MIT World Peace University

Information and Cyber Security

Assignment 5

NAMAN SONI ROLL NO. 10

Contents

1 Aim

Write a program using JAVA or Python or C++ to implement integrity of message using MD5 or SHA

2 Objectives

To use of hashing algorithm to check message integrity

3 Theory

3.1 Explain MD5 or SHA

MD5 and SHA are cryptographic hashing algorithms used to generate a unique digital fingerprint, or hash, for a given message or data.

MD5 (Message Digest Algorithm 5) is a widely used hash function that generates a 128-bit hash value. It was developed by Ronald Rivest in 1991 and is still used today, although it has some security weaknesses that make it vulnerable to certain types of attacks. MD5 is often used for integrity checking, message authentication, and digital signatures.

SHA (Secure Hash Algorithm) is a family of cryptographic hash functions designed by the National Security Agency (NSA). SHA-1 is a widely used hash function that generates a 160-bit hash value. However, due to some security concerns, it is now considered to be insecure and has been replaced by SHA-2 and SHA-3, which generate hash values of various lengths ranging from 224 bits to 512 bits.

Both MD5 and SHA are one-way functions, meaning that it is practically impossible to generate the original message from its hash value. They are widely used in various applications, such as data integrity checking, password storage, digital signatures, and more. However, it is important to note that no hashing algorithm is completely secure and can be vulnerable to various attacks.

4 Code

```python
# Hashing Alogorithm

import hashlib

input1=input("Enter text/image:: ")
output1=hashlib.md5(input1.encode()) #encode string into bytes
print("The hashing value is: ", end="")
print(output1.hexdigest())

input2=input("Enter text/image you want to check for tampering:: ")
output2=hashlib.md5(input2.encode()) #encode string into bytes
print("The hashing value is : ", end="")
print(output2.hexdigest())

if(output1.hexdigest()==output2.hexdigest()):
  print("Text/Image not tampered")
else:
  print("Text/Image tampered")
```

5 Conclusion

Hence, we learned hashing algorithms. We also learned how to implement them in python. We also learned how to check for tampering of data.

6 FAQ's

6.1 *What is MAC? What is the difference between MAC and message digest.*

MAC (Message Authentication Code) is a cryptographic technique used to verify the authenticity and integrity of a message. It involves generating a unique code, or tag, based on a secret key and the message being authenticated. The receiver of the message can then verify the authenticity of the message by recomputing the MAC using the same key and comparing it with the received MAC.

The key difference between MAC and message digest (such as MD5 or SHA) is that MAC involves the use of a secret key, while message digest does not. Message digest generates a fixed-length hash value that can be used to verify the integrity of a message, but it cannot be used to authenticate the message or to verify its origin.

MAC, on the other hand, uses a secret key to generate a tag that is unique to both the message and the key. This tag can be used to verify not only the integrity of the message but also its authenticity and origin. MAC provides stronger security than message digest because it uses a secret key, making it much more difficult for an attacker to tamper with the message or generate a fake MAC.

In summary, the main difference between MAC and message digest is that MAC uses a secret key to generate a unique tag that can be used for both authentication and integrity verification, while message digest generates a fixed-length hash value that can only be used for integrity verification.

6.2 *Compare MD5 and SHA-1.*

MD5 and SHA-1 are both cryptographic hashing algorithms used to generate a unique digital fingerprint, or hash, for a given message or data. However, there are some differences between the two:

1. Output size: MD5 generates a 128-bit hash value, while SHA-1 generates a 160-bit hash value. This means that SHA-1 has a larger output size and is generally considered to be more secure than MD5.
2. Collision resistance: MD5 is vulnerable to collision attacks, where two different messages can have the same hash value. SHA-1 is also vulnerable to collision attacks, but it is much more resistant than MD5.
3. Speed: MD5 is generally faster than SHA-1, making it more suitable for applications that require faster hash generation. However, the difference in speed is not significant in most cases.
4. Security: Both MD5 and SHA-1 are now considered to be insecure and should not be used for new applications. MD5 has some known security weaknesses that make it vulnerable to certain types of attacks, while SHA-1 has been shown to be vulnerable to more advanced attacks as computing power has increased.

In summary, SHA-1 is generally considered to be more secure than MD5 due to its larger output size and stronger collision resistance. However, both algorithms are now considered to be insecure and should not be used for new applications. It is recommended to use newer hashing algorithms such as SHA-256 or SHA-3 for better security.

6.3 *Compare various versions of SHA.*

There are several versions of the Secure Hash Algorithm (SHA) that have been developed by the National Institute of Standards and Technology (NIST) for various purposes. Here's a brief comparison of the most commonly used versions:

- SHA-1: This is the first version of SHA and generates a 160-bit hash value. It has been shown to be vulnerable to collision attacks and is no longer recommended for use.

- SHA-2: This is a family of hash functions that includes SHA-224, SHA-256, SHA-384, and SHA-512, which generate hash values of 224, 256, 384, and 512 bits respectively. SHA-2 is widely used and considered to be secure, but it is slower than SHA-1 and requires more processing power.
- SHA-3: This is the latest version of SHA, also known as Keccak. It was designed as a response to the vulnerabilities found in SHA-2 and generates hash values of 224, 256, 384, and 512 bits like SHA-2. SHA-3 is faster and more secure than SHA-2, but it is not yet as widely adopted.

In summary, SHA-1 is no longer recommended due to its vulnerabilities, while SHA-2 is widely used and considered to be secure. SHA-3 is the newest version of SHA and offers better security and speed than SHA-2, but it is not yet as widely adopted. The choice of which version of SHA to use depends on the specific requirements of the application and the level of security needed.