**PRACTICAL-10**

**AIM :** Write a programto Implement a Digital Signature algorithm.

**INTRODUCTION**:

* [Digital Signatures](https://www.geeksforgeeks.org/digital-signatures-certificates/) are an asymmetrically encrypted hash of a digital message (data). It is a value that can provide a guarantee of authenticity, non-repudiation, and integrity.
* In other terms, it means you can verify the sender, date & time and message content have not been revealed or compromised.

**Digital Signature Flow:**

* Let “A” and “B” be the fictional actors in the cryptography system for better understanding.
* “A” is the sender and calculates the hash of the message and attaches signature which he wants to send using his private key.
* The other side “B” hashes the message and then decrypts the signature with A’s public key and compares the two hashes
* If “B” finds the hashes matching then the message has not been altered or compromised.

**CODE**:

import java.security.KeyPair;

import java.security.KeyPairGenerator;

import java.security.PrivateKey;

import java.security.PublicKey;

import java.security.SecureRandom;

import java.security.Signature;

import java.util.Scanner;

import javax.xml.bind.DatatypeConverter;

public class Digital\_Signature

{

private static final String SIGNING\_ALGORITHM = "SHA256withRSA";

private static final String RSA = "RSA";

private static Scanner sc;

public static byte[] Create\_Digital\_Signature( byte[] input, PrivateKey Key) throws Exception {

Signature signature = Signature.getInstance(SIGNING\_ALGORITHM);

signature.initSign(Key);

signature.update(input);

return signature.sign();

}

public static KeyPair Generate\_RSA\_KeyPair() throws Exception

{

SecureRandom secureRandom = new SecureRandom();

KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance(RSA);

keyPairGenerator.initialize(2048, secureRandom);

return keyPairGenerator.generateKeyPair();

}

public static boolean Verify\_Digital\_Signature(byte[] input,byte[] signatureToVerify,PublicKey key)throws Exception

{

Signature signature = Signature.getInstance(SIGNING\_ALGORITHM);

signature.initVerify(key);

signature.update(input);

return signature.verify(signatureToVerify);

}

public static void main(String args[])throws Exception

{

String input = "bhumit";

KeyPair keyPair = Generate\_RSA\_KeyPair();

byte[] signature = Create\_Digital\_Signature(input.getBytes(),keyPair.getPrivate());

System.out.println("Signature Value:\n "+ DatatypeConverter.printHexBinary(signature));

System.out.println( "Verification: "+Verify\_Digital\_Signature(input.getBytes(),signature,

keyPair.getPublic()));

}

}

**OUTPUT**:

