IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.7, Special Issue.2, Jan 2019 E-ISSN: 2347-2693

Comparative Analysis of Hashing Algorithms used in Data Deduplication

J. Maria Selvam[1*], P. Srivaramangai[2]

[1,2]Dept.of Computer Science, Maruthu pandiyar College of Arts & Science, Thanjavur, Tamilnadu, India

Available online at: www.ijcseonline.org

Abstract— Now-a-days, increasing volume of digital data demands more storage space and efficient technique to handle these data. Duplicate is inevitable while handling huge amount of data. Data Deduplication is an efficient approach in storage environment that utilize different techniques to deal with duplicate data. It identifies and removes redundant data by its message digest value which is generated by using any kind of cryptographic hash algorithms such as MD5 and Secure hash algorithms. This research paper analyses the two hash algorithms, namely MD5 and SHA, based on different parameters and compare the time taken to build a hash value.

***Keywords*—** Data Deduplication, Chunking, Hash Function, MD5, SHA-1, SHA-2.

I. INTRODUCTION

Data deduplication and data compression are the most prominent techniques used among the available techniques for optimal storage. Data deduplication is one of the intelligent compression techniques which eliminate duplicate data by storing only a single copy of each file or block. Duplicate data is replaced by means of a pointer to the unique copy of data.

There are two main strategies used in data deduplication storage systems namely Delta-based Deduplication and Hash-based or Finger printing-based Deduplication. Delta-based deduplication techniques are used in incremental backups. During the incremental backups, delta encoding algorithm computes the difference between the old and new versions of a file. The differences obtained are called delta value. This delta values are stored along with a pointer which points to the old version of a file. [4-delta related paper]

Hash-based or Finger printing based deduplication method uses a cryptographic hashing function which generates a hash value to find and remove redundant data.

II. CRYPTOGRAPHIC HASH FUNCTION

A cryptographic hash function is a deterministic procedure that transforms any arbitrary size of message into fixed size hash value or message digest. It is a one-way hash function that means it's easy to compute hash value but it is difficult to find the original text from the hash value. Most commonly used algorithms to execute the hash function are MD-4, MD-5, SHA-0, SHA-1 and SHA-2. Cryptographic hash function is used to create digital signatures, message authentication codes (MACs), and other forms of authentication.

2.1 MD5 Algorithm

MD5 stands for Message Digest Algorithm 5.It is one of the most widely used cryptographic hashing algorithm in the field of authentication security[8].It was introduced by professor Ronald Rivest [6] in the year 1992.It is an improved version of MD4.It takes various length of input and generates fixed length of 128 bit as output. MD5 has less CPU intensive and less complex than other hash functions. For instance, where MD5 is commonly used approach in the general non-top-secret applications. The weakness of MD5 is reported vulnerable to a hash collision[10]. This weakness permit the malicious users can create multiple input sources to MD5 that result in the hash collision occurrence.

2.1.1. MD5 Algorithm Procedure

MD5 takes arbitrary length of input and generates fixed length of 128 bit as output. Steps involved in MD5 procedure are listed as follows

Step 1: Add padding bits and length field to the original message

Step 2: The message from step 1 is partitioned into blocks of 512 bits: $B_1, B_2, \ldots, B_n$.

Step 3: The message blocks are processed by a compression function C as follows:

$V_k = C(V_{k-1}, B_k); \qquad k = 1,2, \ldots n$

Where V_{k-1} and B_k are the compression function input values and V_k is the output value, which is 128 bits long. In order to start the process, the initial value V_0 is provided by the MD5 standard.

Step 4: Finely, the message digest is given by V_k.

MD5 algorithm works as follows:

1. Padding bits and a 64-bit length field are added to the original message in order to make the resultant message as multiple of 512 bits (represented as M)

2. The resultant message (M) divided into series of 512 bits blocks (B1, B2, ...,Bn).
3. Divide 512 bit block (B) into 16 blocks(X1, X2 . . .X16) of 32 bits in size.
4. Initialize Channing variables

A, B, C, D are called chaining variables which are represented in four 32 bit Shift registers and these are initialized by following hexadecimal number

A = 0x67452301
B = 0xEFCDAB89
C = 0x98BADCFE
D = 0x10325376

5. The values of A, B, C and D are copied into four different variables as AA, BB, CC, and DD respectively.
6. Define 4 Logical functions (F,G,H,I)

F, G,H, I are four basic MD5 functions. Each function takes three 32-bit words as input and generates one 32-bit word as output. The algorithm involves 4 rounds and each round consisting of 16 steps. The total no. of steps is 64.

Table 1

Round	Function
1	$F(b, c, d) = (b \wedge c) \vee ((\neg b) \wedge d)$
2	$G(b, c, d) = (b \wedge d) \vee ((c \wedge (\neg d))$
3	$H(b, c, d) = b \oplus c \oplus d$
4	$I(b, c, d) = c \oplus (b \vee (\neg d))$

7. These 4 rounds utilize same operation structure but use different function namely F, G, H, I for each round respectively. The 4 operations are defined as follows.

FF(a, b, c, d, M[k], s, i): a=b+((a+ F(b,c,d)+M[k]+T[i]<<<s);D=C;C=B;B=A;A=D

GG(a, b, c, d, M[k], s, i): a=b+((a+ G(b,c,d)+M[k]+T[i]<<<s);D=C;C=B;B=A;A=D

HH(a, b, c, d, M[k], s, i): a=b+((a+ H(b,c,d)+M[k]+T[i]<<<s);D=C;C=B;B=A;A=D

II(a, b, c, d, M[k], s, i): a=b+((a+ I(b,c,d)+M[k]+T[i]<<<s);D=C;C=B;B=A;A=D

M[k], $0 \leq k \leq 15$ represents the kth sub-block of the message block(B) and <<<s represents left shift operation

8. At the end of four rounds, the output is added to the input of first round

A=A+AA;B=B+BB; C=C+CC; D=D+DD

9. All of these steps are completed, then the algorithm is continued to run the next 512-bit message block, finally the message digest produced 128 bit as output which is concatenation of values in A,B,C and D.

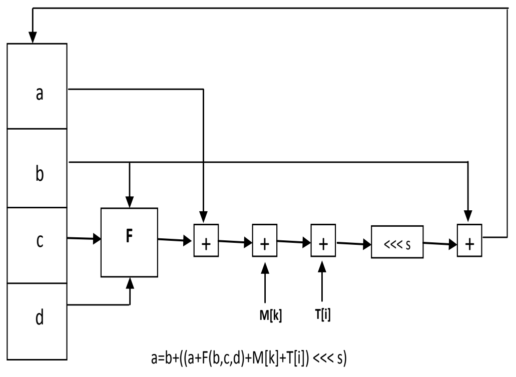

Fig. 1

2.2 Secure Hash Algorithm (SHA)

The Secure Hash (SHA) algorithm is a cryptography hash function based on Message-Digest series and it is used in data integrity.SHA was developed by NIST along with NSA and published in the year 1993, is intended for use with digital signature applications. It has different versions namely SHA-0, SHA-1, SHA-2 and SHA-3.

Table 2

Buffer	Hex Value
A	01 23 45 67
B	89 AB CD EF
C	FE DC BA 98
D	76 54 32 10
E	C3 D2 E1 F0

SHA-0: SHA-0 belongs from SHA family which produces a 160 bit message digest. It was removed shortly after publication due to “significant flaw” and replaced by the enhanced version SHA-1.

SHA-1: SHA-1 works for any length of message that is less than 2^{64} bits. It generates 160 bits length message digest value and was published in 1995.It is most widely used algorithm for its time efficiency and robustness. The weakness of this algorithm is generation of hash collision discovered in the year 2011[12].As computing power increases, SHA-1 will be able to be decrypted more easily and prone to exploitation by hackers.

SHA-2: SHA-2 is the technical successor to SHA-1 and offers greater security than SHA-1. SHA-2 consists of six hash functions, which includes SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. SHA-2 uses 64 to 80 rounds of cryptography operations and generally used to validate digital security certificates. SHA-2 ensures integrity but it is lagging in time management.

SHA-3: SHA-3 was released by NIST on 2015. It supports the same hash length as SHA-2 but it is very different from SHA-2 in design.

2.2.1. SHA-1 Algorithm Procedure

SHA-1 working mechanism is similar to MD5, but it produce 160 bit message digest as output. Steps involved in SHA-1 algorithm is

Step 1: Padding of bits

Message is padded with a single one bit (1) followed by as many 0's bits as required. Hence, it brings the message length into 64 bits less than the exact multiple of 512 bits.

Step 2: Append length

After tagging padding bits, the original length of the message is calculated. Add this original length value to the end of the padded message.

Step 3: split the input into 512 bit blocks: B_1, B_2, ...,B_n. These blocks become the main input for SHA-1 algorithm.

Step 4: Initialize buffers.

Step 5: Process Blocks - here the actual algorithm begins…

Step 5.1: Copy buffer values (A-E) into 5 registers (a-e), in which abcde is considered as 5 combined registers (32 bit in size) and hence total length is 160 bits. This register holds temporary results as well as final results.

Step 5.2: Divide 512 blocks into 16 sub-blocks of 32-bits each.

Step 5.3: SHA-1 has 4 rounds, each round consisting of 20 iterations.

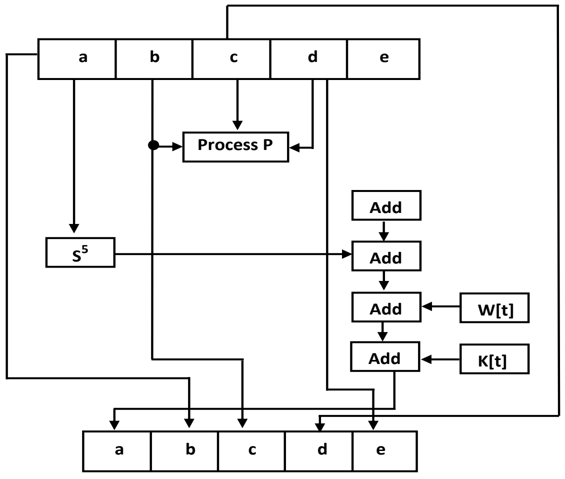

Fig. 2

Step 5.3: SHA-1 has a total number of iteration is 80(4 rounds x 20 Iterations).Each Iteration includes the following operations

abcde = (e + Logical Operation + S^5(a),W[t],K[t]),a,S^{30}(b), c, d

These 4 rounds utilize same structure but use different logical operation for each round respectively. The 4 operations are defined as follows.

Table 3

Round	Function
1	F(b, c, d) =(b ∧ c)∨ ((¬b) ∧ d)
2	G(b, c, d) = b ⊕ c ⊕ d
3	H(b, c, d) = (b ∧ c) ∨ (b ∧ d) ∨ (c ∧ d))
4	I(b, c, d) = b ⊕ c ⊕ d

Where,

W[t] →W[t], 0≤k≤15 corresponds to the kth sub block of the message block(B).

St →Circular left shift operation

K[t] →SHA-1 use one of the four additive constants for each round.

Table 4

Rounds	Iteration	K[t] Value
1	0 to 19	0x5A827999
2	20 to 39	0x6ED9EBA1
3	40 to 59	0x8F1BBCDC
4	60 to 79	0xCA62C1D6

After the 80th iteration the algorithm has produce 160 bit length hash value as output which is stored in combined registers.

III. COMPARATIVE ANALYSIS

Table 5 Comparison between various hash functions

Algorithms	Class of Algorithm	Output Length	Rounds	Collision Found
MD-5	Merkle Damgard Construction	128	64	Yes
SHA-0	Merkle	160	80	Yes

	Damgard Constructio n			
SHA-1	Merkle Damgard Constructio n	160	80	Yes
SHA-2	Merkle Damgard Constructio n	224,256 , 384,512	64/80	Theoretic al
SHA-3	Sponge Functions	224,256 , 384,512	24	No

SHA-1 is stronger but slower than MD5. It has very less chances of data collision occurrence. MD5 is faster but less secure than SHA-1.No significant Collision found on SHA-2 algorithms and it gives greater security than SHA-1, but despite SHA-1 is more preferable than SHA2 due to excellent time efficient. SHA-3 was faster than SHA-2 and SHA-1 in hardware implementation, but it is relatively slow in software.

1. ***Space Analysis***
   Table 2 shows memory space taken by each algorithm for different types of files. From this experiment, SHA512 has occupied more memory space than any other algorithms.

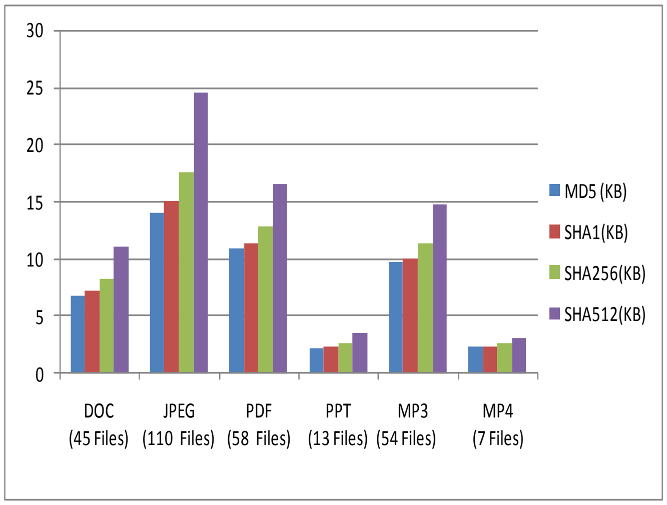

Fig. 3

Table 6

File Type	No.of Files in Folder	Memory space taken by the algorithms			
		MD5 (KB)	SHA1(KB)	SHA256(KB)	SHA512(KB)
DOC Files	45 Files	6.81	7.17	8.25	11.1
JPEG Files	110 Files	14.1	15	17.6	24.5
PDF Files	58 Files	10.9	11.4	12.8	16.5
PPT Files	13 Files	2.27	2.38	2.71	3.59
MP3 Files	54 Files	9.71	10.1	11.4	14.8
MP4 Files	7 Files	2.35	2.42	2.61	3.11

2. ***Time Analysis***
   Table 3 shows the result of time taken by each algorithm to generate hash value for different types of file. It is clearly seen that MD5 generate hash value everything else in a very short time. Furthermore, SHA512 and SHA256 are taking longer time to generate hash values than other two algorithms.

Table 7

File Type	Size (MB)	Time Taken by the Algorithms			
		MD5(ms)	SHA1(ms)	SHA256(ms)	SHA512(ms)
DOC Files	7.6	100	123	169	339
JPEG Files	4.18	131	148	163	618
PDF Files	62.1	384	501	1000	1800
PPT Files	18.4	148	163	385	632
MP3 Files	260	1200	1700	3200	5800
MP4 Files	322.4	1500	2000	4400	7000

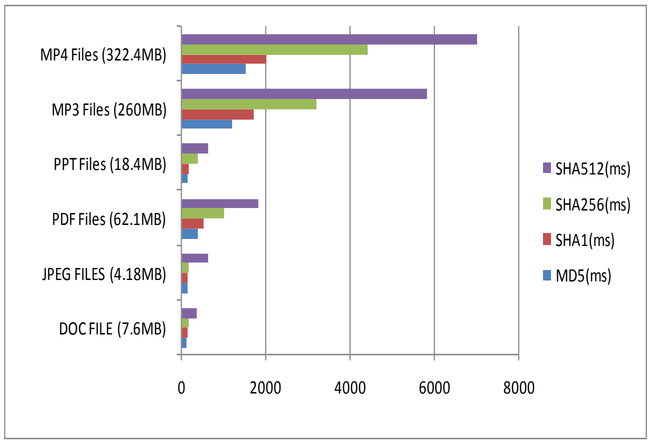

Fig. 4

IV. Conclusion

This paper overview the various cryptographic hash algorithms used to generate message digest value and compare them based on some of the parameters. From this analysis, it is found that MD5 is faster than any other algorithms. In addition, SHA1 is more desirable than SHA2 because of its excellent time efficient.

References

[1] Jean-Sebastien Coron, Yevgeniy Dodis, Cecile Malinaud, and Prashant Puniya,\ Merkle-Damgard Revisited : how to Construct a Hash Function. CDMP05.pdf.
[2] Wikipedia,http://en.wikipedia.org/wiki / Merkle-Damg% C3% A 5rd _has hash function.
[3] Rivest, R. The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
[4] Aiden Bruen, David Wehlau, Mario Forci --nito, Hash Functions Based on Sylvester Matrices," Patents Oce Kilkenny, September 20th 2001.
[5] D. R. Stinson, Some observations on the theory of cryptographic hash func-tions, University of Waterloo, Canada, March 2, 2001.
[6] Rudiger Weis, Stefan Lucks, \Cryptogra-phic Hash Functions, Recent Results on Cryptanalysis and their Implica-tions on System Security," Univeristy of Manheim.
[7] Xiaoyun Wang, Hongbo Yu, \How to Break MD5 and Other Hash Functions, Shandong University, Jinan 250100, China.

Authors Profile

Dr.P.Srivaramangai received her Ph.D Degree from Mother Teresa University, Kodaikanal in the year 2012. She received her M.Phil Degree from Manonmaniam University, Tirunelveli in the year 2003. She received his M.C.A Degree from Bharathidasan University, Trichy in the year 1996. She is working as Associate-Professor, PG and Research Department of Computer Science, Marudupandiyar College of Arts & Science, Thanjavur, Tamilnadu, India. She has above 30 years of experience in academic field. She published 25 papers in National & International journals so far. Her areas of interest include Computer Networks, Internet of Thing, Grid Computing, Cloud Computing and Mobile Computing.

Prof.J.Mariaselvam is a Ph.D scholar of Marudhupandiyar College, Thanjavur and persuing his research from 2017. He received his M.Phil Degree from Marudhupandiyar College, Thanjavur in the year 2009. He received his MSc IT Degree from Bharathidasan University, Trichy in the year 2002. He is working as Assistant Professor, Department of Information Technology, Annai Vailankanni Arts and Science College, Thanjavur, Tamilnadu, India. He has above 10 years of experience in academic field. He published 4 papers in National & International journals so far. His areas of interest include Cloud Computing and Data mining and Object Oriented Concepts.