Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 116 (2017) 381–392

Procedia Computer Science

www.elsevier.com/locate/procedia

2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia

A review of collisions in cryptographic hash function used in digital forensic tools

Zulfany Erlisa Rasjid[a*], Benfano Soewito[b], Gunawan Witjaksono[b], Edi Abdurachman[c]

[a]Doctor of Computer Science, School of Computer Science, Bina Nusantara University
[b]Binus Graduate Program, Bina Nusantara University
[c]Binus Graduate Program, School of Statiatics, Bina Nusantara University

Abstract

Digital forensic tool is a software used by digital evidence investigators to extract data and information from a digital evidence. The integrity of the digital evidence must be maintained through the chain of custody in order to be admissible in court. Most digital extraction tool use either MD5 (Message Digest) or SHA (Secured Hash Algorithm) hashing to check the integrity of digital evidence. The hashing algorithm has been found to have a weakness known as collision in which two different messages have the same hashing values. Although the probability of producing such weakness is very small, this collision can be used to deny the usage of the evidence in court of justice. After the first collision has been found, many cryptanalysts have tried to explore various methods to detect the collisions with shorter and efficient time. This paper is to review the existing methods in digital forensic tools that have been used to create a collision attacks in digital evidence.

Keywords: MD5; digital evidence; hash function; collision; digital forensic; cryptography

[*] Corresponding author.
E-mail address: zulfany@binus.ac.id

10.1016/j.procs.2017.10.072

1. Introduction

Cryptographic hash function is a function that converts a message of any length to a data of fixed length. The purpose of cryptographic hash is to ensure the integrity of data. Digital forensic tool is a tool to extract evidence data from different storage media, such as hard Drive, Memory, file system etc. There are several open source tools that are widely used to carry an investigation. Those tools include EnCase, SanSift Kit and many more. Most digital forensic tool use SHA (mostly SHA-1 because of its performance) and MD5 (Message Digest) hashing function to proof the data integrity. The problem with these digital forensic tools is a collision which has been reported by many researchers to indicate a weakness in this cryptographic has function. Collision is a condition whereby two or more files that has differences in contents and behaviors but having the same hash value. After the discovery of MD5 collision by Wang et al. [1], more and more cryptanalyst try to discover more collisions in more efficient time, for both MD5 and SHA hashes. The purpose of this review paper is to gain extensive knowledge on how collision attacks were performed using various methods, and how different strategies are proposed to improve hash algorithms.

2. Digital Forensics Tools

Six popular tools based on the information from http://resources.infosecinstitute.com/computer-forensics-tools/#gref that are commonly used in Digital Forensics which is summarized in table 1. It shows that top widely used Hash function that is used in Digital forensics is MD5 algorithm and some uses SHA algorithm, even though other algorithms are available such as RIPEMD and HAVAL. FTK Imager uses both MD5 and SHA. Normally investigators choose one instead of using both hashing methods as it involves additional time when using two different hash algorithms. Due to the fact that most Digital Forensics Tools uses MD5 and SHA hash algorithm to check the integrity of files, this paper reviews the two algorithms in terms of collision attacks. Collision attacks compromises the integrity of the digital evidence.

Table 1: List of most widely used Digital Forensic Tool

No.	Digital Forensic Tool	Hash Function	Features
1	EnCase	MD5	Remote Forensic Capability Evidence Processor Manager Smartphone and Table support Case Analyzer Email Review
2	San Sift	MD5	Network Forensics Computer Forensics Cloud Forensics Memory Forensics
3	Sleuth Kit	MD5	Contains a collection of unix commands for volume analysis and file systems
4	FTK Imager	SHA1 and MD5	Acquire and Preserve data from different media Forensics for computer and mobile Detect and validate suspected Malicious activities
5	Bulk Extractor	MD5	Forensic Scanner Feature Extraction Files, images and emails

6	Oxygen Forensic Suite	SHA2	Extract and Analyze data MMS, SMS, email messages Device logs Image, video files

3. Hashing Algorithm MD5

In Digital Forensic, digital evidence to be used must come from a duplicated one and not from the original evidence. Only copies of the original evidence are used by forensic team. However, integrity must be maintained. Hashing algorithm play a very important role to ensure evidence integrity [23]. Since most digital forensic tools either uses MD5 and SHA hashing algorithm, this paper will discuss only those two algorithms even though other algorithms are available. Hash function $H(x)$ is defined as "computationally difficult" to find for a given x where $H(x) = H(x')$ where $x \neq x'$. A hash function should follow three properties: (1) A pre-image resistance, for a given message in a domain (x'), it should be "computationally difficult" to find an $x \in Domain$ such that $H(x) = x'$; (2) Second pre-image resistance, which is defined that it should be "computationally difficult" to find a distinct $x' \in Domain$ for any $x \in Domain$ such that $H(x) = H(x')$; (3) Collision Resistance, It should be "computationally difficult" to find distinct x and x', both in the $Domain$, such that $H(x) = H(x')$. A collision is defined as having the same hash values for different input. Considering two different files that behave differently the hash values for these two files should be different. Dealing with digital evidence, such collision means that the evidence is not admissible in court. MD5 and SHA algorithms are based on the Merkle-Damgard Construction. This paper discusses the Merkle Damgard Constructions and how collisions can be obtained due to this construction.

3.1. Merkle-Damgard Construction

The Merkle-Damgard construction is the foundation to cryptographic hash functions. A cryptographic hash function is a one-way function to create a message digest. Figure 1 shows the Merkle-Damgard construction. The algorithm uses a fixed value known as the initial value (IV). The message of any length is divided into 512 block messages with padding at the last block in order to make equal length of each block. A function (f) is applied to each message block. The resulting value will be used as input for the next message block and so on until the final hash value is obtained. Using the Merkle-Damgard construction where the input is a message of any length however the resulting Message Digest of fixed length. Message Digest Algorithms such as MD5 and SHA series are based on this construction. MD5 and SHA series differs in terms of the message length, the function and the number of rounds.

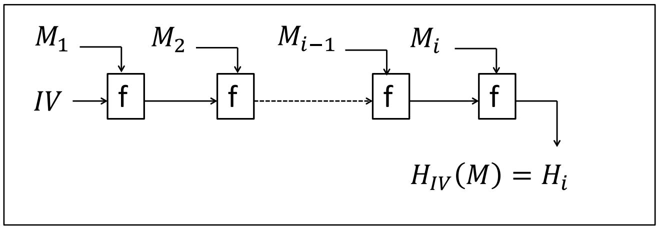


Fig 1. The Merkle-Damgard Construction [4]

This construction has a weakness, it can be attacked by "message expansion attack". A message expansion attack is defined as follows: knowing the hash of a particular message m, then the hash of $m||y$ can easily be calculated for any block of y [5].

This construction is using iterative process, i.e. the output resulting from previous function is used in the next function, there exist a pattern on how the final hash is obtained.

3.2. MD5 (Message Digest)

MD5 (Message Digest) was developed by Ronald Rivest [6] to improve an earlier version which is MD4 algorithm and it is based on the Merkle-Damgard construction. The MD4 algorithm is as follows:

1. Padding of message.

The length of input stream must be 448 mod 512. Therefore the input stream must be padded. The padding is performed by adding a "1" then followed by "0" until the length is congruent to 448 md 512.

2. Let b = the length of the stream before padding. The value of b is appended to the result of the previous step 1, where b is represented in 64-bit representation.

3. In this step, the Message Digest (MD) buffer is initialized, using 32-bit four-word buffer (let say A,B,C and D):

A: 01 23 45 67
B: 89 ab cd ef
C: fe dc ba 98
D: 76 54 32 10

4. The message is processed in 16-word blocks. The process uses three rounds of calculations.

5. Produce output

MD4 has a weakness which is found by Den Boer and Bosselaers and published in 1991[7] and in 1995 a full collision was found and published by Hans Dobbertin [8]. Because of this weakness in MD4, Rivest improves the MD4 functions by adding one more round to the MD4 algorithm. The second round was changed from the function $XY \vee XZ \ \vee YZ$ to the multiplexer function $XZ \vee Y(Z')$ [6]. The order of accessed, round 2 and 3 are interchanged. The number of shifts is also changed. The other changes are that at each step, a unique additive constant is added and at each step, the result of the previous step was used. These changes were performed to handle collision on MD4 [9]. MD5 proved to be more secure and better performance than MD4 [10].

3.3. Hash Collisions

A collision is a condition whereby two messages, let say m_1 and m_2, after applying the hash value, then $H(m_1) = H(m_2)$. A collision can always be found using Brute Force algorithm, however it is computationally difficult. There are two types of collisions, the strong collision and weak collision. Figure 2 and 3 shows the illustration of strong and weak collision respectively.

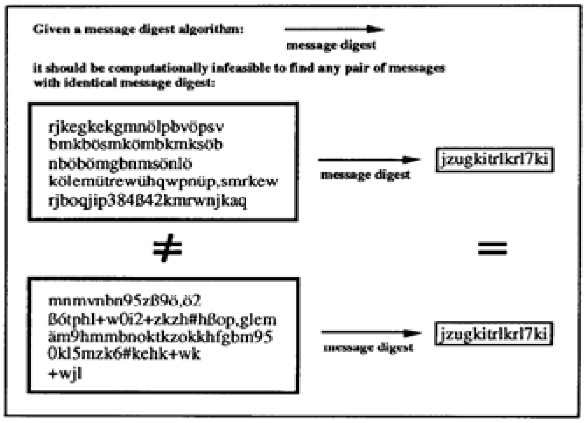
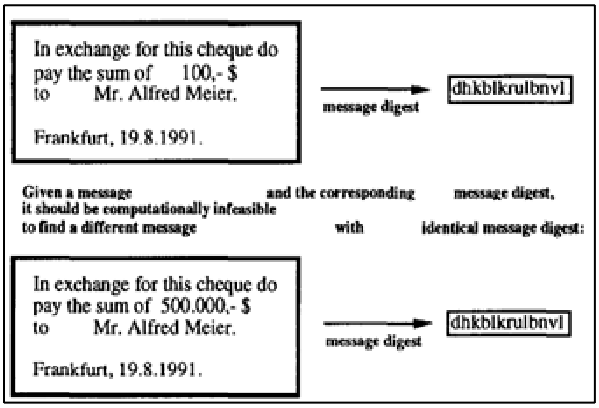


Fig. 2. Strong Collision [11]


In exchange for this cheque do
pay the sum of 100,- $
to Mr. Alfred Meier.

Frankfurt, 19.8.1991.

Given a message and the corresponding message digest, it should be computationally infeasible to find a different message with identical message digest:

In exchange for this cheque do
pay the sum of 500.000,- $
to Mr. Alfred Meier.

Frankfurt, 19.8.1991.

Fig. 3. Weak Collision [11]

Brute Force algorithm has the worst performance. Cryptanalysts use several methods, such as the differential and chosen prefix method. In differential cryptanalysis a hash function is evaluated different inputs and the difference in the result is evaluated and analyzed. The differences are used to build the attack. In chosen Prefix collision, any two message prefixes such as P and P', and suffix S and S' are chosen such that such $P\ OR\ S$ and $P'OR\ S'$ can be constructed that will produce a collision [12]. The message is much larger than the fixed length hash values, therefore collision can exist, in fact, it could be many, as stated by the pigeonhole principle, several collisions definitely exist. A brute force attack can find a collision with n-bit hashes in approximately 2^n hash operations. A brute force approach to generate collisions will succeed in approximately $2^{n}/_2$ hash operations due to the birthday paradox. In the birthday paradox, the probability of having two people with the same birthday is 100% when the number of people reached 367 (as there are 366 days in a year).

3.4. Collision Attacks

The first collision attack on MD5 was announced by Boer et al., proved that the last step of MD5, allows the creation of collision [7]. This is due to the fact that the result of the previous step was used in the addition process. This function is not considered very harmful, as it is only achieved at portion of MD5 step, not the full MD5 process, as stated by H. Dobbertin, by calling it "pseudo-collision" [13]. The collision was found by choosing two different messages using selected initial value (IV).

H. Dobbertin found a weakness in MD5, a pseudo collision consisting of two different messages with a chosen Initial Value (IV) [13]. A pseudo collision occurs if two different initial value, say IV' and IV'', having input x_1 and x_2 such that:

$$H(IV', x_1) = H(IV'', x_2) \tag{1}$$

The major breakthrough in MD5 collision is when Wang et. al. at 2004 at Eurocrypt conference [1] is able to construct collision on MD5, using the initial value IV_0 :

$$IV_0 = A_0 = 0x01234567, B_0 = 0x89abcdef, C_0 = 0xfedcba98, D_0 = 0x76543210$$
$$M_k^{'} = M_k + \Delta c_1, \Delta c_1 = (0,0,0,0,2^{31},0,0,0,0,0,0.2^{15},0,0,2^{31},0), s = 4,11,14$$
$$M_k = M_k + \Delta c_1 + \Delta c_2, \Delta c_2 = (0,0,0,0,2^{31},0,0,0,0,0,0,2^{15},0,0,2^{31},0), s = 4,11,14$$

Such that

$$MD5(M, N_i) = MD5(M'\ , N_i') \tag{2}$$

The collision can also be produced for MD4, HAVAL-128 and RIPEMD.

When collision is found, the hash algorithm has been compromised. Due to this collisions, Gauravaram, P. et. al in 1996 developed a new hash function, the 3C construction to reduce collisions. Guaravvaram improved the Merkle-Damgard construction [4] by introducing the 3C contstructio. Other enhancement of the block cypher was introduced by Puniya et al. where chopping the last bit of the hash result to enhance the security. It is claimed that because of the bits chopping, construction of collision is reduced[5]. However, multi-block collision can still be found as it is an enhancement of the Merkle-Damgard construction [14], in fact the collision (2 block) in SHA-1 also appears in the 3C construction. [14]

In 2006, in the studies of MD5 attacks, Black et. al. analyzes the collision found by Wang et.al. in 2004[13]. They have provided methods to create other differential path to find collisions and improve the collision that was produced in 2004. The algorithm to find collision is as follows: Algorithm find_collision as scribed by Black, J. et al. [15]

```
While collision found is false do
    1. Use random seeds and deterministic methods to find M which satisfies most conditions on Qi
    2. Compute all Qi and Qi’ to see if differential are correct
    3. if (rest of differentials hold) then collision_found ← true else collision_found ← false
Enddo
Return M
```

The main improvement is the time taken to create the collision. They identify patterns in the step such that it would give probable values and this range of values is used in the search therefore reducing the amount of time required to search for collision [15].

The search for collisions using Wang and her team’s method took a long time. Klima, V., improved the collision search proposed by Wang and her team by about 6 times faster by generating the first message blocks that collide in a faster way [16].

Up to 2006, the collision that was generated has no important meaning, i.e. the files are just two arbitrary files having the same hash values. In 2006, researcher Kasyap,N. , used the techniques to create collision and then use it to develop the collision into a serious and valid collision [17] using a technique introduced by Daum and Lucks, 2005 called the poison message attack. This attack resulted in two different files that behave differently.

Wang’s method to find a collision has the complexity of 2^37. Yu Sasaki et al., improved this method to find collision with a complexity of 2^30 [18]. In Wang’s method, modification of each bit is recalculated (to find collisions) at step 15 through to 64. Sasaki’s method repeats modifications from 25 to 64. This improved the performance.

LiangXue Jia and Lai Jie, 2007, provide improvement of collision attack. The attack is based on two-block collision differential attack presented by Wang et al. at Eurocrypt 2005, using a small range searching and also omitting the steps to check the characteristic. By changing the modification techniques, a collision is always found. This method allows to find collision at a faster speed [19].

Marc Stevens et al., showed how chosen-prefix collision can be used to construct MD5 collision. Chosen prefix collision is selecting two randomly chosen message that would give two different Intermediate Hash Value (IHV) that can be appended in a way that would cause messages to collide [1220]. They have chosen two different IHVs padding the message to 416 mod 512, leaving the last block incomplete. Using Wang’s method to find 96-bit values to complete the last block. The values are found using the birthday attack, forming a difference vector between the IHV. The remaining difference is removed by appending “near-collision” blocks, obtained by an automated differential path construction [12]

Aoki and Sasaki in 2008 [21] found a meet-in-the-middle pre-image attack on hash functions, MD5 and HAVAL. Later in 2009 improved their preimage attack by using transformations that reduce the consumption of memory. The pre-image attack on MD5 improved from 2^{43} to 2^{13} [22].

Xiaoyun Wang and Hongbo Yu in 2009, introduced another attack on MD5 [23]. The attack is using a modular differential method instead of the exhaustive differential method. Differential attacks, introduced by E. Biham and A. Shamir analyzes the differences of the plain text pair and the resulted pair [24]. The differences are used to provide the probability and to locate the most probable key. In DES (Data Encryption Standard) the differences is obtained by performing XOR to the two plaintext value.

Xie and Feng in 2010 showed a method to find weak difference that allows collision to be found[25]. Their result indicated that are a lot of such weak differences found, much more to the one found by Wang et.al. Xie and Feng introduced a method to find a collision using one block message [26] where the method is deliberately not published and serve as a challenge. Researcher Marc Stevens responded to the challenge and identified a method to find a single block collision, by choosing message blocks that would satisfy the bit condition [27]. He uses tunnels to correct the bit condition [27]. Steven's research in 2012 has a performance of 2^49.8 [27]. Kuznetsov uses parallel algorithm to improve on finding single-block collision[28].

With modern computer that are available widely nowadays, Chiriaco et al., in 2016, found collision using brute force parallel programming reducing the search time significantly [29]. The following fig. 4 shows the result of finding collision using 1, 2, 4, 8 and 16 processors. It can be seen that the performance improved from 2.26 seconds using one processor and 0.367 using 16 processors.[29]

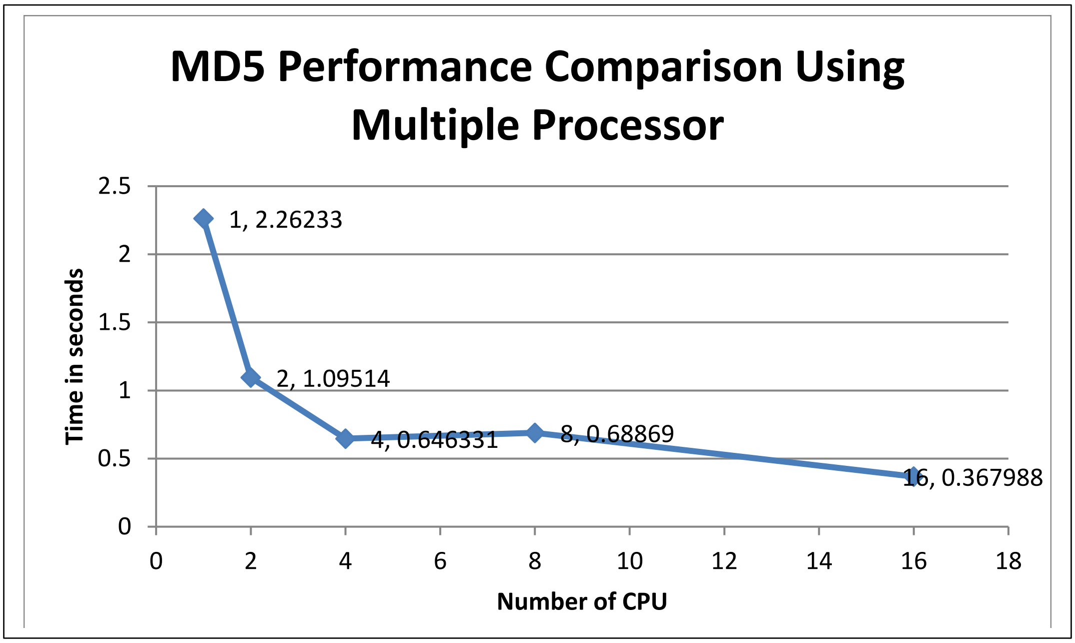


Figure 4. MD5 Performance [29]

4. Hashing Algorithm SHA (Secured Hash Algorithm)

Secured Hash Algorithm (SHA) is an algorithm based on MD4 algorithm designed by the National Security Agency of USA. SHA-0 was issued by the National Institute of Standard and Technology (NIST). SHA is build based on the basics of MD4 . SHA family uses 512 bit message blocks and resulted in a 160 bit output (message digest). The following shows the process of the hashing:

1. Message is padded by adding 1 and 0 and 64 bits integer representing the length of the message.
2. Initialize 5 registers of 32 bits using fixed constants:

A = 0x67452301
B = 0xEFCDhB89
C = 0x98BADCFE
D = 0x10325476

E = 0xC3D2EIF0

3. Copy A, B, C, D and E in AA,BB,CC,DD and EE and then apply the compression functions. The results are added to A,B,C,D and E respectively.

4. The result is the concatenation of A,B,C,D and E.

F. SHA variants

1) SHA-0

SHA-0 is created by a slight modification of a 160-bit hash function that was published in 1993 (SHA). However, due to a serious flaw, it was withdraw and replaced by SHA-1.

2) SHA-1

SHA-1 is a 160-bit hash function designed by the National Security Agency (NSA) in which the algorithm is similar to the early MD5 algorithm. SHA-1 improves SHA-0 by removing the flaws. The difference between SHA0 and SHA1 is in the compression function where a single bitwise rotation is performed[30].

3) SHA-2

SHA-2 comprises of several hash functions, SHA-256, SHA-224 and SHA-512, where the name is based on the digest lengths.

G. SHA Collision Attack

Florent Chabaud and Antoine Joux produced a differential attack on SHA-0 [31] in 1998. However, the method used does not work for SHA-1. In 2005, Wang et al. in 2005 produced a collision for SHA-1 with a complexity of 2^69 hash operations [32]. In 2012 Stevens et. al., introduced the first full collision attack of SHA-1 [32], and one year later, he introduced a new collision attack on SHA-1, using a new technique with a complexity of 2^57.5 [33] . In 2016 Marc Stevens, based on his research presented in 2013 and the research he conducted with Karpman.[34], produced a freestart collision for SHA-1 [35].

The first preimage attack on SHA-2 was found by Isobe and Shibutani in 2009. They have presented a one block preimage attack, which is based on meet-in-the-middle attack, on both SHA-256 and SHA-512 with a complexity of 2^240 and 2^480[36]. Later in the same year Aoki et al. improved the method by splitting the function, calculate them individually and perform a match uses birthday-style method [37]. In 2013, researchers Mendel et al., improved the collision attack by increasing the size of local collisions and use it to create a differential method and hence improved the collision search [38]. Mendel's technique, presented at Eurocrypt 2013, cannot be applied to SHA-512 due to the large search space. In 2014, Eichlseder et al., presented a technique using branching heuristics that improved the strategy to find collision for SHA-512, which increase the speed of collision search by 2^20 [38]. Raghuvanshi et al., analyzes different algorithms based on its basic properties. The following table shows the result:

Table 2: Comparison of Basic Characteristics of MD series and SHA series. [3039].

Algorithm	Output Size	Rounds	Collision Found	Performance (MiB/s) [39]
MD5	128	48	yes	335
SHA-0	160	80	yes	-
SHA-1	160	80	yes	192
SHA-2	224/256/84/512	64/80	theoretical	139-154

Although collision has been found in MD5 and SHA1, and not SHA2, digital forensic tools still use MD5 due to performance issues.

5. Conclusion

Collisions in hash values show that two different files that have the same hash values compromises the integrity of files or data. In the case of digital evidence, the effect can be fatal. Since the first collision was found cryptanalyst have been producing collision attacks on hashing functions for MD5 as well as SHA series. The collisions are produced for arbitrary files as well as meaningful files. The following tables show the summary of collision performance in MD5 and SHA (Table 3 and table 4 respectively).

Table 3. Collision performance in MD5

Performance of Hash Collisions in MD5			
Hash Algorithms	Hash Functions	Performance Complexity	Notes
Wang's algorithm	MD5	2^{37}	repeats modification from steps 15 through to 64
Klima's algorithm	MD5	$2^{34.4}$	
Yu Sasaki	MD5	2^{30}	repeats modification from steps 25 through to 64
Steven's	MD5		

Table 4. Collision Performance of SHA

Performance of Hash Collisions in SHA			
Hash Algorithms	Hash Functions	Performance Complexity	Notes
Wang's algorithm	SHA	2^{69}	
Steven's	SHA	$2^{57.5}$	
Eichlseder	SHA	2^{30}	using branching heuristics

The following chart shows that the performance of collision finding is improving since the first collision found until today, and as can be seen in the charts, the performances keep improving.

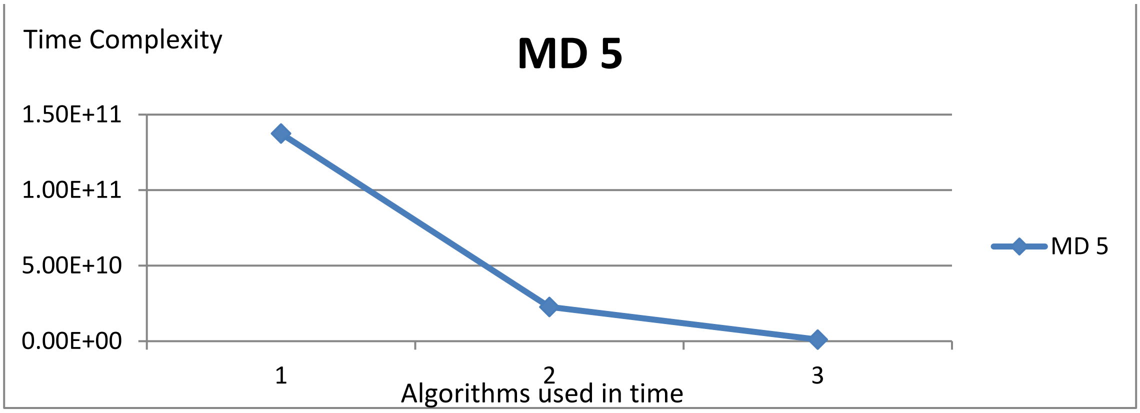
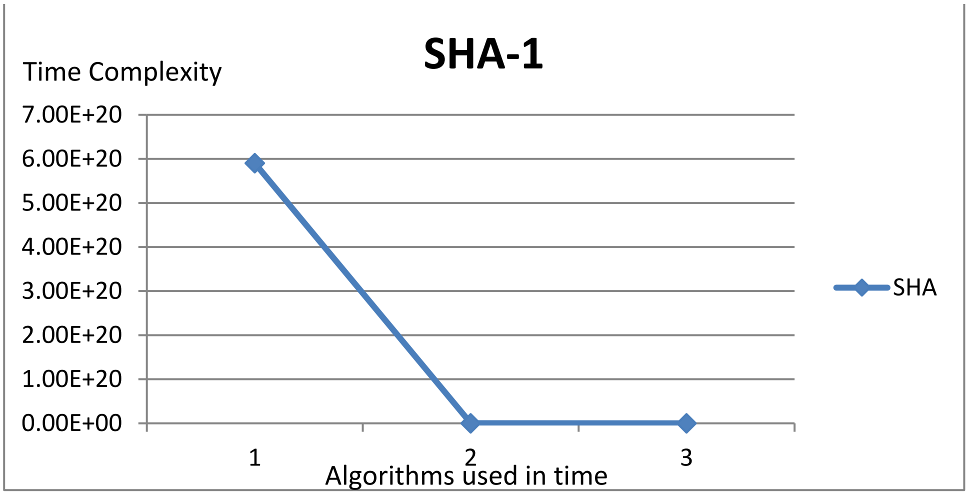


Fig.5 Performance of MD5 collision



Fig, 6 Performance of SHA-1

. Cryptanalysts have found several ways to find collisions in a faster way, however, only a few tried to improve the hash functions to reduce the collisions, and yet digital forensics tools are still using MD5 or SHA series hash functions.

6. Future Works.

In the context of digital forensic, the main issue is the validity of the digital evidence, in which its integrity is validated by the hashing function Since the digital forensic tools uses either MD5 or SHA series hashing functions and collisions can be found, with the complexity of less and improved in time, further research is necessary to reduce the probability of such an attack to ensure the admissibility of evidence in court.

References

1. Wang X, Feng D, Lai X, Yu H. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. Int Assoc Cryptologic Res.

2004;5:5–8.

2. Saleem S. Evaluation of Security Methods for Ensuring the Integrity of Digital Evidence. 2011;220–5.
3. Karie NM. Towards a Framework for Enhancing Potential Digital Evidence Presentation. 2013;
4. Gauravaram P, Millan W, Dawson E, Viswanathan K. Constructing secure hash functions by enhancing Merkle-Damgard construction. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2006;4058 LNCS(Iv):407–20.
5. Jean-Sebastien Coron, Yevgeniy Dodis CM, Puniya P. Merkle Damgard Revisited : how to Construct a hash Function The Random Oracle Methodology. NSF Carrer Award CCR. 2002;1–45.
6. Rivest R. MD5 Message Digest. 1992. p. 1–21.
7. Den Boer B, Bosselaers a. An attack on the last two rounds of MD4. Adv Cryptol – CRYPTO 1991 [Internet]. 1992;194–203. Available at: http://www.springerlink.com/index/4yb85bkpxt5gemmh.pdf
8. Dobbertin H. Cryptanalysis of MD4. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 1996;1039:53–69.
9. Boer B Den, Bosselaers A. Advances in Cryptology — EUROCRYPT '93. Adv Cryptology—EUROCRYPT'93. 1994;765:293–304.
10. Thomas CG, Jose RT. A Comparative Study on Different Hashing Algorithms. 2015;170–5.
11. Bauspiess F, Damm F. Requirements for cryptographic hash functions. Comput Secur. 1992;11(5):427–37.
12. Stevens M, Lenstra A, Weger B De. Chosen-prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities. Adv Cryptol – Eurocrypt 2007 [Internet]. 2007;vol. 4515(Lecture Notes in Computer Science):1–22. Available at: http://www.win.tue.nl/hashclash/EC07v2.0.pdf
13. Dobbertin H. Cryptanalysis of MD5 Compress. Ger Inf Secur Agency. 1996;
14. Joscák D, Tuma J. Multi-block Collisions in Hash Functions Based on {3C} and {3C+} Enhancements of the {Merkle}-{Damgard} Construction. Inf Secur Cryptol – ICISC 2006 [Internet]. 2006;4296:257–66. Available at: http://dblp.uni-trier.de/db/conf/icisc/icisc2006.html#JoscakT06
15. Black J, Cochran M. A Study of the MD5 Attacks : Insights and Improvements. Fast Softw Encryption. 2006;262–77.
16. Klíma V. Finding MD5 Collisions – a Toy For a Notebook. Int Assoc Cryptologic Res. 2005;(February):1–7.
17. Kashyap N. A Meaningful MD5 Hash Collision Attack [Internet]. 2006. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.2659&amp;rep=rep1&amp;type=pdf
18. Sasaki Y, Naito Y. Improved Collision Attack on MD5. Complexity. :1–11.
19. Lai J, LiangXue-Jia. Improved Collision Attak on Hash Function MD5. J Comput Sci Technol. 2007;22(1):79–87.
20. Stevens M, Sotirov A, Appelbaum J, Lenstra A, Molnar D, Osvik DA, et al. Short Chosen-Prefix Collisions for MD5 and the Creation of a RLecture ogue CA Certificate. Lect Notes Comput Sci. 5667:1–17.
21. Aoki, K.; Sasaki Y. Preimage attacks on one-block MD4, 63-step MD5 and more. In: Selected Areas in Cryptography SAC 2008. Berlin, Heidelberg, New York. Springer-Verlag.: Springer; 2008. p. 103–19.
22. Sakiyama K, Ohta K. Meet-in-the-Middle Preimage Attacks Revisited New Results on MD5 and HAVAL. 2009;
23. Wang X, Lai X, Feng D, Chen H, Yu X. Cryptanalysis of the Hash Functions MD4 and RIPEMD. 1998;
24. Biham E, Shamir A. Differential Cryptanalysis of DES-like Cryptosystems 1. 1991;3–72.
25. Xie T, Liu F, Feng D. Fast collision attack on MD5. IACR ePrint Arch Rep [Internet]. 2006;104:17. Available at: http://crppit.epfl.ch/documentation/Hash_Function/Examples/Code_Project/Documentation/104.pdf
26. Xie T, Feng D. Construct MD5 Collisions Using Just A Single Block Of Message. 2013;61070228.
27. Stevens M, Group C. Single-block collision attack on MD5. 2012;1–11.
28. Kuznetsov AA. An algorithm for MD5 single-block collision attack using high- performance computing cluster. 2014;
29. Chiriaco, Vincent;Franzen, Aubrey; Tayil, Rebecca; Zhang X. Finding Partial Hash Collisions by Brute Force Parallel Programming. In Sarnoff Symposium, 2016 IEEE 37th; 2016. p. 1–2.
30. Vadhera P, Lall B. Review Paper on Secure Hashing Algorithm and Its Variants. 2014;3(6):2012–5.
31. Chabaud F, Joux A. Differential collisions in SHA-0. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 1998;1462:56–71.
32. Wang X, Yin YL, Yu H. Finding Collisions in the Full SHA-1. 2005;(90304009):17–36.
33. Stevens M. New collision attacks on SHA-1 based on optimal joint local-collision analysis. Adv Cryptol - CRYPTO 2013. 2013;7881:245–61.
34. Karpman P, Peyrin T, Stevens M. Practical Free-Start Collision Attacks on 76-step SHA-1. Adv CRYPTO 2015. 2015;2012.
35. Stevens M, Karpman P, Peyrin T. Freestart collision for full SHA-1. 2016;2012:1–21.
36. Isobe, T; Shibutani K. Preimage Attacks on Reduced Tiger and SHA-2. Fast Softw Encryption. 2009;5665:139–55.

37. Aoki, Kazumaro; Guo, Jian;Matsusiewics, Krystian; Sasaki, Yu; Wang L. Preimages for step-Reduced SHA2. Adv Cryptology-ASIA CRYPT 2009. 2009;5912.

38. Mendel, Florian; Nad, Tomislav; and Schlaeffe M. Improving Local Collisions: New Attacks on Reduced SHA-256. Adv Cryptol - Eurocrypt 2013. 2013;7881:262–78.

39. Raghuvanshi, Kamlesh Kumar; Khurana P, Bindal P. Study and Comparative Analysis of Different Hash Algorithm. 2014;3(9):2012–4.