iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 10 **Issue:** V **Month of publication:** May 2022

**DOI:** https://doi.org/10.22214/ijraset.2022.43370

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue V May 2022- Available at www.ijraset.com*

# Comparative Analysis of Cryptographic Hash Algorithms

Meena Preethi. B [1], Parameshvar. M [2]
[1, 2]*Department of Computer Science, Sri Krishna Arts and Science College*

***Abstract:*** ***Cloud is the on-demand availability for data storage management in large-scale companies and even for government sites where the data should be highly confidential. Here, data security plays an equivalent role to data storage. Cloud security is a collection of security measures that are designed to protect cloud-based infrastructure, data, and applications. The level of cloud security differs based on the type of cloud services. Cloud security protects the cloud environment from several attacks like distributed denial of attack, and malware, and prevents unauthorized user access. A crucial component of cloud security in large-scale organizations and for the government is focused on protecting the business contents like customer details, secret design documents, financial records, etc.***
***Keywords: MD5, SHA12, padding bits, buffers, salt hashes, keyed hashes, adaptive hashes.***

## I. INTRODUCTION

Cloud computing is a group of servers that allows users to store data in servers with the help of the Internet. The main advantages of cloud computing are users can store data without the help of separate hardware devices like SD cards, internal storage of any devices, and also users need not be in the same location while sharing data, whereas if it was internal storage physical presence of users is must for sharing or accessing the data. Apart from the advantages and goodness of cloud computing, data security becomes a bigger issue because all the data are stored in a platform that the user is not directly controlled or monitored. A cloud service is said to be secured only when it fulfills three conditions: Availability, Confidentiality, and Integrity. Availability means the user can access the data at any time and anywhere; Confidentiality means the data should be prevented from unauthorized people; Integrity means data received by the user should be in the same form as the user sends it. This paper discusses the performance analysis of the Message Digest 5 Algorithm (MD5) and Security Hash Algorithm (SHA), where both are cryptographic hash algorithms. Cloud security is done by the cryptography technique which converts data from readable form to unreadable form during the storage or transmission of data. The unreadable data is called ciphertext.

ENCRYPTION: PLAIN TEXT → CIPHERTEXT
DECRYPTION: CIPHERTEXT → PLAIN TEXT

## II. MESSAGE DIGEST 5 ALGORITHM

MD5 is a cryptographic hash algorithm that creates a 128-bit string value from an arbitrary length string. An MD5 file is a checksum file used to verify cloud data integrity. The checksum value is created from an algorithm based on the number of bits in the file. The final output obtained from the Message Digest 5 algorithm is called Message Digest Data.

### *A. Appending Padding Bits*

The process of adding extra bits to the original message at the end is called padding. In the MD5 algorithm, the data must be processed in 512-bit message blocks. So, if the length of the message is less than 512-bits, padding is done to the original message so that the length is 64 bits less than the exact multiple of 512 until its length becomes equal to 448 mod 512 and the bits are appended to the original plain text.

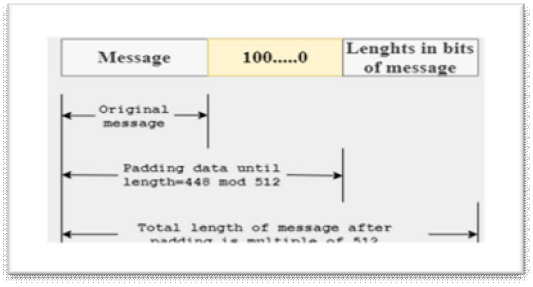



Fig. 1: Appending padding bits

*B. Dividing*

In this process, the messages are divided into n number of 512-bit sub-blocks where the messages will be in the exact multiple of 512.

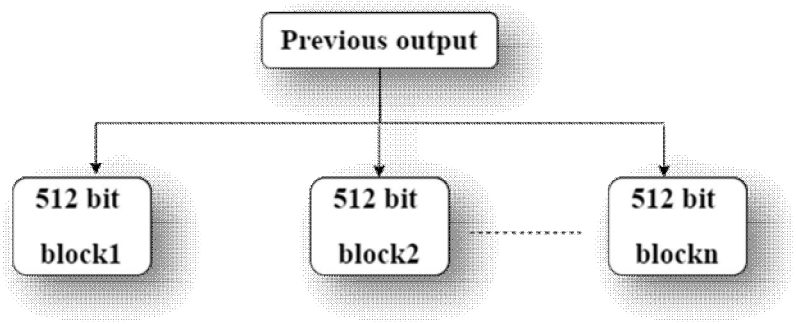



Fig. 2: Dividing appended text into blocks

*C. Initialising MD Buffers*

Four chaining variables namely A, B, C, and D are initialized where the length of each chaining variable is 32 bits and a total of 128 bits. These initialized chaining variables are used to process the data and to store the Message Digest (final output).

- *A → 01234567*
- *B → 89abcdef*
- *C → fedcba98*
- *D → 76543210*

*D. Processing*

All the divided 512-bit sub-blocks are processed using pre-initialized MD buffers. Next, the 512 bits sub-blocks should be divided further into 16-32 blocks, which means 16 blocks of each size are 32 bits.

$$F(B, C, D) = (B \wedge C) \vee ((\neg B) \wedge D) \quad \rightarrow \textbf{First round}$$

$$G(B, C, D) = (B \wedge D) \vee (C \wedge (\neg D)) \quad \rightarrow \textbf{Second round}$$

$$H(B, C, D) = B \oplus C \oplus D \quad \rightarrow \textbf{Third round}$$

$$I(B, C, D) = C \oplus (B \vee (\neg D)) \quad \rightarrow \textbf{Fourth round}$$

These are the four logical functions used. Each logical function is used for each round (16 steps) along with the equation: A ← B + ((A + Function (B C D) X[] + T[i])<<< s)

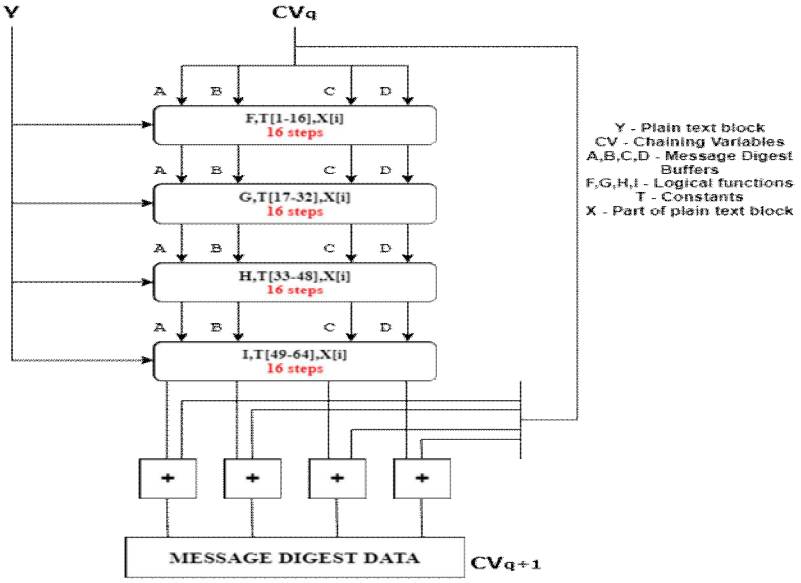



Fig. 3: MD5 process

Fig. 3 shows that the plain text block will initialize the buffers A, B, C, and D and was processed in four rounds. Each round has one logical function, 16 constants, and a part of a plain text block. Totally 64 steps are done for four rounds. After the fourth round, the initial values of buffers will be in addition modulo with the output of processed buffers and that is called the Message Digest. If need to proceed with the next block, the output of this block is given as input for the next block.

## III.SECURITY HASH 512 ALGORITHM

Security Hash algorithm is a modified version of the Message-Digest5 algorithm where the block size in MD5 is 128bits and the block size in SHA is 160-bits because of 4 buffers and 5 buffers respectively (Each of size 32 bits). There are several forms of SHA Algorithms that are determined by the size of the output.

128-BIT → SHA1 ALGORITHM, 256-BIT → SHA 256 ALGORITHM, 512-BIT → SHA 512 ALGORITHM

### *A. Padding bits*

The process of adding extra bits to the original message at the end is called padding. Pad the bits 1 followed by 0s (1000….) so that the length of the original plain text is 128 bits less than the multiple of 1024 bits.

### *B. Appending 128 bits*

Append a 128-bit representation to the original plain text, so that the length of the original plain text is equal to the exact multiple of 1024 bits.

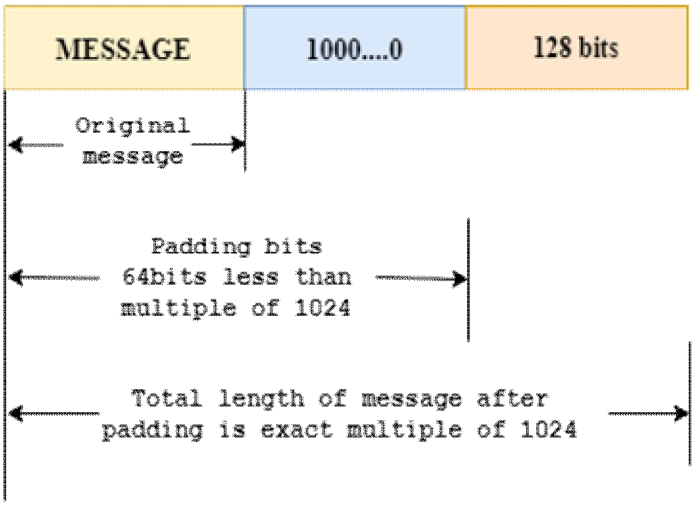



Fig. 4: Appending padding bits

### *C. Initialising Buffers*

Eight buffers are initialized namely A, B, C, D, E, F, G, and H where the length of each buffer is 64 bits and a total of 512 bits. These buffers are initialized in hexadecimal format. These buffers store intermediate and final output, known as results after each step and final hash code.

*a → 0*6A09E667F3BCC908*
*b → 0*BB67AE8584CAA73B*
*c → 0*3C6EF372FE94F82B*
*d → 0*A54FF53A5F1D36F1*
*e → 0*510E527FADE682D1*
*f → 0*9B05688C2B3E6C1F*
*g → 0*1F83D9ABFB41BD6B*
*h → 0*5BE0CD19137E2179*

*D. Processing*

Each block of plain text is processed in 80 rounds. In each round, there three inputs given for processing: they are, a block of plain text, eight initialized buffers, and a constant variable.

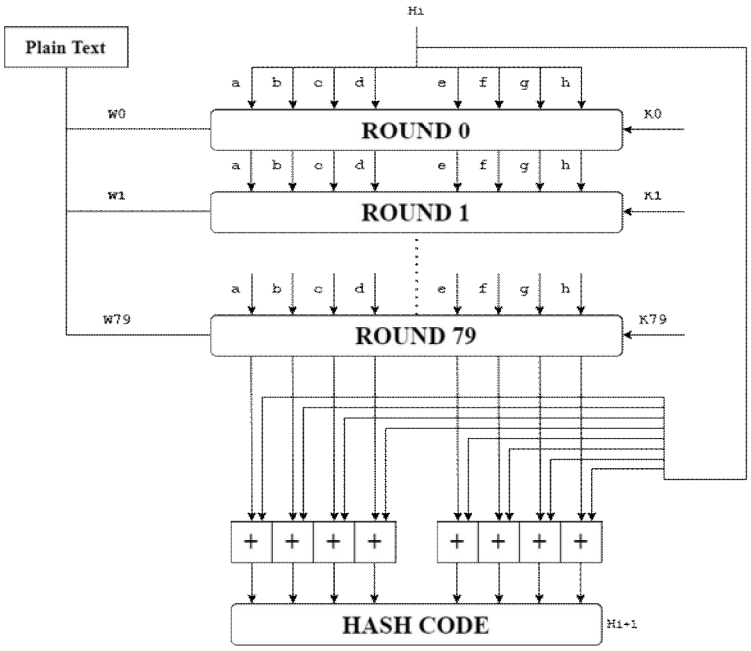



**Fig. 5:** SHA512 process

Fig.5 clearly shows that the plain text block will initialize the buffers A, B, C, D, E, F, G, and H and was processed in 80 rounds. After the last round, the initial values of buffers will be in addition modulo with the output of processed buffers and that is called the hash value.

The following formula along with its respective logical functions works using buffers and plain texts are used in each round of the process:

| | |
|---|---|
| $T_1 = H+con\ H(E, F, G) + (\Sigma_1^{512} E) + W + K$<br>$T_2 = (\Sigma_0^{512} A) + maj\ (A, B, C)$<br>$H = G$<br>$G = F$<br>$F = E$<br>$E = D + T_1$<br>$D = C$<br>$C = B$<br>$B = A$<br>$A = T_1 + T_2$<br>• con →conditional function<br>• maj →majority function<br>• $\Sigma 0^{512}$ A →Rotate A<br>• $\Sigma_1^{512}$ E →Rotate E<br>• W →Part of plain text<br>• K →Constant | Conditional Function: (E AND F) XOR (NOT E AND G)<br>Majority Function: (A AND B) XOR (B AND C) XOR (C AND A)<br>Rotate Functions:<br>Rotate A: ROTR28(A) XOR ROTR34(A) XOR ROTR29(A)<br>Rotate E: ROTR28(E) XOR ROTR34(E) XOR ROTR29(E)<br>• ROTR→Circular Right Shift |

Once the process is done for this block, then the output of this block is sent to the next block and acts as an input for that particular block. At the end of all blocks, the received output is the final output of the SHA 512 algorithm which is known as hash code.

## IV. COMPARATIVE ANALYSIS

### *A. Comparison between SHA algorithms*

| Algorithms | Output size (In bits) | | Internal state size (In bits) | Block size (In bits) | Max message size (In bits) | Word size (In bits) | Rounds | Bitwise Operations | Collisions found | Performance (bits/sec) |
|---|---|---|---|---|---|---|---|---|---|---|
| SHA 256 | 256 | | 256 | 512 | $2^{64} - 1$ | 32 | 64 | and,or, xor, not,rotr | None | 139 |
| SHA 512 | 512 | | 512 | 1024 | $2^{128} - 1$ | 64 | 80 | and,or, xor, not,rotr | none | 154 |

### *B. Comparison of MD5 & SHA512 Algorithms*

| Keys | MD5 | SHA512 |
|---|---|---|
| Block size | 128 bits | 160 bits |
| Number of rounds | 4 | 80 |
| Number of iterations | 64 | 80 |
| Speed | Faster than SHA 512 because fewer iterations | Slower than MD5 because more iterations |
| Attacks required to find out the original message | $2^{128}$ bits | $2^{160}$ bits |
| Security | Less secure than SHA512 | High secure than MD5 |

## V. CONCLUSION

This paper is the analytical and comparative study of two cryptographic hash algorithms, namely MD5 and SHA512, which give 128bits and 160bits hash values of any input size. As the encryption process of the SHA512 algorithm involves maximum steps and buffers to encrypt data than the MD5 algorithm, the encryption of SHA512 is stronger than MD5. Also, the analytical study proved that the decryption of SHA512 becomes more difficult than MD5 which means SHA512 is secured and tough to break plain text.

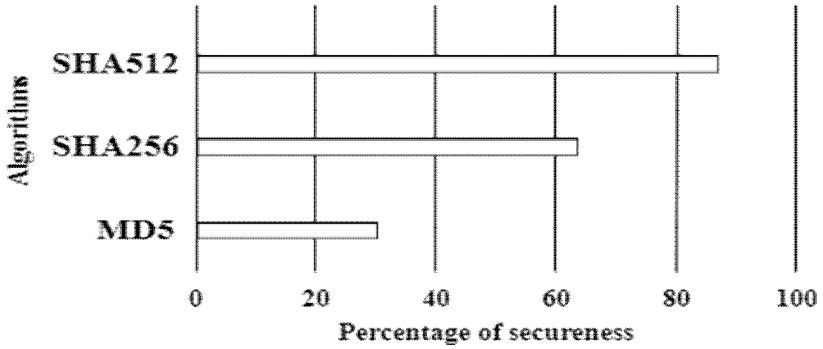



Fig. 6: Percentage of secureness

MD5 is no longer a secure way to store passwords and highly confidential data. Rather use SHA256 or SHA512 algorithms that provide better encryption and decryption techniques where breaking the plain text is almost impossible. If you still want to use MD5 algorithms to secure your database, just concatenate random strings such as b~/c1^12/*Nc along with the plain text where the process of concatenating random strings is called the salt method.

| Test strings | MD5 | SHA512 |
|---|---|---|
| cloud security | b692bb0826305047a235d7dda55ca2a0 | 18ce93bd9a528e14b49794c7956dd42d0 918a48843ad761df5495ea7b85b03e001 982a54a965e9177f25d0f26d9a9c024f4f 14417772aa4905f85ed69c49e85c |
| message digest | f96b697d7cb7938d525a2f31aaf161d0 | 107dbf389d9e9f71a3a95f6c055b9251bc 5268c2be16d6c13492ea45b0199f3309e 16455ab1e96118e8a905d5597b72038dd b372a89826046de66687bb420e7c |
| hash value | 6556fbe145a1ae3fff09d3ef697f13ea | ed0af435aa991c52d5ea94ef2f0883d922 7c74f0a4a8956f33f76aac663039e3a0db 7481c167edc4dcce9bc38ccb47f28d585e 4abdf14d0e1101840570b69779 |

## VI. FUTURE ENHANCEMENTS

Technology always grows rapidly, in both software and hardware fields. As any technology becomes outdated, it automatically gets failed to sustain in an environment. So, there should be always an update in any technology to keep overcoming the real-time problems and survive in the environment. For cloud security, the way and mode of attacks from the attackers are differing day by day which means the protection side should make their wall stronger. There are some ways to improve the protection skills of SHA algorithms which are to be discussed below:

### *A. Salted Hashes*

Salting is the mechanism of adding random data to the input of a hash function which will produce a unique output, even when the inputs were the same. Salting hashes methods can protect our data from several types of cloud attacks like brute-force offline attacks, hash table attacks, etc…

However, there are certain limitations to protections provided by the salting method. If the attacker is hitting an online service or offline service with a credential stuffing attack, salts can able to protect data because the legitimate server is doing hashing process for offline data.

### *B. Keyed Hash Functions*

Keyed hash functions are the functions that are part of cryptographic algorithms that create an authentication code based on both plain message and secret key which is to be shared between two endpoints. It is also known as HMAC – Hash Message Authentication Code. The message authentication code was keyed and hashed by using a cryptographic key and cryptographic function.

### *C. Adaptive Hash Functions*

Adaptive hash functions are designed to iterations of inner workings and feed the output back as input in a manner that causes a longer time to execute. It is known for adaptive functions because it can adapt with any number of iterations. The most popular adaptive hash function is PBKDF2. Along with that, some encryption schemes are taken into consideration such as bcrypt.

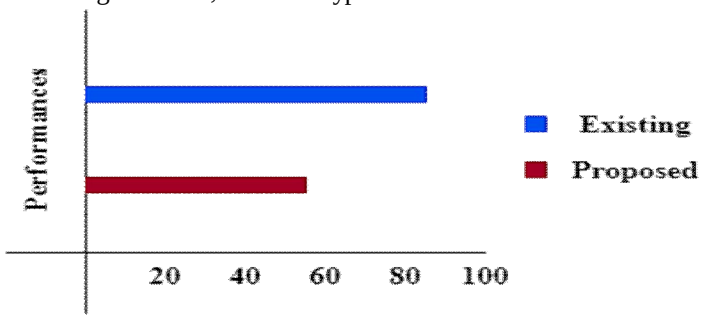
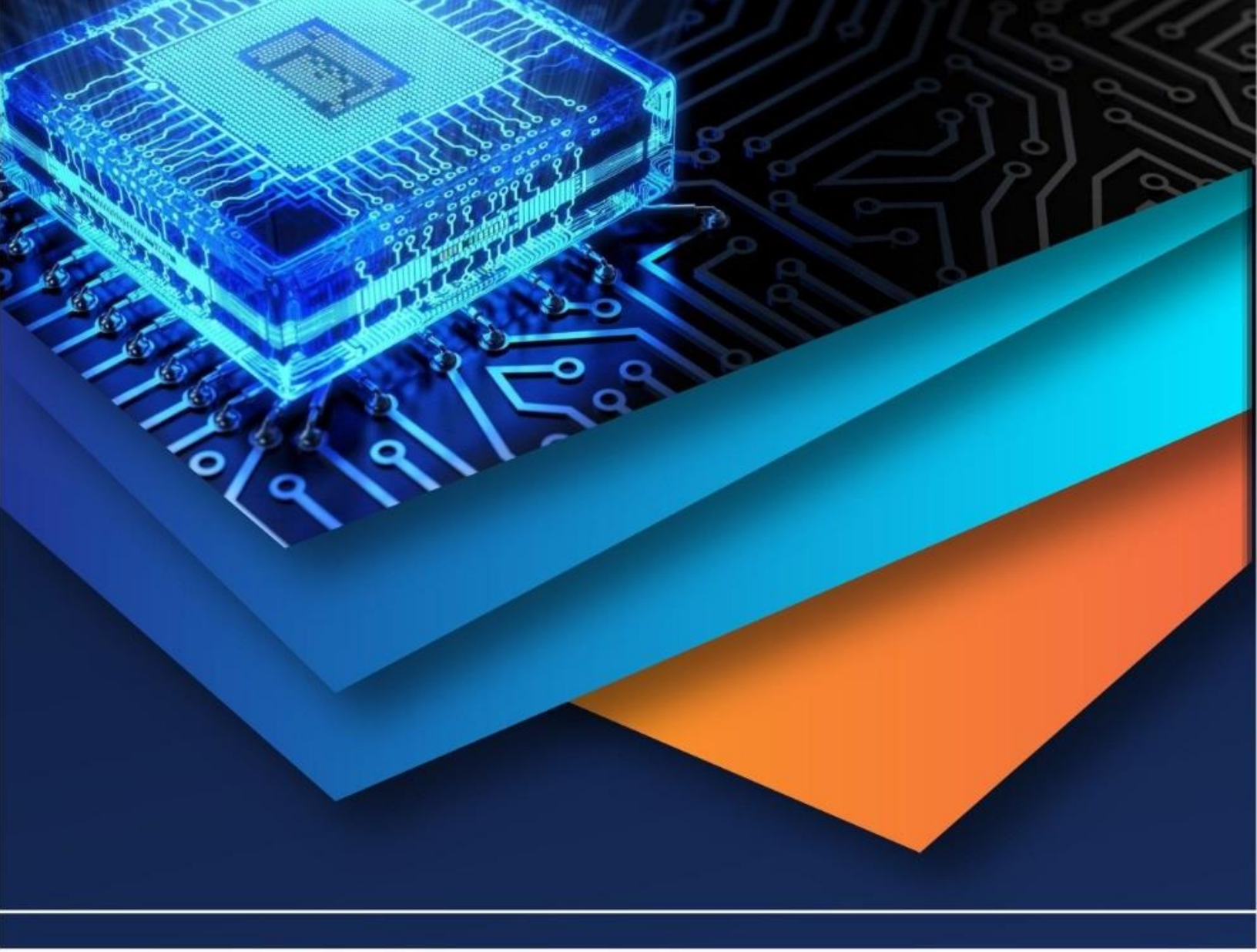



Fig. 7: Comparison of algorithms before and after salt, key, adaptive hashing techniques

## REFERENCES

[1] Security_Analysis_of_MD5_Algorithm_in_Password_Sto.pdf
[2] 23.AComparativeAnalysisofSHAandMD5Algorithm.pdf
[3] jcssp.2020.1439.1450.pdf
[4] https://www.synopsys.com/blogs/software-security/cryptographic-hash-functions/#:~:text=A%20cryptographic%20hash%20function%20is,used%20to%20verify%20the%20user.
[5] file:///C:/Users/PARAMESH/Downloads/gowthampaper.pdf
[6] https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY