Sipna College of Engineering & Technology, Amravati.
Department of Computer Science & Engineering
Session 2022-2023

Branch :- Computer Sci. & Engg. **Class :- Final Year**
Subject :-Block Chain Fundamentals Lab manual **Sem :- VII**
Teacher Manual

PRACTICAL NO 3

AIM: To Understand and implement various SHA algorithms

S/W REQUIRED: Phython

SHA (Secure Hash Algorithms)

SHA stands for secure hashing algorithm. SHA is a modified version of MD5 and used for hashing information and certificates. A hashing algorithm shortens the input information into a smaller form that cannot be learned by utilizing bitwise operations, modular additions, and compression functions.

SHAs also help in revealing if an original message was transformed in any way. By imputing the original hash digest, a user can tell if even an individual letter has been shifted, as the hash digests will be effectively different.

The important element of SHAs are that they are deterministic. This define that consider the hash function used is known, any computer or user can regenerate the hash digest. The determinism of SHAs is one of main reasons that each SSL certificate on the Internet is needed to have been hashed with a SHA-2 function.

SHA, (Secure Hash Algorithms) are set of cryptographic hash functions defined by the language to be used for various applications such as password security etc. Some variants of it are supported by Python in the “hashlib” library. These can be found using “algorithms_guaranteed” function of hashlib.

SHA Hash
The different SHA hash functions are explained below.

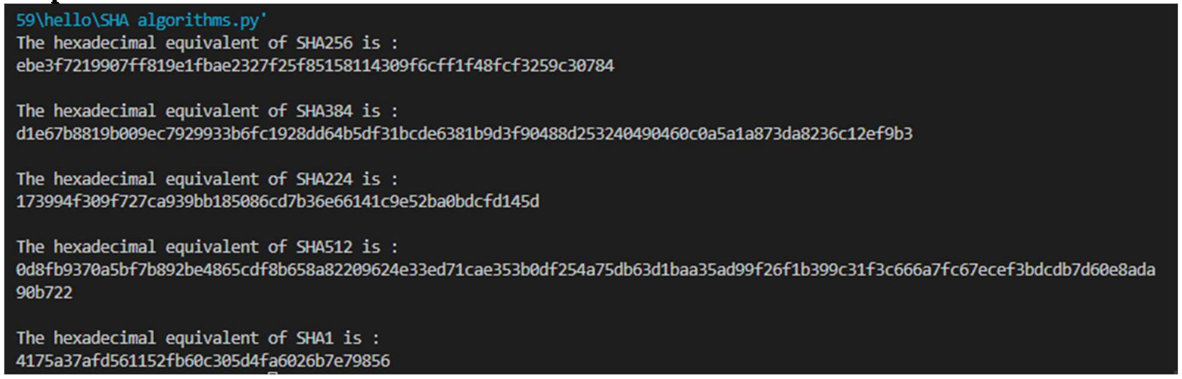
1. SHA256 : This hash function belong to hash class SHA-2, the internal block size of it is 32 bits.
2. SHA384 : This hash function belong to hash class SHA-2, the internal block size of it is 32 bits. This is one of the truncated version.
3. SHA224 : This hash function belong to hash class SHA-2, the internal block size of it is 32 bits. This is one of the truncated version.
4. SHA512 : This hash function belong to hash class SHA-2, the internal block size of it is 64 bits.
5. SHA1 : The 160 bit hash function that resembles MD5 hash in working and was discontinued to be used seeing its security vulnerabilities.

Implementation:

```
# Python 3 code to demonstrate
# SHA hash algorithms.

import hashlib

# initializing string
str = "SIPNA COET"

# encoding SIPNA COETusing encode()
# then sending to SHA256()
result = hashlib.sha256(str.encode())

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of SHA256 is : ")
print(result.hexdigest())

print ("\r")

# initializing string
str = "SIPNA COET"

# encoding SIPNA COETusing encode()
# then sending to SHA384()
result = hashlib.sha384(str.encode())

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of SHA384 is : ")
print(result.hexdigest())

print ("\r")

# initializing string
str = “SIPNA COET”

# encoding SIPNA COETusing encode()
# then sending to SHA224()
result = hashlib.sha224(str.encode())

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of SHA224 is : ")
print(result.hexdigest())

print ("\r")

# initializing string
str = “SIPNA COET”

# encoding SIPNA COETusing encode()
# then sending to SHA512()
result = hashlib.sha512(str.encode())
```

```
# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of SHA512 is : ")
print(result.hexdigest())

print ("\r")

# initializing string
str = “SIPNA COET”

# encoding SIPNA COETusing encode()
# then sending to SHA1()
result = hashlib.sha1(str.encode())

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of SHA1 is : ")
print(result.hexdigest())
```

Output:



```
59\hello\SHA algorithms.py'
The hexadecimal equivalent of SHA256 is :
ebe3f7219907ff819e1fbae2327f25f85158114309f6cff1f48fcf3259c30784

The hexadecimal equivalent of SHA384 is :
d1e67b8819b009ec7929933b6fc1928dd64b5df31bcde6381b9d3f90488d253240490460c0a5a1a873da8236c12ef9b3

The hexadecimal equivalent of SHA224 is :
173994f309f727ca939bb185086cd7b36e66141c9e52ba0bdcfd145d

The hexadecimal equivalent of SHA512 is :
0d8fb9370a5bf7b892be4865cdf8b658a82209624e33ed71cae353b0df254a75db63d1baa35ad99f26f1b399c31f3c666a7fc67ecef3bdcdb7d60e8ada
90b722

The hexadecimal equivalent of SHA1 is :
4175a37afd561152fb60c305d4fa6026b7e79856
```

CONCLUSION: Thus we have studied and implemented various SHA algorithms.