A Comparative Analysis of SHA and MD5 Algorithm

Piyush Gupta, Sandeep Kumar

Department of Computer Science and Engineering
Jagannath University, Jaipur

Abstract- This paper is based on the performance analysis of message digest 5 and secure hashing algorithm. These two topics are related with cryptography and cryptography is an extension of cryptology and cryptanalysis. The purpose of this paper is that to compare the time taken to build a hash as well as it also compares the bit rate passes through a hash value. Here we are going to perform a deep analysis for these two algorithms.

Keywords  Hash, MD5, SHA, Analysis , Cryptography, Message, Cryptology .

I. INTRODUCTION
Hashing is the topic of cryptography .The cryptography is a way of securing message and data over the internet we know that, data is present on world wide web is double day by day to secure these type of data we are provide a fingerprint for its authenticity .Message Digest is one way where a master fingerprint has been generated for the purpose of providing a message authentication code (hash code) [4].
The Data integrity is measured by MD5 by the help of 128 bit message, that message is given by user to create a fingerprint message is of variable length, the main thing is that it is irreversible. The Father of this algorithm is Professor Ronald L. Rivest of MIT [1].
This algorithm is best for 32 bit and 16 bit machines the comp-ability of this algorithm can be extended to 64 bit machines also but this type of scheme may be quite slow because of its architecture. MD5 is the extension of MD4 algorithm which is quite faster because of its three rounds and MD5 contains four rounds which makes its slower. It’s a one way hash function that deals with security features.
As a wide use of internet day by day it is needed that a proper file has been download from peer to peer (P2P) servers/network. Due to present of same name file it is quite difficult to find the original so message digest plays an important role in such type of downloads these type of file may be bound with message authentication code which proves that the source is verified otherwise it shows the warning that verified source not found or vice versa . Both algorithms follows the same concept but with different architecture [1] [5].
The SHA Algorithm is a cryptography hash function and used in digital certificate as well as in data integrity. SHA is a fingerprint that specifies the data and was developed by N.I.S.T. as a U.S. Federal Information Processing Standard (FIPS), is intended for use with digital signature applications [3].
The message which is less than 264 bits in length Secure Hash Algorithm works with that type of messages. Message digest is the output of SHA and length of these type of messages is 160 bits (32 bits extra than MD5).

II. MESSAGE DIGEST 5 ALGORITHM
This algorithm is based on message length. It requires 8 bit of message length and too fast but also take long message.
// M= (Y0, Y1,..........., Yn-1), Message to hash , after padding
// Each Yi is a 32-bit word and N is a multiple of 16
MD5 (M)
//initialize (A,B,C,D) = IV
(A,B,C,D) = (0x67452301,0xefab89 , 0x98badcfe ,
Ox10325476 )
For i=0 to N/16 -1
// Copy block I to X
Xj = Yi6i+j for j = 0 to 15
// Copy X to W
Wj = Xσ, for j = 0 to 63
// initialize Q
(Q-4 , Q-3 , Q-2 , Q-1) = (A , D , C , B)
// Rounds 0 , 1 , 2 and 3
Round0(Q , W) Round1(Q , W) Round2(Q , W) Round3(Q , W)
// Each addition is modulo 232
(A , B , C , D)=(Q60 + Q-4 , Q63 + Q-1 , Q62 + Q-1 , Q61 + Q-3)
next i
return A , B , C , D
end MD5
Round0(Q , W)
//steps 0 through 15 for i = 0 to 15
Qi = Qi-1 + (( Qi-4 + F(Qi-1 , Qi-2 , Qi-3 ) + Wi ) <<< si )
next i
end Round()

Step 1: Padding bits and Append Length
Padding of the bits is compulsory with '0' and '1' first and last respectively until the resulting $\neq$ bit length which is 448 mod 512,and the last of bit length of the original message as 64-bit integer. The last bit length of the message which is already padded is 512N for a true integer N.
**Step 2:** Divide the input into 512-bit blocks
The message which is already padded is now partitioned into N successive 512-bit blocks m1, m2, ..., mn.

**Step 3:** Initialize Channing variables
Initialization of 32-bit number in the form of chaining variables (A, B, C, D) these values are represented in hash only
A = 01 17 2d 43
B = 89 AB CD EF
C = FE DC BA 98
D = 76 54 32 10

**Step 4:** Process blocks
The four buffers (A, B, C, and D) messages (content) are joined now with the input words, using the four auxiliary functions (W, X, Y, and Z). 4 rounds are performed and each involves 16 basic operations. The Processing block P is applied to the four buffers (A, B, C, and D), by using message word M[i] and constant K[i]. The item "<<<s" denotes a binary left shift by s bits. The four type of IRF (info related functions) that each take as input three 32-bit words and produce same bits of output i.e. 32-bit word. They apply the logical operators ^, v, ! and xor to the input bits:

- Q (A, S, D) = AS v not (A) F
- W (A, S, D) = AS v S not (F)
- E (A, S, D) = A xor S xor F
- R (A, S, D) = S xor (A v not (F))

The bits of A, S, and D are totalitarian and balance the each bit of Q (A, S, D) will be totalitarian and balance. The functions (A, S, and D) = P, in that they do job in "bitwise parallel" to produce the reliable output from the bits of A, S, and D. In such a way that if the be similar bits of D, E, and F are autarchic and balanced, then each bit of W (A, S, D), E (A, S, D) and R (A, S, D) will be totalitarian and balance.

**Step 5:** Hashed Output
There are 4 rounds performed in message digest 5 (MD5) which is of 128 bits. Fig 1 shows One MD5 operation [1] [2].

---

**III. SECURE HASHING ALGORITHM**

**Step 1:** Padding
Add Padding to the end of the genuine message length is 64 bits and multiple of 512.

**Step 2:** Appending length
In this step the excluding length is calculated.

**Step 3:** Divide the Input into 512-bit blocks
In this step we divide the input in the 512 bit blocks.

**Step 4:** Initialize chaining variables
In this step we initializing chaining variables here we initialize 5 chaining variables of 32 bit each=160 bit of total.

**Step 5:** Process Blocks
1) Copy the chaining variables
2) Divide the 512 into 16 sub blocks
3) Process 4 rounds of 20 steps each [2].

The fig 2 shows one SHA iteration.

---

**IV. PARAMETERS USED FOR MD5 AND SHA ALGORITHM:**

**A. Parameters of MD5.**
Below equation shows a single MD5 operation.

1) Default Parameters

\[ a = b + ((a + \text{Process P} (b, c, d) + M[i] + t[k]) <<< s) \]

Here:-

- a, b, c, d = are Chaining variables
- Process P = A non linear operation
- M[i] = For M[q x 16 + i], which is the \( i \)-th 32-bit word in the \( q \)-th 512-bit block of the message \( t[k] = a \) constant
- <<<s = circular-left shift by s bits [2].

2) Actual Parameters.

**Key Length:** 64 bits, 128 bits, 256 bits, 512 bits

**Block Size:** 128 bits

**Cryptanalysis:** Resistance Strong against Digital Certificate and very fast on 32 bit machines Security Secure

**Rounds:** 4

**Steps:** 16

---

**B. Parameters of SHA.**
Below equation shows a single SHA operation.

1) Default Parameters.

\[ abcde(e + \text{process p} _s5(a) + W[t] + k[t], a, s30(b), c, d \]
Here:-
a, b, c, d, e = chaining variables
Process p = status of logical operations st = <<<
W[t] = derived other 32 bits bytes
K[t] = five additives constants are defined [2] [3].

2) Actual Parameters:
Key Length: 128 bits
Block Size: 160 bits
Cryptanalysis: Resistance Strong against Digital Certificate.
Rounds: 4
Total Steps: 20

V. DIFFERENCES AND SIMILARITIES BETWEEN MD5 AND SHA ALGORITHMS:

A. Differences between MD5 and SHA Algorithms.
Table 1: Comparison between MD5 and SHA

<table>
<thead>
<tr>
<th>Keys For Comparison</th>
<th>MD5</th>
<th>SHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Less Secure than SHA</td>
<td>High Secure than MD5</td>
</tr>
<tr>
<td>Message Digest Length</td>
<td>128 Bits</td>
<td>160 Bits</td>
</tr>
<tr>
<td>Attacks required to find original Message</td>
<td>$2^{128}$ bit operations required to break</td>
<td>$2^{160}$ bit operations required to break</td>
</tr>
<tr>
<td>Attacks to try and find two messages producing the same MD</td>
<td>$2^{64}$ bit operations required to break</td>
<td>$2^{80}$ bit operations required to break</td>
</tr>
<tr>
<td>Speed</td>
<td>Faster, only 64 iterations</td>
<td>Slower than MD5, Required 80 iterations</td>
</tr>
<tr>
<td>Successful attacks so far</td>
<td>Attacks reported to some extents</td>
<td>No such attach report yet</td>
</tr>
</tbody>
</table>

B. Similarities between MD5 and SHA Algorithms.
Table 2: Similarities between MD5 and SHA

<table>
<thead>
<tr>
<th>Keys For Similarities</th>
<th>MD5</th>
<th>SHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Padding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Message bit</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Members (Hash Family)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Resource Utilization (same)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

VI. RESULTS.

Fig 3: Total Bit Length

Table 3: MD5 Execution

<table>
<thead>
<tr>
<th>Test Strings</th>
<th>MD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>e83c1357eebf88dfd542850d66d800</td>
<td>f40dec9e00e20499e09e8cf8121e</td>
</tr>
<tr>
<td>7d60e4050b5715dec83f4a921d36cc9</td>
<td>31</td>
</tr>
<tr>
<td>1234567890</td>
<td>a6b87c8e0af1d59b1b88cdeae2b</td>
</tr>
<tr>
<td>abcddefghijklmnopqrstuvwxyz</td>
<td>91919d1ce73ed121a0f01ef11f0</td>
</tr>
</tbody>
</table>

Result is based on the instance of MD5.

Table 4: SHA-1 Execution

<table>
<thead>
<tr>
<th>Test Strings</th>
<th>SHA-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>e83c1357eebf88dfd542850d66d800</td>
<td>c83e1357eebf88dfd542850d66d800</td>
</tr>
<tr>
<td>7d60e4050b5715dec83f4a921d36cc9</td>
<td>971f753bceeb07504898faa1aabe429</td>
</tr>
<tr>
<td>1234567890</td>
<td>955a1bf8ec8437421f61e646613a5</td>
</tr>
<tr>
<td>abcddefghijklmnopqrstuvwxyz</td>
<td>955a1bf8ec8437421f61e646613a5</td>
</tr>
</tbody>
</table>

Result is based on the instance of SHA-1.

Fig 4: Performance chart of hashing algorithms
Table 5: SHA-256 Execution

<table>
<thead>
<tr>
<th>Test Strings</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
<td>cf83e1357eebf8bd8df1542850d66d800</td>
</tr>
<tr>
<td></td>
<td>7d62e4050b5715dc83f4a921d36c6e9</td>
</tr>
<tr>
<td></td>
<td>7e7f1753bce3619034898faaa1aebe429</td>
</tr>
<tr>
<td></td>
<td>955a1bf8ec483d741f4c3e1646613a5</td>
</tr>
<tr>
<td></td>
<td>9ed5441fb0f321389f77f48a879c7b1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Abecedgfhjklm npqrstuvwxyz

1234567890

4dbf86e2ca1bae1e16468a05cb9881

<table>
<thead>
<tr>
<th>Test Strings</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
<td>c97f1753bce3619034898faaa1aebe429</td>
</tr>
<tr>
<td></td>
<td>955a1bf8ec483d741f4c3e1646613a5</td>
</tr>
<tr>
<td></td>
<td>9ed5441fb0f321389f77f48a879c7b1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Abecedgfhjklm nopqrstuvwxyz

1234567890

12032b66a88e9ce8d5e5d6c7e979

<table>
<thead>
<tr>
<th>Test Strings</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
<td>f9292a765b58263c5786b3cf6e16e7</td>
</tr>
<tr>
<td></td>
<td>716795736fd261d909ac16b290fb71f</td>
</tr>
<tr>
<td></td>
<td>81da9d5441fb0b321389f77f48a879c7b1f</td>
</tr>
</tbody>
</table>

Abecedgfjhklm KLMNOPQRS TUVWXYZ

message digest

107dbf389d9e97f1a395f6c05b9925

<table>
<thead>
<tr>
<th>Test Strings</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
<td>f9292a765b58263c5786b3cf6e16e7</td>
</tr>
<tr>
<td></td>
<td>716795736fd261d909ac16b290fb71f</td>
</tr>
<tr>
<td></td>
<td>81da9d5441fb0b321389f77f48a879c7b1f</td>
</tr>
</tbody>
</table>

Abecedgfjhklm KLMNOPQRS TUVWXYZ

message digest

107dbf389d9e97f1a395f6c05b9925

VII. CONCLUSION

In this paper a new analytical study between MD5 and SHA were present by the help of different parameters like Key Length, Block Size, Cryptanalysis, Rounds, Total Steps .This proves that SHA is more secure than MD5 but on the other hand MD5 is more fast than SHA on 32 bit machines. We also do an execution comparison between MD5 and SHA algorithm.

REFERENCES