

## Experiment: 3.1

**Student Name:** Yash Kumar

**Branch:** BE-CSE

**Semester:** 5<sup>th</sup>

**Subject Code:** 21CSP-338

**UID:** 20BCS9256

**Section/Group:** 616 'B'

**Subject Name:** WMS Lab

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

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

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

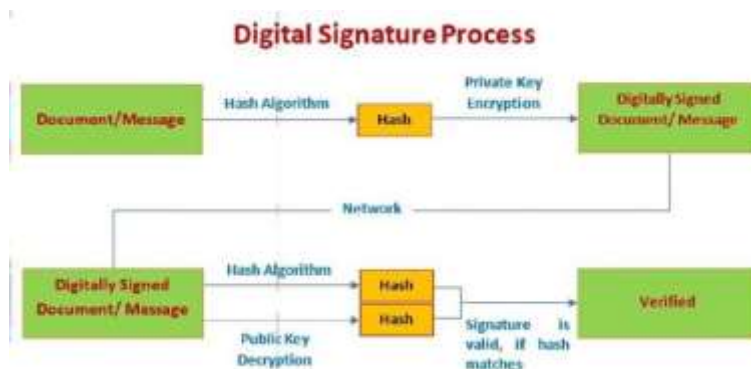
### Discussion:

The digital signature is a mechanism that verifies the authority of digital messages as well as documents. It is very popular because it provides more security than other signatures. In Java, JDK Security API is used to create and implement digital signatures. In this section, we will discuss the digital signature mechanism and also implement the digital signature mechanism in a Java program.

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.

In short, we can say that it ensures the following:

- **Integrity:** It ensures the message or a document cannot be altered while transmitting.
- **Authenticity:** The author of the message is really who they claim to be.
- **Non-repudiation:** The author of the message can't later deny that they were the source.



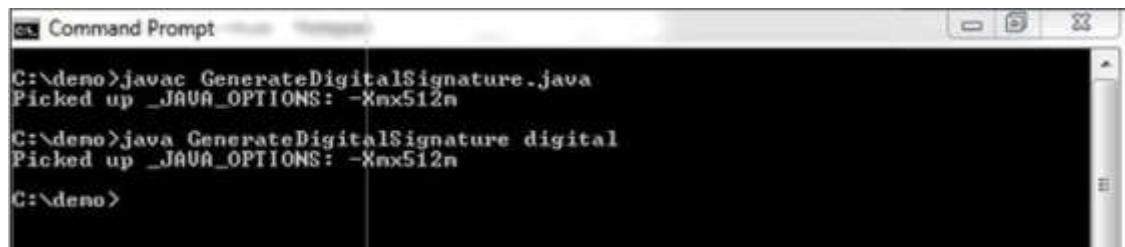
## Examples:

## Steps/Method/Code:

```

1. import java.io.*; //input the file data to be signed
2. import java.security.*; //provides methods for signing the data
3. public class GenerateDigitalSignature
4. {
5. public static void main(String args[])
6. {
7. /* Generate a DSA signature */
8. if (args.length != 1)
9. {
10. System.out.println("Usage: nameOfFileToSign");
11. }
12. else try
13. {
14. // the rest of the code goes here
15. }
16. catch (Exception e)
17. {
18. System.err.println("Caught exception " + e.toString());
19. }
20. }
21. }
  
```

## Output Screenshot:



```
CL Command Prompt
C:\deno>javac GenerateDigitalSignature.java
Picked up _JAVA_OPTIONS: -Xmx512m

C:\deno>java GenerateDigitalSignature digital
Picked up _JAVA_OPTIONS: -Xmx512m

C:\deno>
```

## VerifyDigitalSignature.java

```
1. import java.io.*;
2. import java.security.*;
3. import java.security.spec.*;
4. public class VerifyDigitalSignature
5. {
6.     public static void main(String args[])
7.     {
8.         /* Verify a DSA signature */
9.         if (args.length != 3) {
10.             System.out.println("Usage: VerifyDigitalSignature " + "publickeyfile signaturefile " + "datafile");
11.         }
12.         else try
13.         {
14.             // the rest of the code goes here
15.         }
16.         catch (Exception e)
17.         {
18.             System.err.println("Caught exception " + e.toString());
19.         }
20.     }
21. }
```

## Learning Outcomes:

With this, you have understood the importance of asymmetric cryptography, the working of digital signatures, the functionality of DSA, the steps involved in the signature verification, and its advantages over similar counterparts.