EX.NO: ROLL.NO: 210701275

DATE: Digital Signature Algorithm

AIM:

Demonstrating digital signature generation and verification using RSA and SHA-256.

ALGORITHM:

1. Generate RSA key pair with a 2048-bit key size.
2. Create digital signature by hashing input with SHA-256 and encrypting with private key.
3. Verify signature by decrypting with public key and comparing hash with input.
4. Output signature in hexadecimal format.
5. Output verification result as boolean.

PROGRAM:

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

import javax.xml.bind.DatatypeConverter;

public class Dsa {

private static final String
SIGNING_ALGORITHM
= "SHA256withRSA";
private static final String RSA = "RSA";
private static Scanner sc;

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();
}

public static KeyPair Generate_RSA_KeyPair()
throws Exception
{
SecureRandom secureRandom
= new SecureRandom();
KeyPairGenerator keyPairGenerator
= KeyPairGenerator
.getInstance(RSA);
keyPairGenerator
.initialize(
2048, secureRandom);
return keyPairGenerator
.generateKeyPair();
}

public static boolean
Verify_Digital_Signature(
byte[] input,
byte[] signatureToVerify,
PublicKey key)
throws Exception
{
Signature signature
= Signature.getInstance(
SIGNING_ALGORITHM);
signature.initVerify(key);
signature.update(input);
return signature
.verify(signatureToVerify);
}

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

String input
= "GEEKSFORGEEKS IS A"
+ " COMPUTER SCIENCE PORTAL";
```

```
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:

RESULT: