International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Study of SHA Algorithm in Cryptography

Soe Moe Myint[1], Moe Moe Myint[2], Aye Aye Cho[3]

[1]Lecturer, Faculty of Computer Systems Technologies, University of Computer Studies, Pathein, Myanmar
[3]Associate Professor, Faculty of Computer Science, University of Computer Studies, Hinthada, Myanmar
[2]Lecturer, Information Technology Support and Maintenance,
University of Computer Studies, Pathein, Myanmar

How to cite this paper: Soe Moe Myint | Moe Moe Myint | Aye Aye Cho "A Study of SHA Algorithm in Cryptography" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.1453-1454, **https://doi.org/10.31142/ijtsrd26680**

IJTSRD26680

ABSTRACT

Today, security is important on the network. Therefore, the security is provided by the nature of one way functions, which is a key component of SHA (Secure Hash Algorithms). The purpose of this paper how to use SHA-256 and SHA-512 from SHA-2 alogrithms. SHA-2 is a family of two similar hash functions with different block sizes known as SHA -256 and SHA-512. They differ in the word size;SHA-256 uses 32-bits words where SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-223, SHA-384, SHA-512/224 and SHA-512/256. These were also designed by the NAS.

KEYWORDS: *hash file; SHA-256; SHA-512; John the Ripper tool*

INTRODUCTION

Cryptographic hash functions are utilized in order to keep data secured by providing three fundamental safety characteristics: pre-image resistance, second pre-image resistance, and collision resistance. The cornerstone of cryptographic security lies in the provision of pre-image resistance, which makes it hard and time-consuming for an attacker to find an original message given the respective hash value. This security is provided by the nature of one way functions, which is a key component of SHA.

SHA-256 is faster on 32-bit processors. SHA-512 is faster on 64-bit processors. SHA-512 has 25% more rounds than SHA-256. SHA-256 performs 64 rounds of its compression function over 512 bits at a time. SHA-512 performs 80 rounds of the compression function over 512 bits a time. SHA-512 beats. SHA-256 for hashing anything more than 16 bytes of data at a time.

BACKGROUND THEORY

SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the United States National Security Agency (NAS). They are built using the Merkle–Damgård structure, from a one-way compression function itself built using the Davies–Meyer structure from a (classified) specialized block cipher. SHA-256 and SHA-512 are novel hash functions computed with 32-bit and 64-bit words, respectively. They use different shift amounts and additive constants, but their structures are otherwise virtually identical, differing only in the number of rounds.[1]

John the Ripper (Dictionary Attack Method) John the Ripper is the free password cracking software tool. Initially developed for Unix operating system. It is run in different platform such as DOS, Win32, BeOS.

Fig –John the Ripper tool usage

SHA-256 Algorithm

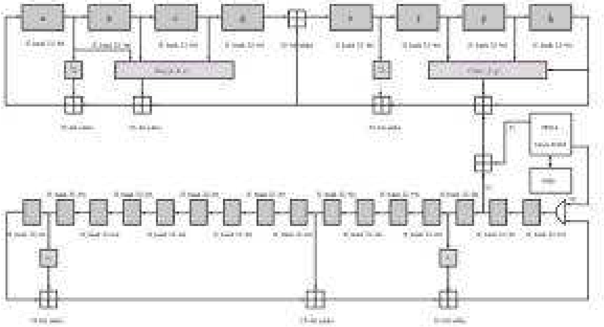
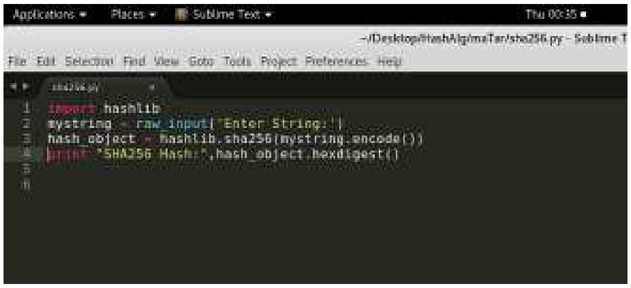


Fig- SHA-256 processes in Algorithm

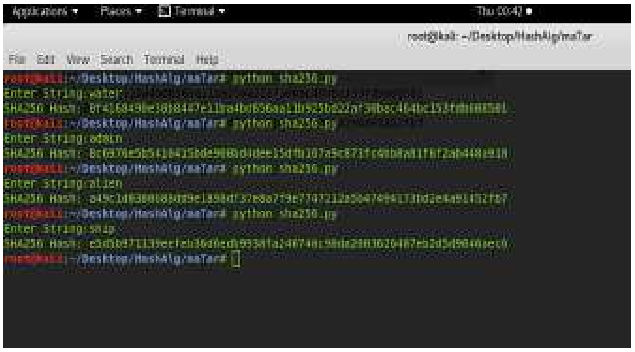
A. Using Python Program to generate Hash form in SHA-256


Applications ▾
Places ▾
Sublime Text ▾
Thu 00:35
~/Desktop/HashAlg/maTar/sha256.py - Sublime T
File Edit Selection Find View Goto Tools Project Preferences Help
sha256.py

```
import hashlib
mystring = raw_input('Enter String:')
hash_object = hashlib.sha256(mystring.encode())
print "SHA256 Hash:",hash_object.hexdigest()
```

Fig2 Python code for execute hash file

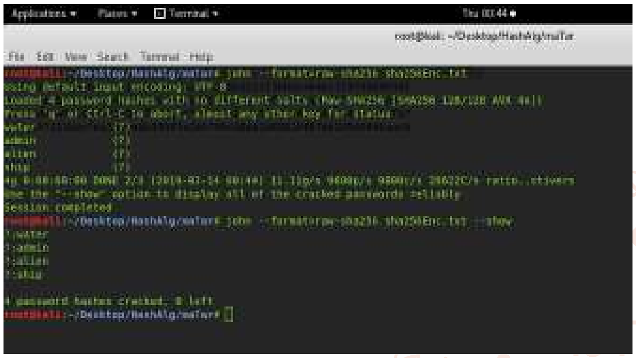
B. Generation of Hash File for SHA-256



```
root@kali:~/Desktop/HashAlg/maTar# python sha256.py
Enter String:water
SHA256 Hash: 0f4169490e30b8447e11ba4bd856aa11b925b022af30bec464bc153fdb688581
root@kali:~/Desktop/HashAlg/maTar# python sha256.py
Enter String:admin
SHA256 Hash: 8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
root@kali:~/Desktop/HashAlg/maTar# python sha256.py
Enter String:alien
SHA256 Hash: a49c1d8380880d9e1898df37e8a7f9e7747212a5047494173bd2e4a91452fb7
root@kali:~/Desktop/HashAlg/maTar# python sha256.py
Enter String:ship
SHA256 Hash: e5d5b97113eefeb36d6edb993ffa246740c98da2003620467eb2d5d9048aec0
root@kali:~/Desktop/HashAlg/maTar# 
```

Fig3 Hex code with hash file in SHA-256

C. After Generation of Hash file by using John the Ripper's brute force (directory attack method) in SHA-256



```
root@kali:~/Desktop/HashAlg/maTar# john --format=raw-sha256 sha256Enc.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-SHA256 [SHA256 128/128 AVX 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
water            (?)
admin            (?)
alien            (?)
ship             (?)
4g 0:00:00:00 DONE 2/3 (2019-03-14 00:44) 11.11g/s 9600p/s 9600c/s 28622C/s ratto..ctivers
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/HashAlg/maTar# john --format=raw-sha256 sha256Enc.txt --show
?:water
?:admin
?:alien
?:ship

4 password hashes cracked, 0 left
root@kali:~/Desktop/HashAlg/maTar# 
```

Fig-4 Using John the Ripper to decrypt the hash file

SHA-512 Algorithm

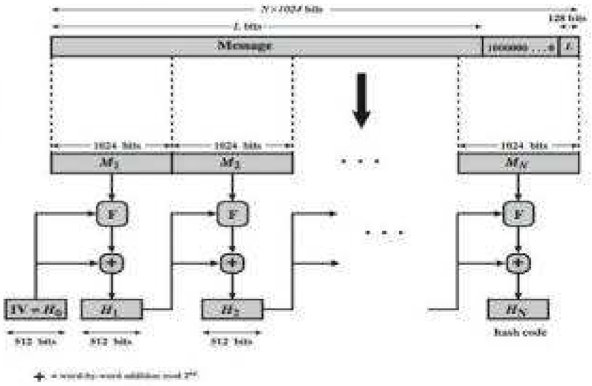
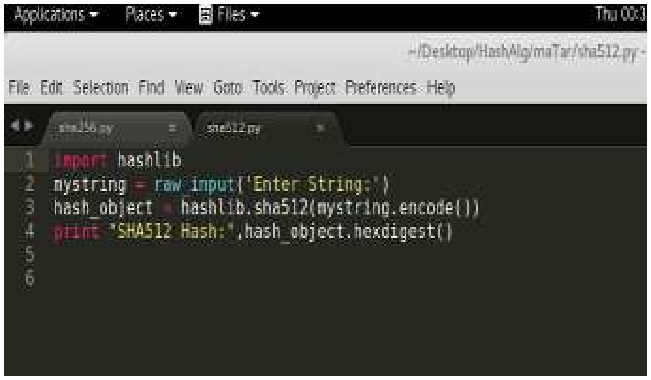


Fig-5 SHA-512 processes in Algorithm

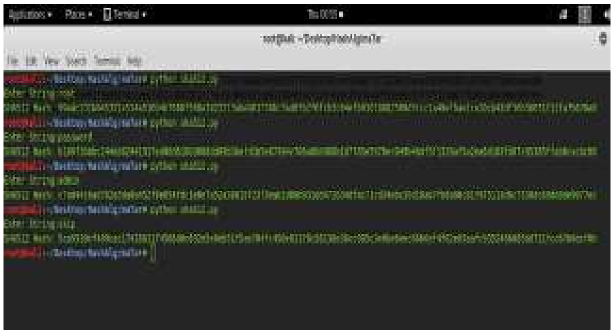
A. Using Python Program to generate Hash form in SHA-512



```
import hashlib
mystring = raw_input('Enter String:')
hash_object = hashlib.sha512(mystring.encode())
print "SHA512 Hash:",hash_object.hexdigest()
```

Fig.6 Python code to execute hash file in SHA-512

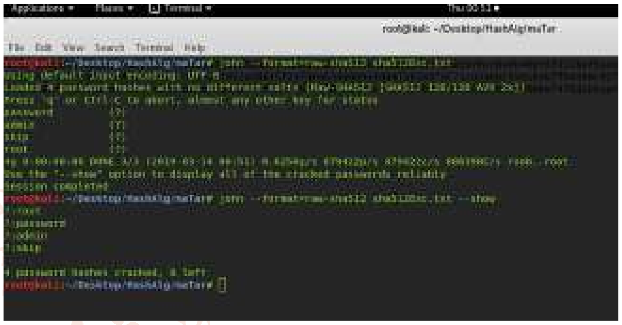
B. Generation of Hash file for SHA-512



```
root@kali:~/Desktop/HashAlg/maTar# python sha512.py
Enter String:root
SHA512 Hash: [illegible]
root@kali:~/Desktop/HashAlg/maTar# python sha512.py
Enter String:password
SHA512 Hash: [illegible]
root@kali:~/Desktop/HashAlg/maTar# python sha512.py
Enter String:admin
SHA512 Hash: [illegible]
root@kali:~/Desktop/HashAlg/maTar# python sha512.py
Enter String:ship
SHA512 Hash: [illegible]
root@kali:~/Desktop/HashAlg/maTar# 
```

Fig-7 Hex code with hash file in SHA-512

C. After Generation of Hash file by using John the Ripper's brute force (directory attack method) in SHA-512



```
root@kali:~/Desktop/HashAlg/maTar# john --format=raw-sha512 sha512Enc.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-SHA512 [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (?)
admin            (?)
ship             (?)
root             (?)
4g 0:00:00:00 DONE 3/3 (2019-03-14 00:51) 8.425g/s 879422p/s 879422c/s 8803906C/s rook..root
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/HashAlg/maTar# john --format=raw-sha512 sha512Enc.txt --show
?:root
?:password
?:admin
?:ship

4 password hashes cracked, 0 left
root@kali:~/Desktop/HashAlg/maTar# 
```

Fig- 8 Using John the Ripper tool to decrypt the hash file

Conclusion

This paper is to study how to use the function of SHA-256 and SHA-512. And then, the tool of John the Ripper (Dictionary Attack Method) usage in the Linux OS. In this paper, the reader can understand easily to use the linux command and procedure of the steps in the figures. So we hope to be useful of the knowledge about the SHA-256 and SHA-512 in linux operating system.

References

[1] https://en.wikipedia.org/wiki/SHA-2

[2] Behrouz A.Forouzan, Cryptography and Network Security, McGraw-Hill International edition, 2008.

[3] Cryptography & Network Security (project_paper) , University of Computer Studies, Pathein, Myanmar,2018

[4] https://www.researchgate.net/publication/325581582_Introduction_to_Secure_Hash_Algorithms

[5] Simmons GJ. Message Authentication with arbitration of transmitter/receiver disputes. Advances in Cryptology- Eurocrypt'87, Lecture Notes in Computer Science, Springer-Verlag, Berlin; 1988; 304: 151-165.

[6] Harshvardhan Tiwari. A Secure Hash Function MD-192 with Modified Message Expansion" Vol. 7 No. 2 February 2010 International Journal of Computer Science and Information Security.

[7] https://www.schneier.com/blog/archives/2005/08/new_cryptanalyt.html [Accessed : Oct. 7, 2014]