IT8761 – Security Laboratory

Reshma Ramesh Babu

312217104129

Exercise 9

Aim: To implement the Signature Scheme - Digital Signature Standard

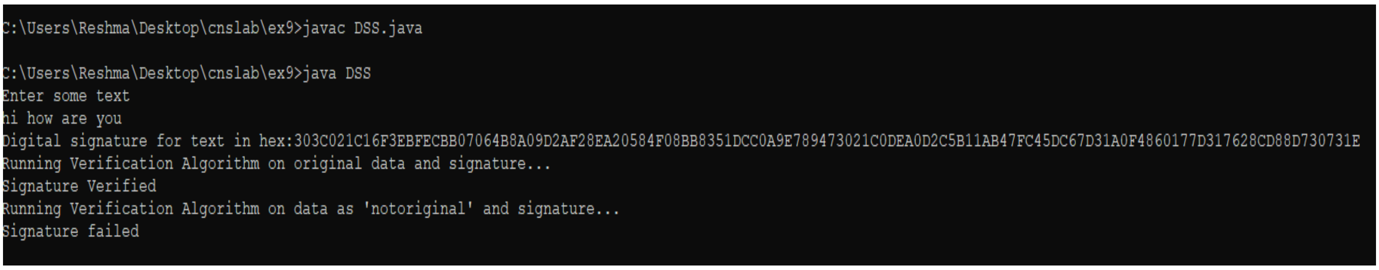
Code:

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.Signature;
import java.util.Scanner;

public class DSS {
  PublicKey pubk;
  private PrivateKey prvk;

  DSS() throws NoSuchAlgorithmException
  {
    KeyPairGenerator kpg = KeyPairGenerator.getInstance("DSA");
    kpg.initialize(2048); // 2048 is the keysize.
    KeyPair kp = kpg.generateKeyPair();
    pubk = kp.getPublic();
    prvk = kp.getPrivate();
  }
  public String createSignature(String text)
```

```
{  try{
      //Creating a Signature object
      Signature sign = Signature.getInstance("SHA256withDSA");
      //Initialize the signature
      sign.initSign(prvk);
      byte[] bytes = text.getBytes();
      //Adding data to the signature
      sign.update(bytes);
      //Calculating the signature
      byte[] signature = sign.sign();
      return bytesToHex(signature);
  }
  catch(Exception e)
  {
    System.out.println("Error:"+e.getMessage());
    return "";
  }

}
public String verifySignature(String text,String signatureReceived)
{   try{

    //Creating a Signature object
    Signature sign = Signature.getInstance("SHA256withDSA");
    byte[] bytes = text.getBytes();
    sign.initVerify(pubk);
    sign.update(bytes);
```

```
      boolean bool = sign.verify(hextoBytes(signatureReceived));
      if(bool==true)
        return "Signature Verified";
      else
       return "Signature failed";
    }
    catch(Exception e)
    {
      System.out.println("Error:"+e.getMessage());
      return "";
    }

  }
  private final static char[] hexArray = "0123456789ABCDEF".toCharArray();
  public static String bytesToHex(byte[] bytes) {
    char[] hexChars = new char[bytes.length * 2];
    for ( int j = 0; j < bytes.length; j++ ) {
       int v = bytes[j] & 0xFF;
       hexChars[j * 2] = hexArray[v >>> 4];
       hexChars[j * 2 + 1] = hexArray[v & 0x0F];
    }
    return new String(hexChars);
 }
 public static byte[] hextoBytes(String hexString)
 {      byte[] val = new byte[hexString.length() / 2];
      for (int i = 0; i < val.length; i++) {
        int index = i * 2;
        int j = Integer.parseInt(hexString.substring(index, index + 2), 16);
        val[i] = (byte) j;
      }

      return val;
  }
  public static void main(String args[]) throws Exception {
    //Accepting text from user
    Scanner sc = new Scanner(System.in);
    DSS dss = new DSS();

    System.out.println("Enter some text");
    String text = sc.nextLine();
    String signature = dss.createSignature(text);
    System.out.println("Digital signature for text in hex:"+ signature);

    System.out.println("Running Verification Algorithm on original data and signature...");
    System.out.println(dss.verifySignature(text, signature));
    System.out.println("Running Verification Algorithm on data as 'notoriginal' and signature...");
    System.out.println(dss.verifySignature("notoriginal", signature));

    sc.close();
  }
}
```

Output:



```
C:\Users\Reshma\Desktop\cnslab\ex9>javac DSS.java

C:\Users\Reshma\Desktop\cnslab\ex9>java DSS
Enter some text
hi how are you
Digital signature for text in hex:303C021C16F3EBFECBB07064B8A09D2AF28EA20584F08BB8351DCC0A9E789473021C0DEA0D2C5B11AB47FC45DC67D31A0F4860177D317628CD88D730731E
Running Verification Algorithm on original data and signature...
Signature Verified
Running Verification Algorithm on data as 'notoriginal' and signature...
Signature failed
```