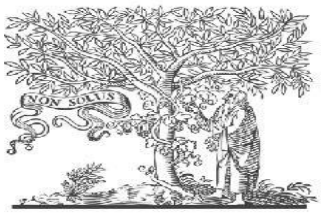
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT



ELSEVIER
SSRN

IJIEMR Transactions, online available on 15th Nov 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-11

DOI: 10.48047/IJIEMR/V10/I11/21

Title **Performance of Cryptographic Hash function used in Digital Forensic tools**

Volume 10, Issue 11, Pages: 154-157

Paper Authors
Kakunuri Sandya, Subhadra Kompella

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Performance of Cryptographic Hash function used in Digital Forensic tools

[1]Kakunuri Sandya, [2]Subhadra Kompella
skakunur@gitam.in, skompell@gitam.edu
Department of Computer Science and Engineering, Institute of Technology
GITAM (Deemed to be University), Vishakhapatnam

ABSTRACT
Cryptographic hash functions are which transform any long message to fixed-length data. It seeks to ensure the confidentiality of the data through the cryptographic hash. The digital forensic tool is a method for extracting information from various storage devices, such as hard drives, memory. SHA-1 and SHA-2 methods are both widely used in forensic image archives. The hash method is usually used during evidence processing, the checking of forensic images (duplicate evidence), then at the completion of the analysis again to ensure data integrity and forensic evaluation of evidence. There was a vulnerability called a collision in the hashing algorithm in which two independent messages had the same hash values. While SHA-3 is secure than its former counterparts, the processors for general purposes are being slow and are not yet so popular. This task proposes a basic yet successful framework to meet the needs of cyber forensics, combining hash functions with other cryptographic concepts, for instance, SALT, such as modified secured hash algorithm (MSHA). A salt applies to the hashing mechanism to make it exclusive, expand its complexity and reduce user attacks like hash tables without increasing user requirements.

Key words:*Cryptography, SALT, digital evidence, hash function, SHA1,*

1. INTRODUCTION
Cryptography has not proven as successful in real-world networks and applications because of engineering problems from a mathematical perspective. The development of cryptographic schemes in the modern world differs from conceptual cryptographic ideas of just pure mathematics. Designers and implementers also faced real-world limitations as engineering challenges. Developers have to learn cryptography in real-world contexts to train students for specific real-world security objectives. Therefore, a cryptography method simulates and documents real-world cryptographic implementations [1- 3] needs to be designed and integrated in case studies. The technique of case studies in cryptography is a positive practice since it involves students in a real-life environment that encourages student imagination. Studies have contributed to adapt encryption to generate answers to research areas.

The cybersecurity system, which primarily serves to secure the identification, conservation, analysis, and presentation of digital data information in a way that is legitimately accepted in any judicial procedure, may be identified as digital forensic. Digital forensics was often used to retrieve materials from a digital media website and network [4]. This forensics utilizes the science analysis approach, where testimony is mainly based in the fields of forensics. They required a code of ethics of neutral and testing methods for the certification of forensics. This digital forensic has a good tradition as the initial people who felt the need to secure the data from every digital gadget found it. They estimated the first gadget that makes its first show in 1970 to be over 40 years old.

This was when the society requested the Court of Justice. The investigator's mandate must be thoroughly familiar with the criminal investigation and investigation. The inquiry is conducted, and the fraud to be resolved and changed. It also required the researcher to use the current digital forensics techniques because of the sped-up technological advancement. The investigator shall provide the evidence locations during the inquiry. The detective can decide when the crime existed and then figure out what might have occurred. Various places will include various types of fraud, leading to multiple types of locations. Each proof will often provide a separate indicator of its resolution. Digital forensics varies from place to place, and measures are also different to tackle these frauds. Proof often provides data of various types based on the venue which demand a certain method for its examination. The site may often include some types of people, meaning that the people have a certain form of coping with forensic crime.

For electronic evidence, hash values are used. Used primarily on digital forensics procedure examinations [5]. We can use Hash values to ensure we also could not change the original copy. They create an image from the initial during the operation. They often considered the initial hard disc as a hash attribute. Even before the hash value is taken, they complete the test. If the values are the same, the copy shall be considered as the original, and the values shall be different, so the copy shall be asked. They sometimes took a third value after it the test was completed. The three hash values including, initial hard drive, hard disc image before the test, and hard drive image after testing must fit. The three values must match. Again, in the court, it is possible to use the hash values for validating testimony. A number and letter string with a long fix and randomly sized file, including a text, paper, image or other data form, created by

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

a mathematical algorithm. This produced string is a one-way feature of the hacked file and can't reverse a computed hash to locate other files to produce a certain hash value. Secure Hash Algorithm-1 (SHA-1), Secure Hashing Algorithm-2 (SHA 2 and SHA-256), and Messaging Digest 5 (MD5) are some of the more common hacking algorithms currently in use. In this paper, two hash functions - MSHA-1 and MSHA- 2, are combined laterally with the use of SALT to compute collision-resistant hash values

2. BACKGROUND WORKS

Some of the relevant and recent papers are mentioned here
In [7], the authors examined the importance of hash functions in digital forensics. The strength of hash functions helped us grasp even the slightest variations in the input. Found out more about different researchers and cryptographic experts using such hacking features, such as MD5, which enabled us to learn. Any cryptography experts have insisted that people no longer use MD5 because it is not collision-resistant and may be attacked for several reasons–birthday attack, Rainbow attack, etc. However, there are those that claim that colliding resistance is improbable in real time, and that it is a waste of resources to think about this.

In [8], the authors present the MD5 and SHA-1 hash functions with the sponge structure modulation. The work constructs the suggested scheme with the Keccak permutation function. The paper addresses the two major security violations threatening the collision and long extension threat cryptography. They define the potentials to deal with collisions and extension attacks by evaluating many samples of collided messages from both algorithms (SHA-1 and MD5). In addition, this paper discusses a proper substitution strategy to prevent such attacks.

In [9], the authors suggested increasing the performance of the SHA-256 hash function during the unfolding process of transformation. Three forms of SHA-256 have been built with the design SHA-256, the interior pipeline design SHA-256, and the inner pipelines design SHA-256 with the development factors 2 and 4. Design SHA-256. The designs were written in Verilog, and ModelSim checked the performance simulations. Simulation findings revealed that the suggested SHA-256 inner pipeline with factor 4 delivered the highest output of 4196,30 Mbps and factor 2 was superior to that of the traditional SHA-256 in terms of high frequency

In 10], the authors used a high-performance application-specific instrument-set processor (ASIP) is provided [10] for cryptographical hash functions. The processor is obtained by co-design technique of hardware and software and speeds up the hash functions SHA (Secure Hash algorithm) and MD5. In the 65 nm CMOS method, the architecture proposed covers 0.28 mm2 (66 kgates), including a single port memory of 4.5 KB and logic of 52 kgates. The performance for the design proposed hits MD5, SHA-1, SHA-512, and SHA-412 respectively below the clock frequency of 1,0 GHz 15.8 Gb/s, 12.5 Gb/s, 12.2 Gb/s and 19.9 Gb/s. They evaluated the design proposed with innovative VLSI designs, which uncover high performance, low silicon cost, and complete programming.

In [11] the authors introduce the Merkle-Damgû theorem, and the construction suggests collision resistance to a compression element. Multi-block collisions were recently discovered using differential cryptanalysis on the MD5, SHA-0, and SHA-1 Hash functions. These multi-block collisions pose many concerns about several concepts and properties used in the literature on hash functions. In this study they have looked at and provided insights into some of the literature on cryptographic hash functions. Later they introduced significant variations between the hash function of Damgû are in 1989 and the following hash functions. They stated that the pseudo-collision attack did not fall under the architecture requirements of the hash functions. They often question if these hashes they design functions according to the philosophy of the construction of Merkle-Damgû.

In [12], the authors present algorithms for Hash functions, also known as the message digest algorithms, which compress an arbitrary-length message input and generate an allocated random output at a fixed length. There are several hash algorithms like MD5 and SHA1. These algorithms check the credibility of the data and restrict unwanted data change. That being said, in particular, when implemented in Bitcoin mining and in low-computing, particularly IoT devices, they experience some complexity. In this study, they propose a new compression feature that decreases the difficulty of functional hash algorithms, including MD5 and SHA-1. It also shows that the initial compressing part provides us with the same effects.

3. METHODOLOGY AND METHODS:

The Modified SHA-1 (MSHA-1) model with the counter is shown in Figure 1. The intermediate hash value is applied to XORed. This addition improved the construction of Merkle-Damgard (M-D) since the counter adjusts every step of the process. The counter starts at an initial value of 0, and with each message, block is increased by 1 to the final block. The proposed SHA compression algorithm maintained eighty rounds. In order to improve the algorithm, the modified SHA-1 has increased the message digest from 160 bits to 192 bits.

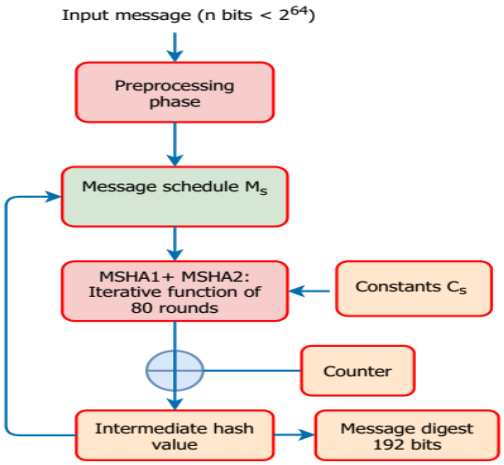


Figure 1. Overview of modified SHA-1 construction

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Adding SALT to Hashing:

Salting hashes seems like a step in the hash browns recipe, but cryptography refers to applying random data to the hash function input to ensure a unique output, a hash, while the inputs are the same. The unique salt hash will also secure us from various attack vectors, including hash table attacks, and slow down the dictionary and brute-force offline attacks. That being said, the protections that salt may offer are restricted. When the intruder hits on an online site which is a brute force attack, a subset, salts won't support since the legit server is trying to hammer and salting for us. When the attacker hits an online service.

Mitigating Password Attacks with Salt

In order to reduce the damage that a hash table or dictionary may do, we salt the passwords. Salt is a value that is created by a stable cryptographic algorithm, which is used to construct a specific hash function for any data, irrespective of the input itself not unique. A salt would not look probabilistic to a hash feature, which is better because, by our hashing, we don't want to see duplicate passwords.

Competences of both MSHA- 1 and MSHA-2:

The intention here is to combine MSHA-1 and MSHA-2 capabilities, with the usage of SALT MSHA-1, for example, by inserting an arbitrary hash value of 128-bit. Some cases have occurred in MSHA-1 collisions and are now known to be small. Similarly, MSHA-2 accepts random input and generates the hash value of 256-bit as there has been no occurrence of collisions in MSHA-2 so far. Thus, MSHA-2 was secure than MSHA-1.

Suppose that the sample data is in gigabyte range. And the hash value of certain data must be computed. We fixed hash values even if a large amount of information is compressed. Several inputs may be the same hash values. That is to say; collisions can occur. Consider the equation below,

X = MSHA1(Inputdata + salt1) (1)

This is not a really important equation per se. This means that it does not resist collisions simply by introducing salt1 into the above equation. Now, look at a certain equation,

H = MSHA2(X + salt2) (2)

Equation (2) has a few benefits. They're like this,

1) In equation (1), the intermediate hash value is sent to MSHA2, along with salt2 in the equation; (2)

2) While equation (1), for any other input details, will generate the same hash value, salt2 in equation (2) secures against more collisions.

3) MSHA-2 does not compress but extends the files. As a result, no collisions because of the equation will occur (2)

Take the example below, X1 = MSHA1(inputdata1 + salt1A) X 2 = MSHA1(inputdata2 + salt1B)

Here, X 1 = X 2 is a possibility

Assume, X 1 = X 2

H1 = MSHA2(X 1 + salt2A)

H2 = MSHA2(X 2 + salt2B)

H1 not equal to H2

1) Even though X1 = X2, it would change salt2A and salt2B

2) In addition, MSHA2 does not compress input data. Instead, the input data is inflated into 256 bits. In the corresponding block diagram as seen in figure 2, the working of the proposed hashing scheme is also shown.

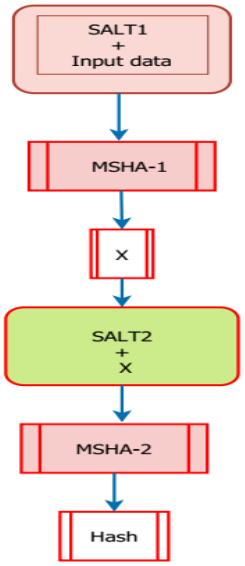
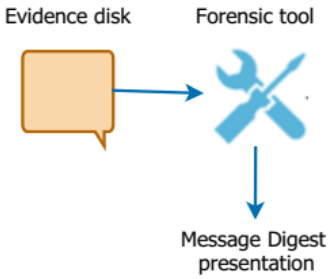


Figure 2: Combined effect of MSHA-1 and MSHA-2 along with salt

We used hashing in the preservation and analysis processes as digital forensics. Forensic methods are used to measure digital evidence hash values. Figure 3 shows a high-level block diagram that illustrates hash value measurement using the forensic method.



,

Figure 3:Hash value computation using a forensic tool

4. RESULTS AND DISCUSSION:

Let's reflect we have password **agri2020NO1** and the salt **a1lmgder201**. We can salt that password by either appending or prepending the salt to it.

Hashing and Salting User 1 Password (User 1: Alex)

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Figure 4: Hashing operation over User 1

Hashing and Salting User 2 Password (User 2: Brisson)

Figure 5: Hashing operation over User 2

The same password, various users. Various salts, various hashes. No one may say that Alex and Brisson have used the same password by looking at the entire set of password hashes. Every single salt expands the agri2020NO1 password into a single password. Besides, the service can also generate new salt when a user changes his password.

5. CONCLUSION

This study summarized the importance of hash functions in digital forensics. It made us learn the ability of hash functions to identify even the slightest input changes. The validity of the digital evidence with which the hashing function checked its integrity is the major problem for digital forensics. Digital forensic tools use either MSHA-1 or MSHA-2 in combination with salt series, reducing and improving collisions' complexity. Salts allow us to mitigate attacks in the hash table by requiring attackers to re-calculate them for each user's salts. It isn't easy to build cryptographically strong random data to use as salts and is left entirely to leading providers and security solutions.

REFERENCES

[1] Tomb, Aaron. (2017). Automated Verification of Real-World Cryptographic Implementations. IEEE Security & Privacy. PP. 1-1. https://doi.org/10.1109/MSP.2017.265093349.

[2] Mouha, Nicky & Hailane, Asmaa. (2021). The Application of Formal Methods to Real-World Cryptographic Algorithms, Protocols, and Systems. Computer. 54. 29-38. https://doi.org/10.1109/MC.2020.3033613.

[3] Kobeissi, Nadim & Nicolas, Georgio & Tiwari, Mukesh. (2020). Verifpal: Cryptographic Protocol Analysis for the Real World. https://doi.org/10.1007/978-3-030-65277-7_8.

[4] Liang, Fang. (2020). AI-Powered Digital Media Platform and Its Applications. 121-126. https://doi.org/10.1145/3433996.3434018.

[5] Rao, Koduru. (2019). Information Security Using Hilbert With Hash Value. International Journal of Advanced Trends in Computer Science and Engineering. 8. 2507-2511. https://doi.org/10.30534/ijatcse/2019/96852019.

[6] Wang, Jielin. (2021). HASH Algorithm of Weighted Probability Model. https://doi.org/10.36227/techrxiv.13606373.v1.

[7] C, Pradeep & Soman, Rajashree & Honnavalli, Prasad. (2020). Validity of Forensic Evidence using Hash Function. 823-826. https://doi.org/10.1109/ICCES48766.2020.9138061.

[8] Al-Odat, Zeyad & Khan, Samee. (2019). The Sponge Structure Modulation Application to Overcome the Security Breaches for the MD5 and SHA-1 Hash Functions. 811-816. https://doi.org/10.1109/COMPSAC.2019.00119.

[9] Suhaili, Shamsiah & Watanabe, Takahiro. (2019). High-Throughput of SHA-256 Hash Function with Unfolding Transformation. Global Journal of Engineering and Technology Review. 4. 73-81. https://doi.org/10.35609/gjetr.2019.4.4(1).

[10] Huo, Yuanhong & Liu, Dake. (2016). A High-Throughput Processor for Cryptographic Hash Functions. Journal of Communications. 11. 702-709. https://doi.org/10.12720/jcm.11.7.702-709.

[11] Gauravaram, Praveen & Millan, William & Neito, Juanma. (2021). Some thoughts on Collision Attacks in the Hash Functions MD5, SHA0 and SHA1.

[12] Hammadi, Yousef & Fadl, Mohamed. (2019). Reducing Hash Function Complexity: MD5 and SHA-1 as Examples. International Journal of Mathematical Sciences and Computing. 5. 1-17. https://doi.org/10.5815/ijmsc.2019.01.01.