Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 6, pp. 2397–2407
DOI : 10.47974/JDMSC-2310

Modification of SHA-256 hashing algorithm in protecting website service user passwords

Sarwindah [‡]
Department of Digital Business
Institut Sains dan Bisnis Atma Luhur
Pangkalpinang
Indonesia

Yohanes Setiawan Japriadi [†]
Department of Informatics Engineering
Institut Sains Dan Bisnis Atma Luhur
Pangkalpinang
Indonesia

Sujono [§]
Maxrizal *
Department of Information Systems
Institut Sains dan Bisnis Atma Luhur
Pangkalpinang
Indonesia

Abstract

The SHA-256 algorithm is still quite popular in password encryption for website user account logins and digital signature encryption. The security of SHA-256 lies in using a one-way hash function that cannot return the message to its original form. However, there are currently many tools on the internet that provide SHA-256 message decryption services so that the original message is obtained. This is what causes the security level of SHA-256 to be less good. For this reason, experts have introduced several improvements to SHA-256. One researcher has modified SHA-256 by adding the Hill Cipher algorithm. This aims to produce a different message digest so that it is not easily described by the decryption provider tool. However, hacking by tools on the internet is still possible using the Hill Cipher algorithm.

[‡] *E-mail:* **indah_syifa@atmaluhur.ac.id**
[†] *E-mail:* **ysetiawanj@atmaluhur.ac.id**
[§] *E-mail:* **sujono@atmaluhur.ac.id**
* *E-mail:* **maxrizal@atmaluhur.ac.id (Corresponding Author)**

© Taru Publications

Using the inverse (reverse) of the key matrix in the Hill Cipher allows the ciphertext to be converted back into plaintext, so there is always a chance of hacking by finding the inverse of the key matrix. For this reason, this study proposes improvements to SHA-256 by adding Hill Cipher modifications. Researchers still use SHA-256 to produce a message digest in the early stages. Furthermore, the resulting message digest will be forwarded with a modification of the Hill Cipher with a singular matrix key (it does not have an inverse). The use of a singular key matrix will invalidate the opportunity for hacking through an inverse matrix attack. In the final stage, researchers will test the execution speed of the proposed system.

Subject Classification: *94A60, 68P25, 15A15.*

Keywords: *SHA modification, SHA-256, Hill cipher 256, Hashing modification, Cipher modification.*

1. Introduction

There are many ways to protect the data, one of which is by using cryptography. The sample of cryptography algorithms are AES and SHA, which serve different purposes. Both algorithms are often used together because they are essential for securing systems; for example, AES can encrypt data, and SHA can hash passwords or verify data integrity. AES has versions based on key length: AES-128, AES-192, and AES-256. Meanwhile, SHA has several versions, starting from: SHA-0, SHA-1, SHA-2 (consisting of SHA-224, SHA-256, SHA-384, SHA-512), and SHA-3 (consisting of SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256). However, studies may enhance the security [1] of existing algorithms against attacks.

The SHA-256 algorithm is still quite popular in password encryption for website user account logins and digital signature encryption [2]. The algorithm can also be implemented as blockchain protocol security [3], [4], to improve scalability in large-scale IoT [4] and hospital management data security [5]. The security of SHA-256 lies in using a one-way hash function that cannot restore the message to its original form. In principle, SHA-256 converts the original message into a 256-bit message digest. To understand user variables is not only important for usability but also for security. For example, by understanding user behavior and characteristics, developers can design security systems that are more effective and user-friendly [6]. However, there are currently many tools on the internet that provide SHA-256 message decryption services so that unauthorized parties can obtain the original message. This is one of the factors that causes the SHA-256 security level to be less good.

For that, experts introduced improvements to SHA-256 by adding hardware acceleration [7], [8] or modifying SHA-256 with other algorithms [9], [10]. Furthermore, in research [11], researchers have modified SHA-256 by adding the Hill Cipher algorithm. This modification aims to produce a different message digest, so the decryption provider tools must describe it more easily. However, we still find the possibility of hacking opportunities by tools on the internet using the Hill Cipher algorithm. To encrypt a message, we need the matrix equation $C = KP \bmod m$, with ciphertext C, plaintext P, key matrix K and a modulo m. Meanwhile, we use the equation $P = K^{-1}C \bmod m$ [12], [13] to decrypt a message.

2. Method

Based on research [11], researchers use SHA-256 and the original Hill Cipher to keep the symmetric key in Hill Cipher confidential. This study combined the original SHA-256 and modified Hill Cipher using the singular matrix concept. The proposed model does not have to keep the Hill Cipher key secret and is mathematically secure against matrix inverse attacks because singular matrices do not have inverses.

The first stage in this study uses the SHA-256 algorithm. Furthermore, researchers modify the Hill Cipher algorithm with the singular matrix concept. In the final stage, researchers implement the results of the proposed algorithm modifications, compare the time, and test the security of the proposed algorithm.

Researchers have employed a comprehensive array of mathematical Hacking experiment techniques to test the proposed algorithm. The incorporation of Hill Cipher modifications is designed to enhance the randomness and irreversibility of the message digest generated from SHA-256. This rigorous testing process instils confidence in the proposed algorithm's robustness.

3. Results and Discussion

In this section, the results of the research on the proposed algorithm will be presented.

3.1 *Proposed Algorithm*

In previous research [9], researchers combined SHA-256 and Hill Cipher. The algorithm's weakness lies in the Hill Cipher key, namely, its security lies in the confidentiality of the key. When the hacker knows the

Hill key, the hacker can perform the inverse operation to find the correct password. In this study, we propose a modification that could have a significant impact on the security of the Hill Cipher key, by replacing it with a singular matrix.

Suppose a password is processed using SHA-256 to produce an output of 64 characters. Furthermore, the output is processed with a modified Hill Cipher. The production of 64 characters is converted using the table A = 0, B = 1,…, Y=24, Z=25, 1=26, 2=27,…, 8=34, 9 = 35. Furthermore, the 64 characters that have been converted are partitioned into 16 matrix blocks 2 × 2. The modified Hill Cipher will be generated with a key matrix 2 × 2 with a key matrix determinant test must be 0 (singular matrix). In this study, the pattern of the first row is a multiple of the second row, so the determinant is guaranteed to be zero [14], [15].

Consider again the 16 converted blocks, say $B_1, B_2, \ldots, B_{16}$. Let's take a sample of block B_1. We generate a singular matrix *K*as the key. Let's say our output is called C_1. Then we do $C_1 = K.B_1 \bmod 36$. Each block will be done with the same operation and the same key. Next, the results of each block are collected and converted back so that the final result is a more random character with a fixed length of 64 characters.

The novelty of this research lies in the use of a singular matrix. The concept of a singular matrix guarantees that there is no reverse operation that can decrypt the message. If we use a singular matrix *K*, then the inverse *K*is not found, so the plaintext $C_1, C_2, \ldots, C_{16}$ cannot be reversed.

3.2 *A Toy Example*

A new user registers with the username "maxrizal" in the system (shown in Figure 1). He plans to save the password "prize". This password is not displayed and is not stored in the database. The system he uses uses a password security algorithm, namely the SHA-256 algorithm that has been modified as in this study.

The first stage of the "prize" password is processed with SHA-256, resulting in "72a7a4f82e487ef3a4dd7287aa472ffacc126e4fb72ee4d307f913 589873e7b9" with a length of 64 bytes. This process is carried out like the commonly known SHA-256 algorithm. Next, we convert the results with the table A = 0, B = 1, C = 2, ….., Z = 25, 1 = 26, 2 = 27,….., 8 = 34, 9 = 35. Next, the results of this conversion are divided into 16 blocks consisting of a 2x2 matrix. In this example, we only discuss block 1. The treatment for the other blocks is the same as for block 1.

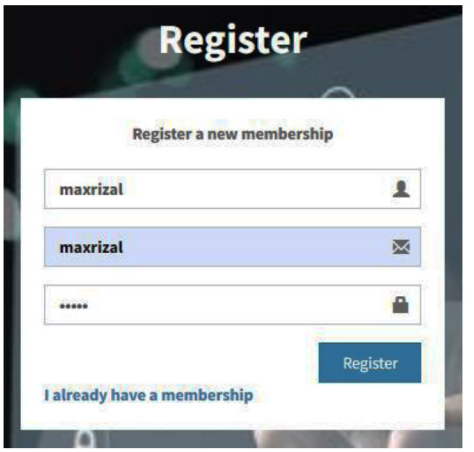

Figure 1

Registration procedure for new users

The next stage is to process the results with the Hill Cipher modified singular matrix. In this example, the system generates a matrix 2×2 with $K=\begin{bmatrix}19 & 1\\ 23 & 5\end{bmatrix}$. Note that it is singular because applies

$$\begin{aligned} Det[K] &= 19.5-1.23 \bmod 36 \\ &= 72 \bmod 36 \\ &= 0 \end{aligned}$$

.

Look again at block 1. The result of block one, initially "72a7", is converted to B1="33-28-0-33". We obtain the matrix $B_1=\begin{bmatrix}33 & 28\\ 0 & 33\end{bmatrix}$. The system do the matrix multiplication

$$\begin{aligned} C_1 &= K.B_1 \bmod 36 \\ &= \begin{bmatrix}19 & 1\\ 23 & 5\end{bmatrix}\begin{bmatrix}33 & 28\\ 0 & 33\end{bmatrix} \bmod 36 \\ &= \begin{bmatrix}627 & 895\\ 759 & 809\end{bmatrix} \bmod 36 \\ &= \begin{bmatrix}15 & 25\\ 3 & 17\end{bmatrix} \bmod 36 \end{aligned}$$

.

The same thing is done for $B_2, B_3, \ldots, B_{16}$. We get $C_1, C_2, C_3, \ldots, C_{16}$. We convert back $C_1 = 15-25-3-17$ to "pzdr". We convert the entire *C* to get

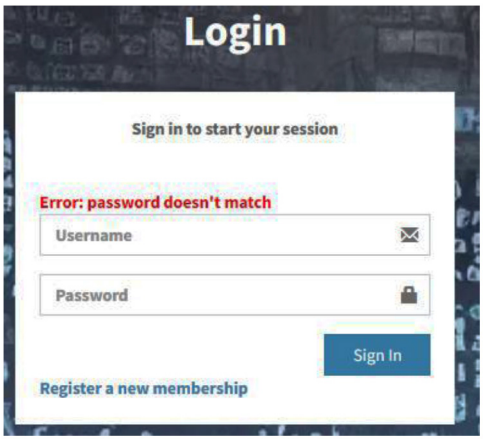

Figure 2

Login procedure failure for registered users

"pzdrf2z6wccku72vd7pvnz3r47gv7xvh34bg0jwjlttxhx9h5ol8ejujo181foz8". This 64-character password is stored in the database so that it is not easily hacked by hackers or system makers.

For example, if the user "maxrizal" wants to log back in, he only needs to enter the hidden password "prize". If the password is wrong, the system will reject the login (shown in Figure 2).

The results of this study found that the password "prize" was never stored in the database but rather "pzdrf2z6wccku72vd7pvnz3r47gv 7xvh34bg0jwjlttxhx9h5ol8 ejujo181foz8".

3.3 *Mathematical Attacks on Singular Matrices*

This section will analyze the possible mathematical attacks on the Hill Cipher modification algorithm with singular matrix modification. If K is non-singular (has an inverse), such as [11], then the equation $C = KP \bmod 36$ can be reversed to become plaintext $P = K^{-1}C \bmod 36$. Thus, hackers can find it in plaintext and hack through SHA-256 decryption. This is why the study [11] hides the key K in the Hill Cipher. In this study, K is singular, so it does not have an inverse K, so the equation $P = K^{-1}C \bmod 36$ does not apply. Mathematically, no inverse can reverse the plaintext. This means that the proposed algorithm is safe from mathematical attacks by hackers. The characteristic of a singular K matrix is a matrix that has a zero matrix determinant.

We take a plaintext chunk, the SHA-256 output chunk, namely $P = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$. We choose any nonsingular key matrix $K = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$. Based on [9], we can obtain the inverse matrix $K^{-1} = \begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix}$. It's important to note that the secrecy of P is paramount, as the system calculates $C = KP \bmod 36 = \begin{bmatrix} 4 & 11 \\ 11 & 29 \end{bmatrix}$. Any mistake or correct guess for the key *K* by a hacker can compromise the system's security. The hacker can calculate the inverse *K*, revealing $P = K^{-1}P \bmod 36 = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$.

In this study, we choose the singular matrix $K = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$. Note that $\det(K) = 1.4 - 2.2 = 0,$ so the inverse *K* cannot be found. Because K^{-1} is not found, the hacker cannot find the plaintext $P = K^{-1}P \bmod 36.$ This means that the Hill Cipher key in this study does not need to be kept secret, while in the study [11], it must be hidden.

1.4 *Encryption Speed and Algorithm Capacity Measurement*

In this study, we will measure the encryption speed of the proposed model and the space capacity required by the proposed system (shown in Table 1).

Table 1

Table 1: Speed Table Based on Encryption Time

No	Password sample	SHA-256 output	Attempt to						
			Bits	1	2	3	4	5	Average encryption time (ms)
1	a	ca978112ca1 bbdcafac23 1b39a23dc4d a786eff8147 c4e72 b98077 85afee48bb	16	0.175	0.231	0.499	0,527	0,460	0.302
2	c4	0012a3fa000 c5dc26ee658 c3c58e12cecd 58d6455cec3 d5621f0c78 7675b38aa	32	0.213	0.239	0.326	0.154	0,236	0.239

Contd...

3	v2#	1b73f3b13c b0d5a924a0 9fc352a91a6 f2f6f79cca3 587401dabed 1c1f9222a3d	64	0.390	0.303	0.211	0.212	0.216	0.266
4	c%k7	3ca5563f74b 24787d2027 af35a85de99 4c8419028046 d5c5 30472f6c 7bc9c375	128	0.257	0.180	0.257	0.302	0.288	0.257
5	xT4CA	d92e148825 a7c98800fb 8833 035e0 f82ada22d0 0bc1618de0 d730b7873d4 40ec	256	0.226	0.229	0.335	0.319	0.433	0.308

Encryption time can vary because each trial's K matrix generated in Hill Cipher constantly changes. The larger the number in the generated matrix, the greater the matrix multiplication time used for randomization. Next, we also test the space capacity of the proposed algorithm (shown in Table 2).

Table 2

Space Capacity of The Proposed Algorithm

			Attempt to						
No	**Password sample**	**SHA-256 output**	**Bits**	**1**	**2**	**3**	**4**	**5**	**Average space capacity (kbyte)**
1	a	ca978112ca 1bbdcafac2 31b39a23dc 4da786eff8 147c4e72b9 807785afee 48bb	16	27,078	26,784	28,628	27,312	27,080	27,376

Contd...

2	c4	0012a3fa000c5dc26ee658c3c58e12cecd58d6455cec3d5621f0c787675b38aa	32	27,516	27,344	27,220	27,952	27,152	27,437
3	v2#	1b73f3b13cb0d5a924a09fc352a91a6f2f6f79cca3587401dabed1c1f9222a3d	64	27,288	27,092	27,144	27,928	27,232	27,337
4	c%k7	3ca5563f74b24787d2027af35a85de994c8419028046d5c530472f6c7bc9c375	128	27,844	27,400	27,171	27,852	27,796	27,613
5	xT4CA	d92e148825a7c98800fb8833035e0f82ada22d00bc1618de0d730b7873d440ec	256	27,564	28,448	29,276	28,252	28,748	28,458

Note that the proposed algorithm requires a relatively fast and efficient 1-time encryption time, and the space capacity needed is also manageable.

4. Conclusion

This study proposes improvements to SHA-256 by adding Hill Cipher modifications. Researchers still use SHA-256 to generate message digests in the early stages. Furthermore, the generated message digest will be forwarded with Hill Cipher modifications with a singular matrix key (without an inverse). Using a singular key matrix will cancel the chance of hacking through an inverse matrix attack. The measurement results show that the proposed algorithm requires a relatively fast and efficient 1-time encryption time, and the space capacity needed is also manageable.

Acknowledgment

The Research Team would like to thank the Ministry of Education, Culture, Research, and Technology for funding the Fundamental Research Grant scheme with contract number 104/E5/PG.02.00/PL/2024. The Research Team would also like to thank the Atma Luhur Institute of Science and Business for supporting facilities and computer laboratories.

References

[1] S. Shahapure, V. Sule, and R. D. Daruwala, "Variation and security enhancement of block ciphers by embedding," *J. Discret. Math. Sci. Cryptogr.*, vol. 22, no. 2, pp. 151–160 (2019), doi: 10.1080/09720529.2019.1576336.

[2] D. Rachmawati, J. T. Tarigan, and A. B. C. Ginting, "A comparative study of Message Digest 5(MD5) and SHA256 algorithm," *J. Phys. Conf. Ser.*, vol. 978, no. 1 (2018), doi: 10.1088/1742-6596/978/1/012116.

[3] R. Martino and A. Cilardo, "Designing a SHA-256 processor for blockchain-based IoT applications," *Internet of Things (Netherlands)*, vol. 11, p. 100254 (2020), doi: 10.1016/j.iot.2020.100254.

[4] R. Fotohi and F. Shams Aliee, "Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT," *Comput. Networks*, vol. 197, no. May, p. 108331 (2021), doi: 10.1016/j.comnet.2021.108331.

[5] M. D. Mohanty *et al.*, "Design of Smart and Secured Healthcare Service Using Deep Learning with Modified SHA-256 Algorithm," *Healthc.*, vol. 10, no. 7, pp. 1–16 (2022), doi: 10.3390/healthcare10071275.

[6] K. Sagar and A. Saha, "The effect of user variables on academic websites usability : An empirical study," *J. Stat. Manag. Syst.*, vol. 22, no. 2, pp. 161–186 (2019), doi: 10.1080/09720510.2019.1580898.

[7] Y. Chen and S. Li, "A high-throughput hardware implementation of SHA-256 algorithm," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2020-Octob, pp. 17–20 (2020), doi: 10.1109/iscas45731.2020.9181065.

[8] M. Kammoun, M. Elleuchi, M. Abid, and M. S. Bensaleh, "FPGA-based implementation of the SHA-256 hash algorithm," *DTS 2020 - IEEE Int. Conf. Des. Test Integr. Micro Nano-Systems* (2020), doi: 10.1109/DTS48731.2020.9196134.

[9] S. Zhu, C. Zhu, and W. Wang, "A new image encryption algorithm based on chaos and secure hash SHA-256," *Entropy*, vol. 20, no. 9 (2018), doi: 10.3390/e20090716.

[10] Aqeel-ur-Rehman, X. Liao, A. Kulsoom, and S. Ullah, "A modified (Dual) fusion technique for image encryption using SHA-256 hash and multiple chaotic maps," *Multimed. Tools Appl.*, vol. 75, no. 18, pp. 11241–11266 (2016), doi: 10.1007/s11042-015-2851-7.

[11] Z. Panjaitan, E. F. Ginting, and Y. Yusnidah, "Modifikasi SHA-256 dengan Algoritma Hill Cipher untuk Pengamanan Fungsi Hash dari Upaya Decode Hash," *J. SAINTIKOM (Jurnal Sains Manaj. Inform. dan Komputer)*, vol. 19, no. 1, p. 53 (2020), doi: 10.53513/jis.v19i1.225.

[12] P. Sundarayya and G. Vara Prasad, "A public key cryptosystem using Affine Hill Cipher under modulation of prime number," *J. Inf. Optim. Sci.*, vol. 40, no. 4, pp. 919–930 (2019), doi: 10.1080/02522667.2018.1470751.

[13] Maxrizal, "Public Key Cryptosystem Based on Singular Matrix," *Trends Sci.*, vol. 19, no. 3, p. 2147 (2022), doi: 10.48048/tis.2022.2147.

[14] H. Anton and C. Rorres, *Elementary Linear Algebra: Applications Version*, 9th ed. John Wiley & Sons, Inc. (2005).

[15] D. S. Dummit and R. M. Foote, *Abstract Algebra*, 3rd ed. John Wiley & Sons, Inc. (2004).

Received September, 2024