MIT World Peace University

Information and Cybersecurity
Second Year B. Tech, Semester 1

Implementation of Digital Signatures

Lab Assignment 7

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

March 29, 2023

Contents

1 Aim

Write a program using JAVA or Python or C++ to implement Digital signature using DSA

2 Objectives

To learn authentication technique for access control

3 Theory

3.1 The Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) is a public key cryptographic algorithm that is used to create and verify digital signatures. It was developed by the US National Institute of Standards and Technology (NIST) and is based on the mathematical concept of modular exponentiation.

The DSA algorithm consists of three main components: key generation, signature generation, and signature verification.

3.2 Algorithm

3.2.1 Key Generation:

1. Choose a prime number p such that p is 1024 bits or longer, and $p-1$ is divisible by a 160-bit prime number q.
2. Choose an integer g such that $1 < g < p$ and $g^{(p-1)/q} \bmod p \neq 1$.
3. Choose a random integer x such that $0 < x < q$.
4. Compute $y = g^x \bmod p$.
5. The public key is (p, q, g, y), and the private key is x.

3.2.2 Signature Generation:

1. Choose a random integer k such that $0 < k < q$.
2. Compute
$$r = (g^k \bmod p) \bmod q$$
3. Compute
$$s = k^{-1}(H(m) + xr) \bmod q$$
where $H(m)$ is the hash of the message m.
4. The signature is (r, s).

3.2.3 Signature Verification:

1. Verify that $0 < r < q$ and $0 < s < q$.
2. Compute $w = s^{-1} \bmod q$.
3. Compute
$$u_1 = (H(m)w) \bmod q$$
and
$$u_2 = (rw) \bmod q$$
4. Compute
$$v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q$$
5. If $v = r$, the signature is valid. Otherwise, it is invalid.

3.3 Example

Here is an example of how the DSA algorithm works:

Suppose Alice wants to send a message to Bob and sign it using DSA. Alice and Bob have already generated their public and private keys.

3.3.1 Signature Generation:

1. Alice chooses a random integer $k = 123$.
2. Alice computes
$$r = (g^k \bmod p) \bmod q = (2^{123} \bmod 467) \bmod 61 = 8$$
3. Alice computes
$$s = k^{-1}(H(m) + xr) \bmod q = 123^{-1}(H(m) + 22 \cdot 8) \bmod 61$$
where $H(m)$ is the hash of the message m.
4. Alice sends the message and the signature (r, s) to Bob.

3.3.2 Signature Verification:

1. Bob receives the message and the signature (r, s) from Alice.
2. Bob verifies that $0 < r < q$ and $0 < s < q$.
3. Bob computes
$$w = s^{-1} \bmod q = 123^{-1} \bmod 61 = 31$$
4. Bob computes
$$u_1 = (H(m)w) \bmod q$$
and
$$u_2 = (rw) \bmod q$$
where $H(m)$ is the hash of the message m.

5. Bob computes

$$v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q = ((2^{39} \cdot 22^{43}) \bmod 467) \bmod 61 = 8$$

6. Since $v = r$, the signature is valid, and Bob knows that the message was sent by Alice and has not been altered.

4 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters : Python 3.10.1

5 Input and Output

```
The Given Signature is Valid
```

6 Code

```python
import hashlib
import Crypto.PublicKey.RSA as RSA

# Generating keys
key = RSA.generate(2048)
private_key = key.exportKey('PEM')
public_key = key.publickey().exportKey('PEM')


# Message to sign
message = b'This is a message to be signed'

# Hashing the message
hash = hashlib.sha256(message).digest()

# Signing the hash
signature = key.sign(hash, '')

# Verifying the signature
if key.publickey().verify(hash, signature):
    print("Signature is valid")
else:
    print("Signature is not valid")
```

Listing 1: "DSA Signature Validity using PyCrypto Library"

7 Conclusion

Thus, we have seen how to implement digital signatures using DSA algorithm.

8 FAQ

1. **What are various digital signatures algorithms?**

   (a) RSA (Rivest-Shamir-Adleman): The RSA algorithm is one of the most widely used public-key cryptographic algorithms. It is used for both encryption and digital signatures. RSA signatures are computed using the signer's private key and verified using their public key.

   (b) DSA (Digital Signature Algorithm): The DSA algorithm is a public-key cryptographic algorithm used to create and verify digital signatures. It was developed by the US National Institute of Standards and Technology (NIST).

   (c) ECDSA (Elliptic Curve Digital Signature Algorithm): The ECDSA algorithm is a variant of the DSA algorithm that uses elliptic curve cryptography instead of modular arithmetic. It is commonly used in mobile and IoT devices because it requires less computational power than other signature algorithms.

   (d) EdDSA (Edwards-curve Digital Signature Algorithm): The EdDSA algorithm is another variant of the DSA algorithm that uses Edwards curves instead of elliptic curves. It is designed to be faster and more secure than other signature algorithms.

2. **Draw the diagrams of digital signature generation and verification.**

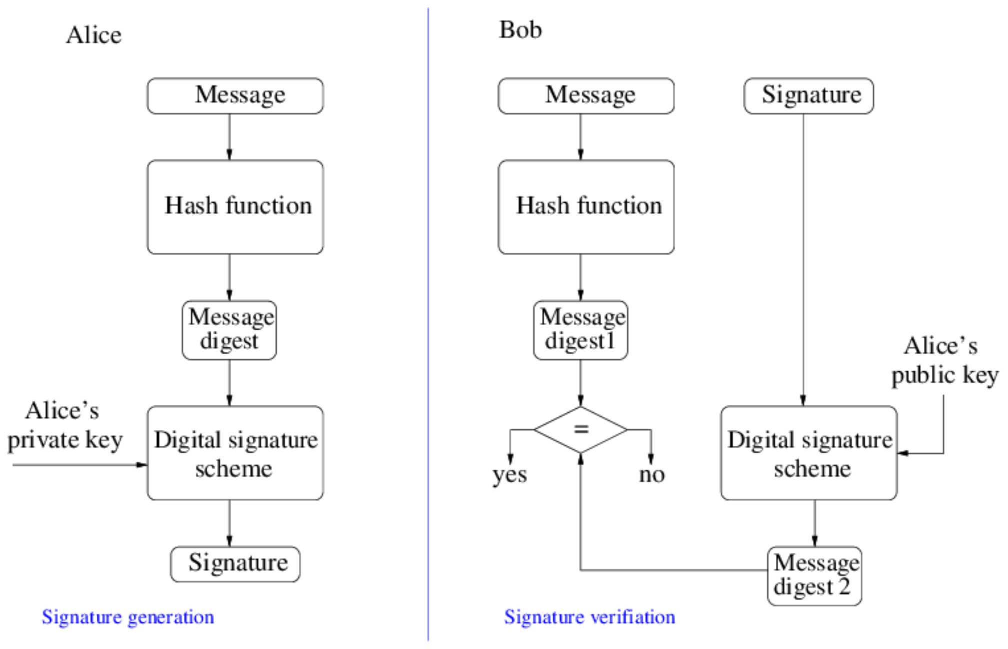

Figure 1: The General Process of Digital Signature Generation and Verification

3. **Which government agencies are involved to issue the digital signature? What is the validity of digital signature**

Government Agencies:

The government agencies involved in issuing digital signatures may vary depending on the country. In general, it is common for a government agency responsible for electronic signatures or digital certificates to issue digital signatures.

For example, in the United States, the *National Institute of Standards and Technology (NIST)* provides guidelines and standards for digital signatures and certificates.

Agencies in India

- In India, digital signatures are issued by *Certifying Authorities (CAs)* that are licensed by the Controller of Certifying Authorities (CCA). The CCA is a government agency under the Ministry of Electronics and Information Technology, responsible for the regulation and licensing of CAs in India.

- The CCA is responsible for implementing the provisions of the Information Technology (IT) Act, 2000, which provides for the legal recognition of electronic records and digital signatures in India. The CCA issues licenses to CAs for issuing digital certificates, and also maintains a national repository of digital certificates that can be used to verify the authenticity of digital signatures.
- In addition to the CCA, the Ministry of Electronics and Information Technology and the Ministry of Law and Justice also play a role in the regulation and promotion of digital signatures and electronic transactions in India.

Validity of Digital Signature:

The validity of a digital signature also varies depending on the country and the laws governing electronic signatures. In general, a digital signature is considered to be legally binding and valid if it meets certain criteria, such as:

(a) The signature is created using a valid digital certificate issued by a trusted certification authority.

(b) The signer's private key is kept secure and not accessible to others.

(c) The signature was created at the time the document was signed and has not been altered since then.

(d) The certificate used to create the signature has not expired or been revoked.

(e) In most countries, digital signatures are considered to be as legally binding as traditional signatures.

(f) The validity of a digital signature can be verified by using the signer's public key to decrypt and verify the signature, and by checking the certificate used to create the signature to ensure that it is valid and has not been revoked.