Shankar Narayanan S
312217104146
30-10-2020

SSN COLLEGE OF ENGINEERING
Department of Computer Science and Engineering
IT 8761 - Security Lab

Exercise 9: To implement the Signature Scheme -Digital Signature Standard

CODE :

```
import java.util.*;
import java.lang.*;
import java.io.*;
import java.security.*;
import java.math.*;

class DSS{
   public String SIGNING_ALGORITHM = "SHA256withRSA";

   public KeyPair generateRSAKeyPair() throws Exception{
         KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
         kpg.initialize(2048);
         return kpg.generateKeyPair();
   }

   public byte[] createDigitalSignature(
                  byte[] input,
                  PrivateKey Key
         ) throws Exception{
         Signature Sign = Signature.getInstance(SIGNING_ALGORITHM);
         Sign.initSign(Key);
         Sign.update(input);
         return Sign.sign();
   }

   public boolean verifyDigitalSignature(
                  byte[] input,
                  byte[] sign,
                  PublicKey Key
         ) throws Exception{
```

```
        Signature signature = Signature.getInstance(SIGNING_ALGORITHM);
        signature.initVerify(Key);
        signature.update(input);

        return signature.verify(sign);

    }
}

public class DSSDriver{

    public static void main(String[] args) throws Exception{
        Scanner in = new Scanner(System.in);

        String input;
        System.out.println("Enter the text to sign : ");
        input = in.nextLine();

        DSS dss = new DSS();
        KeyPair keypair = dss.generateRSAKeyPair();

        byte[] signature = dss.createDigitalSignature(input.getBytes(), keypair.getPrivate());

        BigInteger s = new BigInteger(signature);
        System.out.println("\n\nSignature : "+s.toString(16).toUpperCase());

        if(dss.verifyDigitalSignature(input.getBytes(), signature, keypair.getPublic())){
                System.out.println("\nVerification : successful");
        } else {
                System.out.println("\nVerification : failure");
        }

        System.out.println("Modifying the signature : ");

        signature[0]=3;

        System.out.println("Performing verification again ; ");

        s = new BigInteger(signature);
        System.out.println("\n\nSignature : "+s.toString(16).toUpperCase());

        if(dss.verifyDigitalSignature(input.getBytes(), signature, keypair.getPublic())){
                System.out.println("\nVerification : successful");
```

```
        } else {
                System.out.println("\nVerification : failure");
        }

   }
}
```

OUTPUT :

(base) Shankars-MacBook-Pro:Ex14 shankar99$ javac DSSDriver.java
(base) Shankars-MacBook-Pro:Ex14 shankar99$ java DSSDriver

Enter the text to sign :
i am kira

Signature :
4F7858953E4D0EC6087A5512A196791AAAFF772AF268E9C52539F8CDCA98D5F54048375
1E912584AE50BCCFB74B50EF175B59819EFDB35E0500C4E093E1BF1118A3D3029B50A6
7D305F221CBFF374BEB9FF75A500792EBEB700DA9647FD7256AF077D82680695E3C520
B18E517E5E5658D76C7FE80130C5FE652D0FA2863B345D2F02D47F3CD934C8D23083060
75CDD6D831A6C8C8D622106A7E9D83949CD9A05068FA0C7B366BB7788DC66BF5651BC
2078C15C0BDE3B35A2288860D2C8F2887C03360931149ACBF15D6B416BE81415E3632F6
77DC3E68B83942643152E99DE946906D387AAED83160C23561BE1F8553A2F6E5F53B028
B5B46360C988E97A992

Verification : successful

Modifying the signature :

Signature :
37858953E4D0EC6087A5512A196791AAAFF772AF268E9C52539F8CDCA98D5F540483751
E912584AE50BCCFB74B50EF175B59819EFDB35E0500C4E093E1BF1118A3D3029B50A67
D305F221CBFF374BEB9FF75A500792EBEB700DA9647FD7256AF077D82680695E3C520B
18E517E5E5658D76C7FE80130C5FE652D0FA2863B345D2F02D47F3CD934C8D230830607
5CDD6D831A6C8C8D622106A7E9D83949CD9A05068FA0C7B366BB7788DC66BF5651BC2
078C15C0BDE3B35A2288860D2C8F2887C03360931149ACBF15D6B416BE81415E3632F67
7DC3E68B83942643152E99DE946906D387AAED83160C23561BE1F8553A2F6E5F53B028B
5B46360C988E97A992

Performing verification again :

Verification : failure

(base) Shankars-MacBook-Pro:Ex14 shankar99$

Result : Performed digital signature authentication using DSS with SHA-256 as the hash function and RSA for key generation.Verification was successful.