/message (e.g. document, email, or file).
2. **Integrity** — Digital signatures use a hashing algorithm to ensure that a message received is untampered. (More about hashing below.)
3. **Non-Repudiation** — Due to the two previous features, digital signatures ensure a sender who has signed the source data cannot deny having signed it at a later time.

Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.



**Code:**

import Crypto

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1\_OAEP

from Crypto import Random

import base64

def rsakeys():

length=1024

privatekey = RSA.generate(length, Random.new().read)

publickey = privatekey.publickey()

encryptor = PKCS1\_OAEP.new(publickey)

decryptor = PKCS1\_OAEP.new(privatekey)

return decryptor, encryptor

def encryptMsg(plain\_text):

cipher\_text=encryptor.encrypt(plain\_text)

b64cipher=base64.b64encode(cipher\_text)

return b64cipher

def decryptMsg(b64cipher):

decoded\_ciphertext = base64.b64decode(b64cipher)

plaintext = decryptor.decrypt(decoded\_ciphertext)

return plaintext

decryptor,encryptor=rsakeys() #generating keys

text=b"Hello Students" #Text to encrypt

print("Text: ",text)

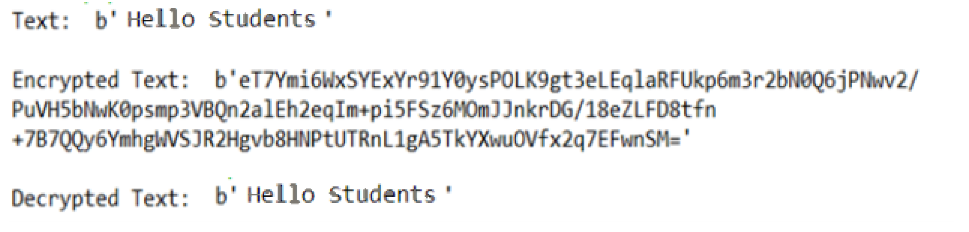
et=encryptMsg(text)

print("\nEncrypted Text: ",et)

dt=decryptMsg(et) #decryption

print("\nDecrypted Text: ",dt)

**Output:**



**Conclusion:** Thus, in this practical I performed Digital Signature which increases security and confidence as they cannot be reverse-engineered or forged. This is a very secure algorithm compare to the other algorithm.