



Name: Devanshu Surana

Roll No.: 23

Prn:1032210755

Panel: C batch:C1

Lab A5: Implementation of Integrity of messages using MD5 or SHA

Objective of Lab

1. To understand and implement hashing algorithm MD5 and SHA
2. To generate hash value using MD5 and SHA and understand how Hash values have a variety of uses and purposes and originally were important for cryptography for concerning secure or coded communications and integrity of message.

Theory

MD5 (Message Digest Algorithm 5) and SHA (Secure Hash Algorithm) are both cryptographic hash functions, but they have key differences in terms of security and usage. Here's a brief overview of each:

MD5 (Message Digest Algorithm 5):

Description: MD5 is a widely used hash function that produces a 128-bit (16-byte) hash value, typically expressed as a 32-character hexadecimal number.

Usage: MD5 was commonly used for integrity checking and to create digital signatures. However, it is now considered broken for cryptographic purposes due to vulnerabilities that allow for collision attacks (different inputs producing the same hash).

Security Concerns: MD5 is considered insecure for cryptographic purposes, and it's not recommended for security-sensitive applications. Its vulnerabilities led to the development of more secure hash functions like the SHA series.

SHA (Secure Hash Algorithm):

Description: SHA is a family of cryptographic hash functions designed by the National Security Agency (NSA). The most commonly used versions are SHA-1, SHA-256, SHA-384, and SHA-512, each producing hash values of different lengths (in bits).

Usage:



SHA-1: While still used in some applications, SHA-1 is deprecated for cryptographic purposes due to vulnerabilities. It's recommended to use stronger variants like SHA-256 or SHA-3.

SHA-256, SHA-384, SHA-512: These are more secure and commonly used in various security applications, including digital signatures, certificate generation, and integrity checking.

Security Concerns: While SHA-1 is considered insecure for cryptographic use, SHA-256 and higher variants are currently considered secure. However, the security community continually evaluates and updates cryptographic standards as new vulnerabilities and technologies emerge.

In summary, MD5 is generally considered insecure for cryptographic applications due to vulnerabilities, while SHA-1 is deprecated for similar reasons. The SHA-2 family, which includes SHA-256, SHA-384, and SHA-512, is widely used and considered secure. Additionally, SHA-3, a separate hash function family, is also available as a newer alternative. When choosing a hashing algorithm, it's essential to consider the specific security requirements of the application and use the most appropriate and up-to-date algorithm.

Program

```
import hashlib

data = "Hello, MD5!"

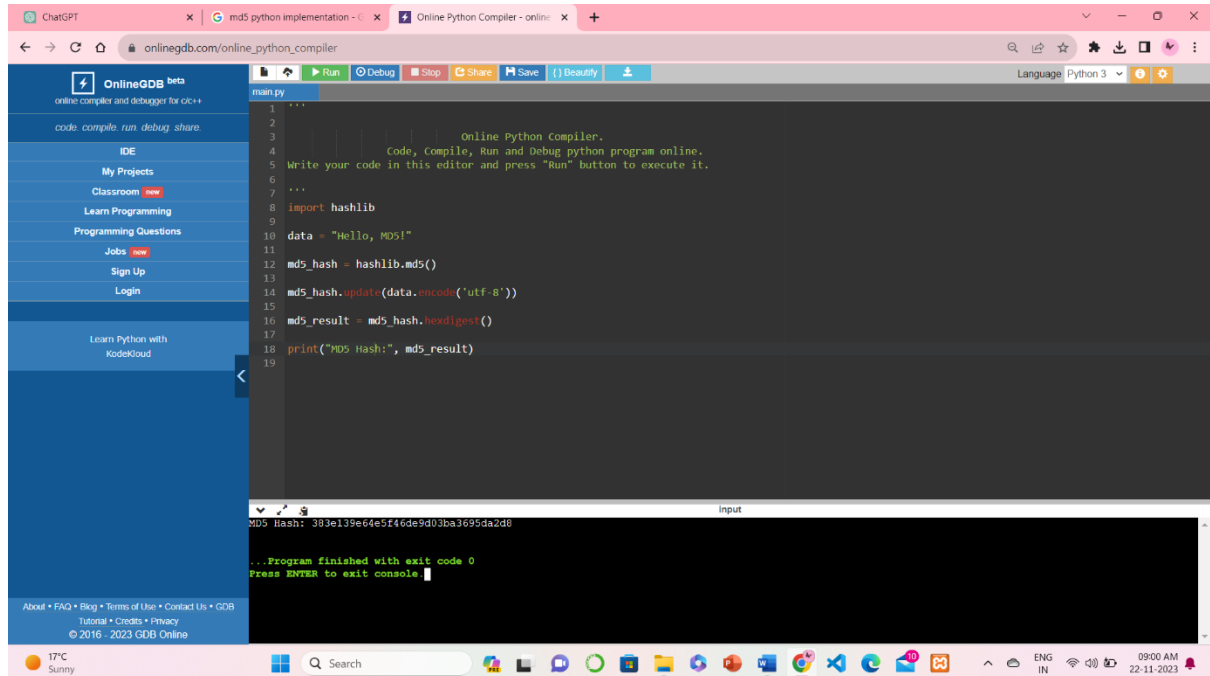
md5_hash = hashlib.md5()

md5_hash.update(data.encode('utf-8'))

md5_result = md5_hash.hexdigest()

print("MD5 Hash:", md5_result)
```

Output Screen shots (



```
1 '''
2 Online Python Compiler.
3 Code, Compile, Run and Debug python program online.
4 Write your code in this editor and press "Run" button to execute it.
5 '''
6
7 import hashlib
8
9 data = "Hello, MD5!"
10
11 md5_hash = hashlib.md5()
12 md5_hash.update(data.encode('utf-8'))
13
14 md5_result = md5_hash.hexdigest()
15
16 print("MD5 Hash:", md5_result)
```

MD5 Hash: 383e139e64e5f46de9d03ba3695da2d8

...Program finished with exit code 0
Press ENTER to exit console

Conclusion: Thus we have generated hash value using MD5 and SHA and learned how Hash values have a variety of uses and purposes and originally were important for cryptography for concerning secure or coded communications and integrity of message.

FAQs:

1. List down some Hashing Algorithms
2. What is the MD5 message-digest algorithm?
3. What are Alternatives to MD5 algorithm
4. Difference between MD5 and SHA algorithm

References: