Practical no. 9

Aim:- To write a program to implement signature and digital signature technique.

Theory:- Digital Signatures are often calculated using elliptical curve cryptography especially in IOT devices, but we will be using RSA for ~~demosta~~ demonstration purposes. First, we will take the input message and create a hash of it using SHA-256 because of its speed and security. On the other side the receiver will decrypt it using the public key and compare the hash to ensure they are indeed the same.

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 private key.
- The other side 'B' hashes the message ~~like~~ and then decrypts the signature with 'A''s public key and compares two hashes.
- If 'B' finds the hashes matching then the message has not been altered or compromosed.

Algorithm:-

Let us implement the digital signature using algorithm SHA and RSA and also verify if the hash mathes with a public key.

1) Created a method named Create_Digitare_Signature() to implement Digital signature by passing two parameters input message and the private key. In this method we will get an instance of the signature objects passing the signing algorithm and assign it with a private key and finally pass the input this will return byte array.

2> The next step is to generate assymetric key pair using RSA algorithm and secure random class functions.

```
secureRandom = new secureRandom();
Key pair generator keypairgenerator.get-Instance(ALGORITHM);
```

3> Finally verifying the signature using public key. Verify-Digital-Signature() method is used to check whether the signature matches by passing it the input, signature and public key.

```
Signature signature = Signature.getInstance(SIGNING-ALGORITHM)
    Signature.initVerify(publickey);
    Signature.update(input);
```

Conclusion:- A program to implement signature and digital signature technique has been executed successfully.

Ashahakar
A

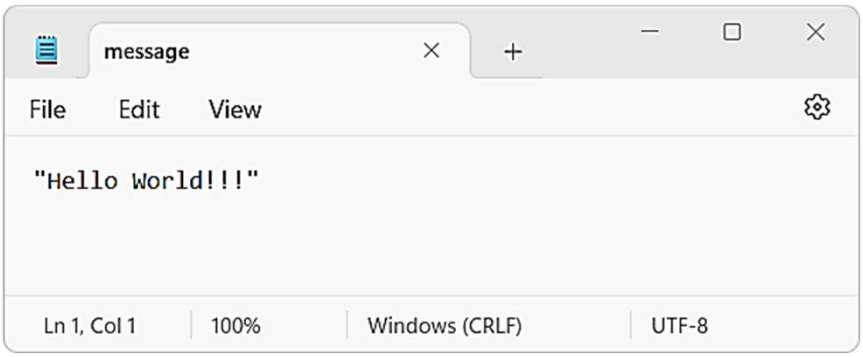
Program:

GenerateDigitalSignature.java file:

```
import java.io.*;
import java.security.*;

public class GenerateDigitalSignature {
   public static void main(String[] args) {
      if (args.length != 1)
         System.out.println("Usage: nameOfFileToSign");
      else try {
         KeyPairGenerator keyGen = KeyPairGenerator.getInstance("DSA", "SUN");
         SecureRandom random = SecureRandom.getInstance("SHA1PRNG", "SUN");
         keyGen.initialize(1024, random);
         KeyPair pair = keyGen.generateKeyPair();
         PrivateKey priv = pair.getPrivate();
         PublicKey pub = pair.getPublic();
         Signature dsa = Signature.getInstance("SHA1withDSA", "SUN");
         dsa.initSign(priv);
         FileInputStream fis = new FileInputStream(".\\message.txt");
         BufferedInputStream bufin = new BufferedInputStream(fis);
         byte[] buffer = new byte[1024];
         int len;
         while (bufin.available() != 0) {
            len = bufin.read(buffer);
            dsa.update(buffer, 0, len);
         };
         bufin.close();
         byte[] realSig = dsa.sign();
         FileOutputStream sigfos = new FileOutputStream(".\\signature.txt");
         sigfos.write(realSig);
         sigfos.close();
         byte[] key = pub.getEncoded();
         FileOutputStream keyfos = new FileOutputStream(".\\publicKey.txt");
         keyfos.write(key);
         keyfos.close();
      }
      catch (Exception e) {
         System.err.println("Caught exception " + e.toString());
      }
   };
}
```

Message.txt file:



Generating Signature and Public Key files:

```
Terminal    Local ×
PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java> cd src
PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java\src> javac GenerateDigitalSignature.java
PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java\src> java GenerateDigitalSignature message

 Digital Signature Generation Algorithm

 Generated Signature file : 'signature.txt'

 Generated Public Key file : 'publicKey.txt'

PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java\src>
```

Program:

VerifyDigitalSignature.java file:

```
import java.io.*;
import java.security.*;
import java.security.spec.*;

public class VerifyDigitalSignature {
   public static void main(String args[]) {
      if (args.length != 3)
         System.out.println("Usage: publickeyfile signaturefile datafile")
      else try {
         FileInputStream keyfis = new FileInputStream(".\\publicKey.txt");
         byte[] encKey = new byte[keyfis.available()];
         keyfis.read(encKey);
         keyfis.close();
         X509EncodedKeySpec pubKeySpec = new X509EncodedKeySpec(encKey);
         KeyFactory keyFactory = KeyFactory.getInstance("DSA", "SUN");
         PublicKey pubKey = keyFactory.generatePublic(pubKeySpec);
         FileInputStream sigfis = new FileInputStream(".\\signature.txt");
         byte[] sigToVerify = new byte[sigfis.available()];
         sigfis.read(sigToVerify );
         sigfis.close();
         Signature sig = Signature.getInstance("SHA1withDSA", "SUN");
         sig.initVerify(pubKey);
         FileInputStream datafis = new FileInputStream(".\\message.txt");
         BufferedInputStream bufin = new BufferedInputStream(datafis);
         byte[] buffer = new byte[1024];
         int len;
         while (bufin.available() != 0) {
            len = bufin.read(buffer);
            sig.update(buffer, 0, len);
         };
         bufin.close();
         boolean verifies = sig.verify(sigToVerify);
         System.out.println("signature verifies: " + verifies);
      }
      catch (Exception e) {
         System.err.println("Caught exception " + e.toString());
      };
   }
}
```

signature.txt file:

signature

File Edit View

```
0-□□X*5ûè˜ èsF□b
□)ŽG}6
□□ ‡Kj© À□‹9-I¹□“Íª¸Ü□ý
```

Ln 3, Col 24 | 100% | Macintosh (CR) | ANSI

publicKey.txt file:

publicKey

File Edit View

```
0,□·0,□,□ *†HÎ8□□0,□□□□□ ý□S□□u□)RßJœ.ìäçö□·R<ïD Ã□?
€¶Q&iE]@"QûY=□Xú¿Åõº0öË›Ul×□;€□4oòf`·k™P¥¤ŸŸè□{□"ÂO»©×þ·
Æ□ø;WçÆ¨¦□□□ûƒöÓÅ□Ã□5T□Z□‘2öuó®+a×*ïò"□□□ÑH□Ç□□ —`P□□#
Ì²’¹,¢ë„
ðX□õ□□□ ÷á …Ö›=ÞË¼«\6¸W¹y”¯»ú:ê,ùWL
= ,gQYWŽºÔYOæq □□€´I□q#èL(□□·Ï  2ŒÈ¦á<□z‹T|□(à£®□+³¦u‘n£□
ú!5bñûbz□$;Ì¤ñ¼¨Q□‰¨ƒßáZåŸ□’‹f^€{U%d□L;þÏI*□□„ □□€O¥¢<·›
□,²Jx □'›/,Úõí□□□DyÒ□È‰□÷®□ygIX3_+G—    U
¤6ä©□□‘,ÁÎŽâUÇ□¬È†„¿Á.dlá•»-BÞ’G"‡E@§a{p6öû□Œ@@çõ%□VNO
âë¥Ð7?¥™6õºµ#A"&-Dðœ©□¤□
-ÿÆá
```

Ln 1, Col 1 | 100% | Macintosh (CR) | ANSI

Verifying Signature:

Terminal Local

```
PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java\src> javac VerifyDigitalSignature.java
PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java\src> java VerifyDigitalSignature publicKey signature message

 Digital Signature Verification Algorithm

 Verifying Signature...

 Signature verifies: true

PS D:\IntelliJ IDEA Community Edition 2021.2.2\Projects\Java\src>
```