Practical No – 04

Digital Signatures

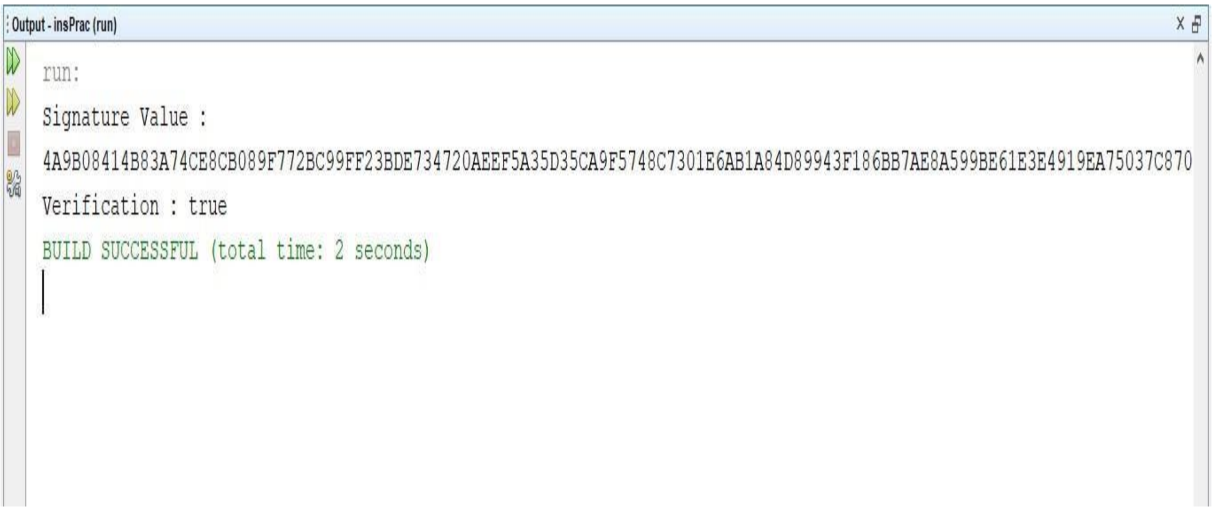
Aim: Implement digital signature algorithms such as RSA-based signatures, and verify the integrity and authenticity of digitally signed messages.

Source Code :

```
import java.security.PrivateKey;
import java.security.*;
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;
  //Function to implement Digital signature
  //Using SHA256 and RSA algorithm
  //By Passing private key
  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();
  }
  //Generate the Asymmetric key pair
  //Using SecureRandom class
  //Function and RSA Algorithm
  public  static KeyPair Generate_RSA_KeyPair() throws Exception{
    SecureRandom secureRandom = new SecureRandom();
    KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance(RSA);
    keyPairGenerator.initialize(2048, secureRandom);
```

```
        return keyPairGenerator.genKeyPair();

    }

    //Function for Verification of the Digital Signature by using the Public Key

    public static boolean Verify_Digital_Signature(byte[] input, byte[] signaturweToVerify, PublicKey key) throws Exception{

        Signature signature = Signature.getInstance(SIGNING_ALGORITHM);

        signature.initVerify(key);

        signature.update(input);

        return signature.verify(signaturweToVerify);

    }

    //Deliver Code

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

        String input = "Good Morning";

        KeyPair keyPair = Generate_RSA_KeyPair();

        //Function Call

        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 :



```
run:
Signature Value :
4A9B08414B83A74CE8CB089F772BC99FF23BDE734720AEEF5A35D35CA9F5748C7301E6AB1A84D89943F186BB7AE8A599BE61E3E4919EA75037C870
Verification : true
BUILD SUCCESSFUL (total time: 2 seconds)
```