**EXPERIMENT-8**

**Student’s Name: Kanishk Soni UID: 20BCS9398**

**Section/Group: 707\_WM\_B Subject Code: 20CSP-338**

**Subject Name: WMS Lab Date of performance:27/10/2022**

**Branch: BE CSE Semester:5th**

**Aim:** Write a program to sign and verify a document using DSA algorithm

**Objective:** To generate the concept of digital signature

**Software/Hardware Requirements:** Java, Python platform

**Discussion**:

The **digital signature** is an electronic signature to sign a document, mail, messages, etc. It validates the **authenticity,** and **integrity** of a message or document. It is the same as a handwritten signature, seal, or stamp. It is widely used to verify a digital message, financial documents, identity cards, etc.

**Coding:**

**GenerateDigitalSignature.java**

import java.io.\*;

import java.security.\*;

public class GenerateDigitalSignature {

public static void main(String[] args) {

try {

KeyPairGenerator keyGen = KeyPairGenerator.getInstance("DSA", "SUN");

SecureRandom random = SecureRandom.getInstance("SHA1PRNG", "SUN");

keyGen.initialize(1024, random);

KeyPair pair = keyGen.generateKeyPair();

PrivateKey privateKey = pair.getPrivate();

PublicKey publicKey = pair.getPublic();

Signature dsa = Signature.getInstance("SHA1withDSA", "SUN");

dsa.initSign(privateKey);

File inf = new File("src/wms/DSA/digital.txt");

FileInputStream fin = new FileInputStream(inf.getAbsolutePath());

BufferedInputStream bin = new BufferedInputStream(fin);

byte[] buf = new byte[1024];

int len;

while(bin.available() != 0) {

len = bin.read(buf);

dsa.update(buf, 0, len);

}

bin.close();

byte[] realSig = dsa.sign();

File outf = new File("src/wms/DSA/signature.txt");

FileOutputStream sfout = new FileOutputStream(outf.getAbsolutePath());

sfout.write(realSig);

sfout.close();

byte[] key = publicKey.getEncoded();

File outfk = new File("src/wms/DSA/publicKey.txt");

FileOutputStream kfout = new FileOutputStream(outfk.getAbsolutePath());

kfout.write(key);

kfout.close();

} catch (Exception e) {

System.out.println("Caught Exception: " + e);

}

}

}

**VerifyDigitalSignature.java**

import java.io.\*;

import java.security.\*;

import java.security.spec.\*;

public class VerifyDigitalSignature {

public static void main(String[] args) {

try {

File inf = new File("src/wms/DSA/publicKey.txt");

FileInputStream keyfin = new FileInputStream(inf.getAbsolutePath());

byte[] encKey = new byte[keyfin.available()];

keyfin.read(encKey);

keyfin.close();

X509EncodedKeySpec pubKeySpec = new X509EncodedKeySpec(encKey);

KeyFactory keyFactory = KeyFactory.getInstance("DSA", "SUN");

PublicKey publicKey = keyFactory.generatePublic(pubKeySpec);

File sinf = new File("src/wms/DSA/signature.txt");

FileInputStream sigfin = new FileInputStream(sinf.getAbsolutePath());

byte[] sigToVerify = new byte[sigfin.available()];

sigfin.read(sigToVerify);

sigfin.close();

Signature sig = Signature.getInstance("SHA1withDSA", "SUN");

sig.initVerify(publicKey);

File dinf = new File("src/wms/DSA/digital.txt");

FileInputStream datafin = new FileInputStream(dinf.getAbsolutePath());

BufferedInputStream bufin = new BufferedInputStream(datafin);

byte[] buf = new byte[1024];

int len;

while(bufin.available() != 0) {

len = bufin.read(buf);

sig.update(buf, 0, len);

}

bufin.close();

boolean verifies = sig.verify(sigToVerify);

System.out.println("Signature verifies: " + verifies);

} catch (Exception e) {

System.out.println("Caught Exception: " + e.toString());

}

}

}

**Output:**







