e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:05/Issue:12/December-2023 Impact Factor- 7.868 www.irjmets.com

# A REVIEW OF VARIOUS SECURE HASH ALGORITHMS FAMILY

**Ibrahim H.S. Mahmoud*[1], Ahmed I. Galal*[2], Adel A. Elbaset*[3]**

*[1,2]Electrical Engineering Department, Faculty Of Engineering, Minia University, El-Minia, Egypt.

*[3]Department Of Electromechanics Engineering, Faculty Of Engineering, Heliopolis University, Cairo, Egypt.

DOI : https://www.doi.org/10.56726/IRJMETS47726

## ABSTRACT

Hash functions are essential tools for online data security. Cryptographic hash capabilities are those hash functions that are employed in various security related applications. They take into account the varying lengths of subjective information and produce a message process, which is a typically small, settled size yield. They are designed to provide message uprightness, which means that in the unlikely event that the message has been altered after it has been transmitted from the sender and before it could be obtained by the recipient who is comparing, the collector can follow the altered message and proceed accordingly, disposing of the altered message. This property is also useful in many other applications, such as the creation of computerized signatures and arbitrary number ages, among others. The great majority of hash functions, such as SHA-1, SHA2, SHA3, and so on, rely on Merkle-Damgard development and are thus unquestionably not completely impervious to attacks. This is a review paper that includes comparisons between different secure hashing algorithms.

**Keywords:** SHA-0, SHA-1, SHA-2, And SHA-3, Message Digest, SHA Functions, Message Authentication.

## I. INTRODUCTION

Digital signatures, message authentication codes (MACs), and other types of authentications are only a few examples of the various information security uses for cryptographic hash functions. Additionally, they can be used as standard hash functions, as data indexes in hash tables, for fingerprinting, to find duplicate data or uniquely identify files, and as checksums to find unintentional data damage. Cryptographic hash values are sometimes referred to as (digital) fingerprints, checksums, or simply hash values in information security contexts, even though all of these terminologies refer to more generic functions with somewhat distinct characteristics and objectives.

A cryptographic hash function is a hash function that accepts an arbitrary block of data and outputs a fixed-size bit string, the cryptographic hash value, with the property that any modification to the data (deliberate or unintentional) will almost certainly change the hash value. The hash value is commonly referred to as the message digest or just the digests, while the data to be encoded is frequently referred to as the message. These techniques mimic the block cipher modes of operation that are typically employed for encryption. All well-known hash functions, including MD4, MD5, SHA-1, and SHA-2, are constructed from specially created block-cipher-like components with feedback to assure that the final function is not invertible.

## II. REVIEW OF VARIOUS SECURE HASH ALGORITHMS FAMILY

### 2.1 SHA-0:

A phrase used to refer, in retrospective, to the 1993 release of the 160-bit hash function under the name "SHA." Afterwards, it was abandoned because to an unidentified "significant flaw, " it was replaced by the greatly modified version of SHA-1. The slightly revised version of SHA-1 took its place [1].

### 2.2 SHA-1 Family:

SHA-0 is the initial iteration of the 160-bit hash algorithm. Though it modifies the original SHA hash specification to address purported flaws, SHA-1 is remarkably similar to SHA-0. The most popular SHA hash function now in use, SHA-1 [1, 5], is included into a number of extensively used protocols and applications.

Though it was designed using a more conservative approach, SHA-1 generates a message digest based on ideas that Ronald L. Rivest of MIT used to build the MD4 and MD5 message digest algorithms. The only bitwise rotation that separates SHA-1 from SHA-0 is in the message schedule of the compression function; the NSA claims that this was done to fix an issue with the previous algorithm that made it less secure cryptographically. Nevertheless, the NSA did not identify the fixed flaw or offer any additional explanation. Subsequent reports of vulnerabilities have been made in SHA-0 and SHA-1. The NSA's claim that the update improved security seems to be supported by SHA-1's increased resistance to attacks.

Many popular security programs and protocols, such as TLS and SSL, PGP, SSH, S/MIME, and IPSec, contain SHA-1 [6].

Given that SHA-1 and MD5 are both descended from MD4, those applications can also employ MD5. Distributed revision control systems like Git, Mercurial, and Monotone also employ SHA-1 hashing to recognize revisions and spot data corruption or manipulation. The algorithm is also utilized for signature verification during booting on Nintendo's Wii game console; however, a serious implementation error makes it possible for an attacker to get around the system's security measures.

Although SHA-1 hasn't been cracked by anyone, the important thing to remember is that Git doesn't even consider SHA-1 to be a security feature. It is only an attempt at consistency. Since the security components are located elsewhere and Git employs SHA-1, many people believe that Git is cryptographically safe.

Owing to the algorithms' block and iterative design as well as the lack of further finishing stages, length-extension and partial-message collision attacks can affect all SHA functions.

### 2.2.1 SHA-1 structure:

- Expand the original message by adding pauses. To make the original input 448 mod 512, or 64 bits short of any multiple of 512, you will lengthen its total length.
- Finish the padded message with some length. After this operation, the message's total bit count becomes a multiple of 512, or 64 bits.
- Set up MD buffers so that the message digest can be calculated. A lengthy string of 32-bit words and two buffers are needed for this algorithm:
  - Specific initial values are assigned to each of the two five 32-bit registers ("A, B, C, D, E" and "H0, H1, H2, H3, H4").
  - The 80 32-bit word sequence (W[0], W[1], W[2]... W[68], W[69]).
- Proceed with processing the message in blocks of 512 bits each. Every block goes through eighty rounds of compression, each consisting of twenty steps, and a difficult process of expansion. The current buffer (hash state) is expanded with the value that is obtained following each compression.
- Generate the final hash value of 160 bits. The final hash value output is the hash state as of the moment the last block is processed.

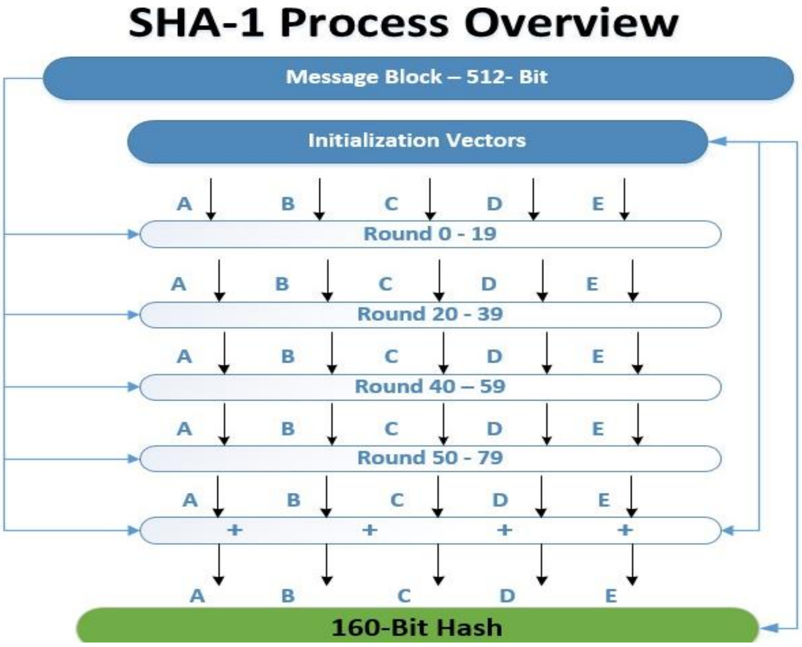

**Figure 1:** A simplified overview diagram that illustrates how the SHA-1 hashing algorithm works

### 2.2.2 SHA-1 Advantage:

- This algorithm is sluggish. It was helpful for storing password hashes because of this feature, which slows down brute force attacks.
- It can be used to compare codes or files in order to find inadvertent corruptions alone.
- can take the place of SHA-2 in the event that legacy codes don't work with it.

### 2.2.3 SHA-1 Disadvantage:

- Slower than other algorithms, which makes it inappropriate for many uses other than storing passwords (like comparing files or creating secure connections to websites).
- less secure, with numerous vulnerabilities discovered over time.
- Finding collisions is simple and inexpensive.
- Too short of a key for resisting attacks.

### 2.3 SHA-2 Family:

A group of cryptographic hash routines is called SHA-2. The SHA-2 variations include SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. SHA-2 differs greatly from SHA-1 in a number of ways. There are now six hash algorithms available for SHA-2, with digest sizes of 224, 256, 384, or 512 bits [10].

New hash functions called SHA-256 and SHA-512 are calculated using 32-bit and 64-bit words, respectively. With the exception of the number of rounds, their structures are nearly identical, although they employ different shift amounts and additive constants. SHA-224 and SHA-384 are essentially abbreviated variants of the initial two, calculated using distinct starting values. Both SHA-512/224 and SHA-512/256 are abbreviated variants of SHA-512[3.6], however the FIPS PUB 180-described technique is used to generate the initial values. SHA-1 was found to have four security flaws, one of which suggested the possibility of a mathematical weakness and the need for a more robust hash function. Despite certain similarities between SHA-2 and SHA-1, these attacks have not been effectively extended to SHA-2.

The most effective public attacks at the moment break preimage resistance after 52 or 57 SHA-512 rounds as well as collision resistance after 46 SHA-256 rounds.

Three more hash functions were added to the SHA family by NIST. The three algorithms—SHA-256, SHA-384, and SHA-512—are collectively referred to as SHA-2 because of their respective digest lengths (in bits).

The SHA-1[1, 4] algorithm was included in the revised standard, along with updated technical notation that was in line with the description of the SHA-2 family's internal operations. defining SHA-224, a further variant whose key length is the same as that of two-key Triple DES. Although SHA-224 from the change notice was added to the standard in FIPS PUB 180-3, no other significant changes were made to the standard. The main driving force behind the standard's update was the transfer of security data regarding hash algorithms and usage guidelines to Special Publications 800-107 and 800-57. Furthermore, the limitation on padding the input data before hash computation was eliminated, enabling hash data to be computed in tandem with the creation of content, like a real-time audio or video stream. The last data block must still be padded before the hash output.

After 2013, the publication forbids the creation of digital signatures with a hash security of less than 112 bits. The deadline was set for the end of 2010 in the previous revision from 2007.NIST made the same revisions to SP800-107 in August 2012. In 2012, SHA-3 was chosen as the winner of the NIST hash function competition. There is no derivation of SHA-2 algorithm from SHA-3.

### 2.3.1 SHA-2 (256) structure:

- Expand the original message by adding pauses. To make the original input 448 mod 512, or 64 bits short of any multiple of 512, you will lengthen its total length. Thus, to get 448 bits, you'll need to add a 1 and a lot of 0s.
- Finish the padded message with some length. To make the padded message a multiple of 512, 64 bits are added at the end.
- Set up the MD buffer in order to calculate the message digest. A, B, C, D, E, F, G, and H are the eight 32-bit registers that represent the buffer.

- Proceed with processing the message in blocks of 512 bits each. The message is divided into 512-bit chunks, and each chunk undergoes 64 compression rounds and a complicated process. The current hash value is increased by the value that is obtained following each compression.
- Generate a final hash value of 512 or 256 bits. All of the chunk values that come from the processing step are concatenated, or linked together, to create the final hash value or digest.

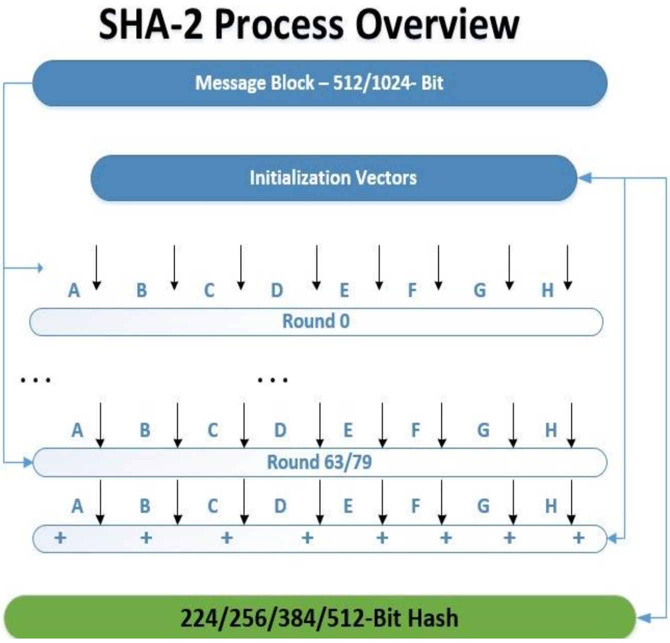

**Figure 2:** A simplified diagram that illustrates how the SHA-2 hashing algorithm works.

### 2.3.2 SHA-2 Advantage:

- It can withstand collisions as well as attacks using preimages and second preimages.
- It fixes the weakness in SHA-1.
- The mail clients, OS platforms, browsers, and mobile devices all support SHA-256.
- It is the ideal option for verifying and signing digital security files and certificates since it produces a longer hash value and provides a higher level of security.

### 2.3.3 SHA-2 Disadvantage:

- Compared to its predecessors, SHA-256 is slower.
- For certain software to support SHA-2 encryption, updates may be required.

## 2.4 SHA-3 Family

The most recent member of the SHA family is SHA-3. It is entirely distinct from MD5, SHA-1, and SHA-2, but it is a component of the same standard, having been developed through a public competition supported by NIST [1, 11].

The foundation of SHA-3 is a novel cryptographic technique developed by Keccak called sponge construction. Similar to how a sponge absorbs and releases water, the data are essentially "absorbed" into the sponge and then "squeezed" out.

It is noteworthy that NIST regards SHA-3 as an enhancement to its entire set of hash algorithms, rather than a complete replacement of SHA-2.

The SHA-3 family of algorithms consists of four distinct hash functions and two expandable output functions. These algorithms can be applied to stream encryption, domain hashing, randomized hashing, and MAC address generation: SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128 (output function), and SHAKE-256 (output function).

### 2.4.1 SHA-3 structure:

- Expand the original message by adding pauses. In this manner, the total length precisely multiplies the rate of the associated hash function. Since we've selected SHA3-224 in this instance, it needs to be a multiple of

1152 bits (144 bytes). The two primary action categories that comprise the SHA-3 process are "absorbing" and "squeezing, " which we will address in the following sections.

- To begin computing the hash value, absorb the padded message values. Blocks of the padded message have a fixed size. Next, a total of 24 permutation rounds consisting of five operations are performed on each block. Ultimately, an internal state size of 1600 bits is obtained.
- Squeeze the hash value out. This is the point at which the message is extracted. Based on the associated rate and capacity (the “r” and “c” we mentioned in the image caption above), the 1600 bits obtained with the absorption operation are divided.
- Generate the hash value in the end. Ultimately, the first 224 bits (the rate of SHA3-224) are taken out of the 1152 bits. The entire message's hash digest is contained in the extracted value, which is 224 bits.

SHA-3 Process Overview

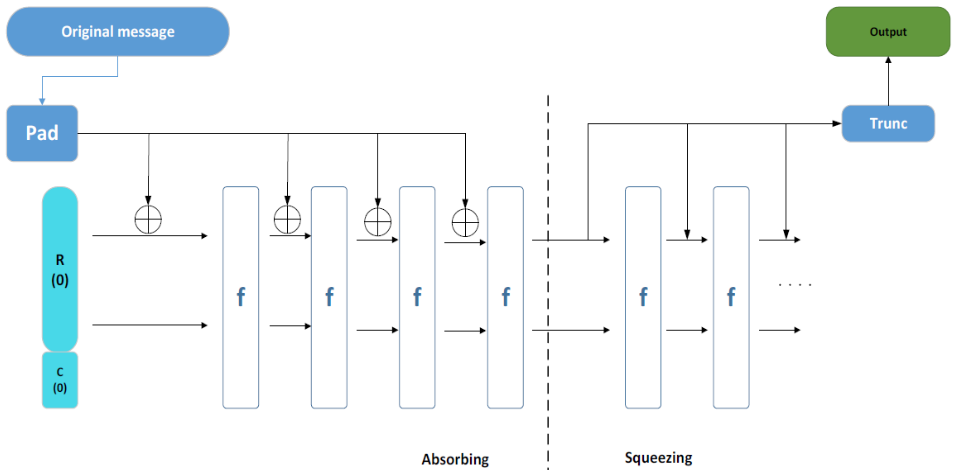

**Figure 3:** A simplified diagram that illustrates how one of the SHA-3 hashing algorithms works.

### 2.4.2 SHA-3 Advantage:

- It is perfect for a hash function because it is flexible enough to accommodate different input and output lengths.
- Its instances reduce implementation costs by using a single permutation for all security strengths.
- Performance-security trade-offs are made possible by the SHA3 family of algorithms, which selects the appropriate capacity-rate pair.
- Not susceptible to attacks using length extensions.
- Significantly faster than its predecessors when hardware is used to handle cryptography.

### 2.4.3 SHA-3 Disadvantage:

- Vulnerable to attacks by collisions.
- Much slower than SHA-2 (problem limited to software).
- Insufficient software and hardware support.

## III. SECURE HASH ALGORITHM COMPARISON

### 3.1.1 Secure Hash Algorithm Comparison based on general properties:

The standard hash algorithm is compared based on general properties as indicated in Table 1, such as block size, word size, output size, logical operation, and number of rounds [1].

**Table 1.** Secure Hash Algorithm Comparison based on general properties [1]

| | SHA-1 | SHA-2 | SHA-3 |
|---|---|---|---|
| **Available Since** | 1995 | 2002 | 2008 |
| **Block Size** | 512 bits | 512/1024 bits | 1152/1088/8<br>32/576 bits (this is |

| | | | |
|---|---|---|---|
| | | | referred to as a Rate [R] for SHA-3 algorithms) |
| **Hash Digest Size (Output)** | 160 bits (i.e., 20 bytes), or 40 hexadecimal digits | 256 bits (i.e., 32 bytes), or 64 hexadecimal digits/512 bits (i.e., 64 bytes), or 128 hexadecimal digits | 224/256/384/512 bits (i.e., 28/32/48/64 bytes), or 56/64/96/128 hexadecimal digits |
| **Rounds of Operations** | 80 (4 groups of 20 rounds) | 64 (for SHA-224 and SHA-256)/80 (SHA0384/SHA-512) | 24 |
| **Construction** | Merkle–Damgård | Merkle–Damgård | Sponge (Keccak) |
| **Collision Level** | Cheap and easy to find as demonstrated by a 2019 study. | Low — No known collisions found to date. | Low |
| **Successful Attacks** | Yes, many. The first one called SHAttered happened in 2017. | SHA-256 has never been broken. | Few collision type attacks have been demonstrated. |
| **Common Weaknesses** | Vulnerable to collisions. | Susceptible to preimage attacks. | Susceptible to:<br>Practical collision.<br>Near collision attacks. |
| **Security Level** | Low | High | High |
| **Applications** | Previously widely used in TLS and SSL. Still used for HMAC (even if it's recommended to move to a more secure algorithm), and for verifying the integrity of files against involuntary corruption. | Widely used in:<br>Security applications and protocols (e.g., TLS, SSL, PGP, SSH, S/MIME, Ipsec)<br>Cryptocurrencies transactions validation<br>Digital certificates<br>Other applications | Used to replace SHA-2 when necessary (in specific circumstances). |
| **Deprecated?** | Yes | No | No |

### 3.1.2 Summary of Frequent Attacks Against Secure Hash Algorithm:

As shown table 2 provides an overview of common attacks against these algorithms.

**Table 2.** Summary of Frequent Attacks Against Secure Hash Algorithm

| **Algorithm** | | **Type of attacks** | **Complexity** |
|---|---|---|---|
| **SHA-1** | | Collision [12] | $< 2^{69}$ |
| | | Collision [13] | $2^{61}$ |
| | | Free start Collision [14] | |
| **SHA-2** | **256** | Preimage [15] | $2^{255.5}$ |
| | **512** | Preimage [15] | $2^{511.2}$ |
| **SHA-3** | **256** | Practical Collision and near-Collision [16] | |

| | | | |
|---|---|---|---|
| | **512** | Possibility first Collision [17] | |

As can be seen from the discussion above, the majority of well-known hash functions from various families are not only slow, but also vulnerable to collision attacks. Other algorithms were suggested by researchers as a fix for this issue.

### 3.1.3 Review of Various Secure Hash Algorithms:

Numerous researchers have put forth their own algorithms to address the aforementioned problem. The authors have covered a few of the variations in hash function algorithms in this section. A few of them relied on the chaotic design, while others were based on the chaotic neural network, the chaos tent map, and the complex chaotic system. A hash value size of only 128 bits is produced by some of the current hash function designs, which is insufficiently secure against collision attacks. These algorithms have provided acceptable statistical performances, but their ability to withstand collision attacks is still lacking. Because current algorithms are insufficient to meet the demands of modern technologies and security concerns, it is necessary to develop new approaches to hash function algorithm design that are able to prevent attacks effectively in comparison to existing algorithms.

**P. Zhang, X. Zhang, and J. Yu** [18] in this study proposes a parallel hash algorithm with variable beginning values. After first obtaining the new initial value of the chaining variable through a three-round iteration, we split a message into 512-bit message blocks at random. Next, we combine the two adjacent blocks to assign all blocks to be calculated in parallel. Several computation rounds are needed to determine the final hash value. The suggested approach meets the hash function's performance requirements. The function's security is ensured, and the collision resistance property is enhanced by adjusting the chaining variables' starting values. Efficiency analysis shows that the structure's improvement measure is effective.

**Belfedhal and Faraoun** [19] introduced a hash function algorithm that yields a 256-bit hash value using a variation of the Merkle Damgard construction based on cellular automata. The algorithm was not tested against collision and preimage attacks, despite producing good results for statistical tests.

**Li, Ge and Xia** [20] employed a dynamic S-box to create a chaotic hash function that compromises the S-box's flexibility and practicability by producing 128-bit hash values as the final hash code. This proposed algorithm's main flaw is that there isn't enough hash code length to provide security against collision or second preimage attacks.

A new hash function based on MD5 was created by **Abdulah, Al-rawi and Hammod** [21], yielding a 224-bit hash value. The fact that the message digest takes as long to produce as the MD5, indicating extremely low efficiency, may be the most significant drawback of this development.

**Tur and Javurek** [22] developed hash function generation using neural networks, yielding a hash value of 128 bits. These kinds of approaches, while having a small hash value, are also very challenging to implement.

A unique hash function scheme was developed by **Ahmad, Alsharari and others** [23] using integrated 2D and 1D chaotic maps, resulting in the generation of a 128-bit hash value for messages of any length. However, because of its short length, the hash value is not immune to collision attacks.

A hash function was proposed by **Li and Liu** [24] using generalized chaotic mapping and variable parameters. In their work, a 6-unit iteration with variable message values and parameters was executed, converting an arbitrary length message to corresponding ASCII values. Iteration state values were used to cascade extracted bits toward the final hash value. A definition of the birthday attack indicates that a 128-bit hash value is insufficient to ensure algorithm security.

A hash algorithm was proposed by **Wang and Li** [25] that produced a variable size digest. An enhanced version of the MD-5 algorithm on the output length was the algorithm they suggested. The suggested algorithm still employs the same construction as MD5, despite having a variable size digest, which is a useful feature to boost security.

**D. K. N and Bhakthavatchalu** [26] The Plan of Parameterizable Execution of SHA-256 algorithm in FPGA providing Blockchain Ideas is the main focus of this work. The main principle in Blockchain design that adds security and protection to a structure is SHA-256. The suggested method enables any message with a piece length to be converted to a fixed length message overview called a hash. Using Artix 7 FPGA, the proposed engineering plan was replicated in Modelsim and integrated into Xilinx Vivado Structure Suite.

**Yahya, and others** [27] Nowadays, a notable test is verifying a large number of related, asset-obligated processing devices. To further complicate matters, outside specialist cooperatives require regular access to the framework. A small number of business processors support secure boot in order to ensure the integrity of the framework and the legitimacy of the product merchant. In this composition, we suggest a secure boot engineering method based on lightweight equipment. The engineering makes use of Direct Memory Access (DMA), the Secure Hash Algorithm 3 (SHA3) hashing algorithm, and the Elliptic Curve Digital Signature Algorithm (ECDSA).

**Sengupta and Rathor** [28] A key component of consumer hardware devices is a licensed innovation (IP) center built around digital signal processing (DSP) technology. Thus, protection of these IP centers from reconnaissance attacks is crucial. To counter this equipment danger, functional confusion serves as an excellent tool. The suggested method, which makes use of custom equipment and the lightweight secure hashing algorithm (SHA-512) for key encryption, rearranges the keybits of the securing rationale, which is a functionally jumbled DSP configuration expanded with the overall rationale amalgamation of the structure.

**H. Liu, Kadir and J. Liu** [29] A major role for the secure hash function exists in cryptography. The hyperchaotic Lorenz framework, which serves as a wipe function to preserve information message through various parameters time-fluctuating bother, is used in this work to build a hash algorithm. Since the annoyed parameters are still within their critical interims, the framework remains in a hyperchaotic state. Initially, the information message is isolated into four 1D clusters, producing four bother groupings through parameter refreshing standard. The evaluation and analysis of the trial demonstrated that the hash function is resistant to both differential attack and second pre-picture attack. The figure signature, information trustworthiness, and recognizable proof can all be linked to the suggested hash function.

**Biswas, Gupta and Haque** [30] In this focused world, protecting information or information has become a test. Information can be verified using a variety of techniques, such as steganography and cryptography. Through the use of RSA and AES, hybrid cryptography has been connected in this work. A higher level of security is ensured by this hybrid cryptography since the symmetric key used for message encryption is also encoded. An additional component of this work involves creating a digital signature by jumbling the message's hash estimate. This digital signature is used for verifiability checks at the accepting end. In this case, steganography strengthens the security provided by hybrid cryptography. One remarkable feature of this algorithm is the message uprightness checking. It appears that effective reenactments support the algorithm's potential.

**Kundu and Dutta** [31] Applications built on top of cryptographic hash functions can assist a system administrator in identifying changes to important data on their network. Cryptographic hash functions are widely used in the field of computer security. These ideas are especially important in the expanding online world, where each file stored on a server is a valuable resource and every message sent over the wire has the potential to be worth money. Without protections like hash functions provide, data would be much more open to attack. Since there is a new attack on the hash function world almost every day, hash function security has become a hot topic. This paper presents a thorough analysis of cryptographic hash functions, including their properties, applications, and thorough classification. Additionally, it analyzes some attacks on particular hash functions and provides a general classification of all potential hash function attacks.

**Patel** [32] Cryptography plays a vital and critical role in achieving the primary aims of security goals, such as authentication, integrity, confidentiality, and no-repudiation. Cryptographic algorithms are developed in order to achieve these goals. Cryptography has the important purpose of providing reliable, strong, and robust network and data security. In this paper, they demonstrated a review of some of the research that has been conducted in the field of cryptography as well as of how the various algorithms used in cryptography for different security purposes work. Cryptography will continue to emerge with IT and business plans in regard to protecting personal, financial, medical, and ecommerce data and providing a respectable level of privacy.

**Gupta** [33] it observed a number of characteristics and traits of different hashing techniques. We observed the families of DES, RSA, MD, and SHA. We also studied the principles of cryptography and the various types of keys used in encryption. Here, shortcomings or limitations of numerous algorithms were also investigated, such as the fact that DES is inappropriate for encrypting sensitive data and that selecting large p and q in RSA can be difficult.

**Abdelbasit & Varol** [34] In this instance, the history of cryptography, the constantly evolving need for algorithms, and the role that these algorithms play in digital security were all covered. This article also covered some historical algorithms, such as the stream cipher, caesar cipher, simple substitution cipher, transposition cipher, and contemporary hash algorithms.

**Joseph and others** [35] Cryptography is essential to achieving the main goals of protecting the confidentiality, integrity, and authenticity of data. The goal of the cryptography algorithms is to facilitate the accomplishment of goals. Because of the regulations, calculating the hash price takes time. With the goal of advancing IT and business objectives, cryptography will continue to provide a reasonable level of privacy while safeguarding financial, medical, e-commerce, and personal information. By creating web and Android applications and customizing their user interfaces to meet our needs, we hope to build on this work and expand upon it. This study compares and contrasts different secure hashing techniques and their variants. The hash value must be calculated for each algorithm, which takes time. by figuring out how long each of these algorithms would take to complete, and then choosing the one that would compute the hash value the fastest.

**Verma and others** [36] Hashing algorithms play a critical role in achieving the main objectives of safeguarding data confidentiality, integrity, and authenticity. The algorithms used in hashing are designed to help achieve specific objectives. The hash price computation is time-consuming due to the set of rules. With regard to safeguarding personal, financial, health, and electronic trade data as well as offering a respectable level of privacy, cryptography will continue to gain prominence in IT and business enterprise plans.

**Babalola and Others** [37] In this study, they went over the concept of cryptography. Both benefits and drawbacks were mentioned. The reason cryptography is complicated is that, although computers can only recognize 0 and 1, humans can recognize numbers as digits from 0 to 9. As a result, binary systems use bits rather than digits. After that, the user will have a fair idea of what it represents. This is the reason that becoming a cryptographer requires such a high level of education.

## IV. CONCLUSION

In this paper, various secure hashing algorithms are compared with one another. The hash value computation takes time for each algorithm. We can combine the best secure hashing algorithm with network security algorithm to increase the security of the data being sent by calculating the time required for each of these algorithms and identifying the algorithm that will require the least amount of time for hash value computation.

Although it is evident that organizations cannot revert to a world where the digital and physical realms are kept apart, there are strategies to deal with the formidable risks posed by rapidly advancing technology and this new, hybrid environment.

There are other security solutions you should have in your organization's defense arsenal besides hash algorithms. But they are unquestionably a crucial component of it. With the aid of this comparison of hash algorithms, we hope you are able to gain a deeper understanding of the realm of secure hash algorithms and determine which hash functions work best for you.

While there is never a perfect hash algorithm, they are continuously being enhanced to fend off attacks from ever-more-skilled threat actors. Don't wait any longer; make sure your security strategy and implementations use the appropriate hash algorithms. Future-proof your information and structure right now.

## V. REFERENCES

[1] Federal Information Processing Standards Publication, Fips Pub 180-4, Secure Hash Standard (SHS), August 2015, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf

[2] "Crypto++ 5.6.0 Benchmarks". Retrieved 2013.

[3] Marc Stevens, "Cryptanalysis of MD5 & SHA-1", 2012

[4] Stevens, Marc; Bursztein, Elie; Karpman, Pierre; Albertini, Ange; Markov, Yarik (2017). Katz, Jonathan; Shacham, Hovav (eds.). The First Collision for Full SHA-1 (PDF). Advances in Cryptology – CRYPTO 2017.

[5] Critical flaw demonstrated in common digital security algorithm". Nanyang Technological University, Singapore. 24 January 2020.

[6] Henri Gilbert, Helena Handschuh: Security Analysis of SHA-256 and Sisters. Selected Areas in cryptography 2003

[7] Kasgar A. K., Agrawal Jitendra, Sahu Santosh, 2012, "New Modified 256 -bit MD5 Algorithm with SHA Compression Funct ion", IJCA (0975–8887) Volume 42 (12), pp47-51.

[8] Kahate, Atul, 2003, "Cryptography and Network Securit y", TataMcGraw -Hill, India.

[9] William Stallings, Cryptography and Network Security: Priciples and Practice, 5th Edition Prent ice Hall; 5 edition (January 24, 2010).

[10] Vandana P., V.K Mishra, Architecture based on MD5 and MD5-512Bit Applications, IJCA (0975 – 8887) Vol. 74– No.9, July 2013.

[11] Piyush Gupta et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 4492-4495

[12] X. Wang, Y. L. Yin, and H. Yu, ―Finding Collisions in the Full SHA-1, ‖ Int. Assoc. Cryptologic Res. 2005, no. 90304009, pp. 17–36, 2005.

[13] M. Stevens, ―New Collision Attacks on SHA-1 Based on Optimal Joint Local-Collision Analysis, ‖ Int. Assoc. Cryptologic Res. 2013, pp. 245–261, 2013.

[14] M. Stevens, P. Karpman, and T. Peyrin, ―Freestart collision for full SHA-1, ‖ EUROCRYPT 2016., vol. 2012, pp. 1–21, 2016.

[15] D. Khovratovich, C. Rechberger, and A. Savelieva, ―Bicliques for Preimages : Attacks on Skein-512 and the SHA-2 family, ‖ pp. 244–263, 2012.

[16] I. Dinur, O. Dunkelman, and A. Shamir, ―New attacks on Keccak-224 and Keccak-256, ‖ pp. 1–27, 2011.

[17] I. Dinur, O. Dunkelman, and A. Shamir, ―Collision Attacks on Up to 5 Rounds of SHA-3 Using Generalized Internal Differentials, ‖ no. 827, pp. 1–30, 2013

[18] P. Zhang, X. Zhang, and J. Yu, ―A Parallel Hash Function with Variable Initial Values, ‖ Wirel. Pers. Commun., vol. 96, no. 2, pp. 2289–2303, 2017.

[19] A. E. Belfedhal and K. M. Faraoun, ―Building Secure and Fast Cryptographic Hash Functions Using, ‖ J. Comput. Inf. Technol., pp. 317–328, 2015.

[20] Y. Li, G. Ge, and D. Xia, ―Chaotic hash function based on the dynamic S-Box with variable parameters, ‖ Nonlinear Dyn., vol. 84, no. 4, pp. 2387–2402, 2016.

[21] H. S. Abdulah, M. A. H. Al-rawi, and D. N. Hammod, ―Message Authentication Using New Hash Function, ‖ J. Al-Nahrain Univ., vol. 19, no. 3, pp. 148–153, 2016.

[22] M. Tur and M. Javurek, ―Hash Function Generation by Neural Network, ‖ in 2016 New Trends in Signal Processing (NTSP), 2016.

[23] M. Ahmad, S. Khurana, S. Singh, and H. D. Alsharari, ―A Simple Secure Hash Function Scheme Using Multiple Chaotic Maps, ‖ 3D Res., vol. 8, no. 2, pp. 1–15, 2017.

[24] Y. Li, X. Li, and X. Liu, ―A fast and efficient hash function based on generalized chaotic mapping with variable parameters, ‖ Neural Comput. Appl., vol. 28, no. 6, pp. 1405–1415, 2016.

[25] M. Wang and Y. Zhen Li, ―Hash Function with Variable Output Length, ‖ in 2015 International Conference on Network and Information Systems for Computers, 2015, pp. 3–6.

[26] D. K. N and R. Bhakthavatchalu, "Parameterizable FPGA Implementation of SHA-256 using Blockchain Concept, " 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0370-0374.

[27] J. Haj-Yahya, M. M. Wong, V. Pudi, S. Bhasin and A. Chattopadhyay, "Lightweight Secure-Boot Architecture for RISC-V System-on-Chip, " 20th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2019, pp. 216-223.

[28] A. Sengupta and M. Rathor, "Security of Functionally Obfuscated DSP Core Against Removal Attack Using SHA-512 Based Key Encryption Hardware, " in IEEE Access, vol. 7, pp. 4598-4610, 2019.

[29] H. Liu, A. Kadir and J. Liu, "Keyed Hash Function Using Hyper Chaotic System With Time-Varying Parameters Perturbation, " in IEEE Access, vol. 7, pp. 37211-37219, 2019.

[30] C. Biswas, U. D. Gupta and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography, "2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 2019, pp. 1-5.

[31] Rituparna Kundu and Ambar Dutta, Cryptographic Hash Functions and Attacks – A Detailed Study, International Journal of Advanced Research in Computer Science (IJARCS), ISSN No. 0976-5697, Volume 11, No. 2, March-April 2020.

[32] V. K. Patel, A Review Paper on Cryptography, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 10 Issue 02, February 2021.

[33] R. K. Gupta, A Review Paper On Concepts Of Cryptography And Cryptographic Hash Function. European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 07, Issue 07, 2020.

[34] Abdalbasit Mohammed, Nurhayat Varol. A review paper on cryptography. DOI:10.1109/ISDFS.2019.8757514, 2019.

[35] L. R. Joseph, Sreeji S., Aiswarya P. S., Sandhya S. And S. Kumar, Cryptographic Secure Hash Algorithm with Its Variants, International Journal of Creative Research Thoughts (IJCRT), ISSN: 2320-2882, Volume 11, 1 January 2023.

[36] Jai Verma, Md Shahrukh, Mukul Krishna and Ruchi Goel, A Critical review on cryptography and hashing algorithm SHA-512, International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), e-ISSN: 2582-5208, Volume:03/Issue:12/December-2021.

[37] Amosa Babalola, Ekuewa Bamidele, Olatunbosun Esther and Oyetunji Olumayowa, Cryptography: A Review, International Journal of Advances in Engineering and Management (IJAEM), ISSN: 2395-5252, Volume 3, Issue 10 Oct 2021.