Improve MD5 Hash Function For Document Authentication

Tameem Hameed Obaida[1] , Hanan Abbas Salman[2] , Hasan Najim Zugair[3]

[1]Department of Computer Systems Techniques, Al-Furat Al-Awsat Technical University, Najaf Technical Institute, AL Najaf, Iraq.

[2]Department of Computer Systems Techniques, Al-Furat Al-Awsat Technical University, Najaf Technical Institute, AL Najaf, Iraq.

[3]Department of Computer Systems Techniques, Al-Furat Al-Awsat Technical University, Najaf Technical Institute, AL Najaf, Iraq.

Abstract

With ever-increasing network connectivity, message integrity and authenticity are critical. The primary building component of message integrity is cryptographic hash the functions. Hash functions are utilized and developed in a variety of ways. The purpose of this study is to a propose and discuss the new keyed hash function. The Hash function and the Henon map are used in this suggested technique. For whatever length of input, this technique generates a 128-bit hash code. The function hashes a message with a key so that an intruder who doesn't know the key can't fabricate the hash code, and so it meets a security, authentication, and integrity requirements for the communication in a network. The paper explains the function design an algorithm, as well as its security and implementation details. The simulation findings suggest that a text authentication and forensics algorithm with strong tampering localization ability may be utilized to authenticate and forensics text authenticity and integrity.

Keywords: Hash function, MD5 algorithm, Henon map.

1. Introduction

Network Security means protecting User confidentiality, integrity, and resource availability [1]. Network Security initializes with an authorization i.e., with the help of credentials such as a username and a password to access a specific device commonly. Network security consists of policies each network administrator has adopted to prevent and track illegitimate access privileges, alteration and denial of a computer network and network resources. When a user is approved to do something else, a firewall will require them to follow rules such as what resources the network user is allowed to access [2]. Thus, these policies are reasonable to prevent unauthorized access to the device, but this component may fail to track potentially dangerous content such as software warms or network transmission of a Trojans [3].

A cryptographic hash function is a function that accepts an arbitrary-length input and a results a compact, fixed-length output [3, 4]. It's a message integrity mechanism that also provides source authentication if keys are utilized. Integrity refers to the process of sending a message to the intended recipient without any modifications or changes. Source Authentication protects messages from deliberate deception and impersonation [5].

The use of a keyed hash function might help to overcome this problem. The intruder or deceiver cannot create a new message or change an existing one without knowing the genuine key, which is only known by the sender and recipient and is used for authentication or hash generation. [6]. As a result, only the sender and recipient may communicate in this manner. Message authentication code (MAC) is the use of a key for hash creation [7].

Hash algorithm can be mainly classified into MDx and SHA, where MD5 and SHA1 are the most widely used and MD5 algorithm can produce a 128-bit abstract and SHA algorithm can produce a 160-bit abstract [8]. The MD5 algorithm was developed by Ronald L.Rivest of the MIT Laboratory for Computer Science and RSA Data Security in the 1990s, and it evolved from MD2, MD3, and MD4 [9]. A MD5-Hash method separates the input data into 512-bit groups, then divides each group into 16 32-bit groups, and finally concatenates the four groups into a 128-bit hash value, which is the output result 8.

Hash algorithm, as an encryption algorithm in the field of information security, plays an important role in modern cryptology. Its main properties [10] are:1) it can be applied to any given message and it’s easy to get its Hash value; 2) it can compress arbitrary-length message or data into fixed-length message abstracts. The storage space of an output value is often considerably less than that of an input value; 3) the inverse hash is unlikely to be solved, therefore we can't recover the original input value from a given one; 4) Even minor changes in the input value will result in significant changes in the output value; 5) The Hash function has two sorts of collision resistance: for a given message A1, finding another message A2 to set up the equation H(A1)=H(A2) is difficult; for separate messages B1 and B2, setting up the equation H(B1)=H(B2) is impossible. First and foremost, every cryptographic hash function should be able to survive all potential assaults. The input length is flexible in cryptographic hash functions, while the output length is fixed [10]. As a result of the infinite-to-finite length mapping, collisions in hash functions are always present. As a result, the requirement definition may be changed from "it is impossible to identify two separate messages that create the same hash value" to "it should be extremely difficult to locate two different messages that produce the same hash value." The fundamental design algorithm should impose this challenge [11].

To begin, Yuval [12] suggested the Birthday Paradox as a means for identifying the collisions in hash functions, which led to a birthday attack. By this technique, a collision is discovered with probability q2 / 2n after q queries to a hash function whose output is of n-bit length [13]. Aside from that, simple operations like addition, XOR, complements, and so on must be included in the algorithm to make it run quickly. In addition, the design algorithm's security must be shown. Only relying on security assumptions might result in disaster. It should also have a customizable structure, allowing it to be changed and made safe against future threats. A hash function is a mathematical relation or procedure that turns a random-length input message into a compressed message of a set length. Message digest is the name for the hash function's fixed length compressed message output. The resulting message digest is regarded as the message's unique digital mark [14]. Figure 1 shows how the MD5 algorithm works. The steps that follow explain how the algorithm works [15]. Recent attacks on MD4 [16] and MD5 [3, 17] have prompted additional study towards the development of new cryptographic hash functions as well as cryptanalysis of current ones [18]. This article outlines the creation of a novel hash algorithm that includes a key and meets both a message integrity and source authentication criteria.

The goal of this study is to improve the MD5 algorithm by addressing its flaws. Propose an improved MD5 algorithm that may be utilized to improve data security and integrity, and hence the system's security, dependability, integrity, and performance. To protect user privacy and data by hashing the information in a way that it cannot be readily stolen or hijacked.

2. Background of MD5 Function

A variable-length message is converted into a 128-bit fixed-length output using MD5 [19]. Figure 1 shows how the input message is broken down into 512-bit blocks (sixteen 32-bit words) and padded to be divisible by 512. [20] The following is how padding works: The message is appended with a single bit, 1 at the end, at first. The message is then followed by as many zeros as are required to reduce the length to 64 bits fewer than 512. The leftover bits are filled with 64 bits modulo 264, which is the original message's length.

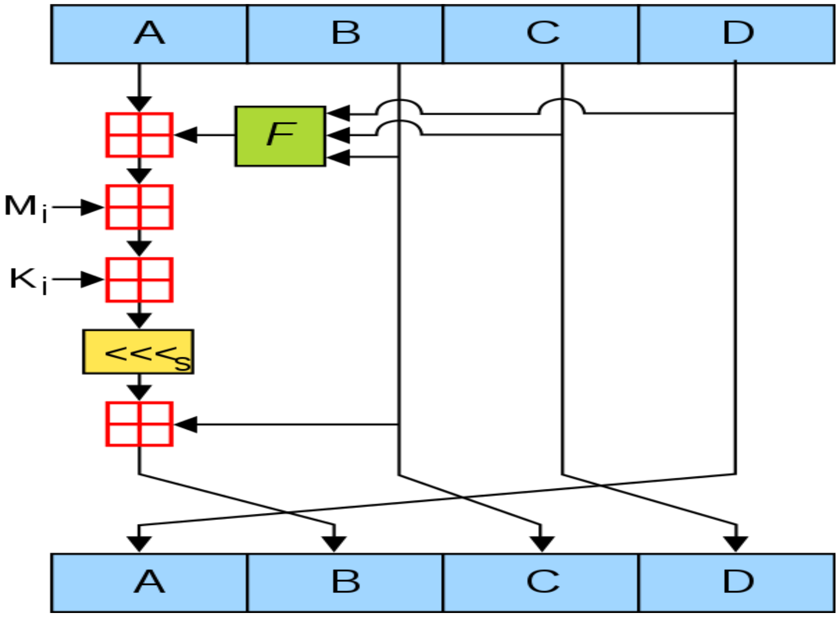

Figure 1: The MD5 structure

The fundamental MD5 algorithm uses a 128-bit state divided into four 32-bit words: A, B, C, and D. These are preset to a series of constants [21]. The state is subsequently modified by the main algorithm, which employs each 512-bit message block in turn. Each cycle consists of 16 operations based on a non-linear function F, modular addition, and left rotation. One action inside a round is depicted in Equation (1). There are four potential functions, each of which is employed in a different round [22]:

$$\begin{gathered} \mathrm{F}(\mathrm{B},\mathrm{C},\mathrm{D}) = (\mathrm{B} \wedge \mathrm{C}) \vee (\neg\mathrm{B} \wedge \mathrm{D}) \\ \mathrm{G}(\mathrm{B},\mathrm{C},\mathrm{D}) = (\mathrm{B} \wedge \mathrm{D}) \vee (\mathrm{C} \wedge \neg\mathrm{D}) \\ \mathrm{H}(\mathrm{B},\mathrm{C},\mathrm{D}) = \mathrm{B}\oplus\mathrm{C}\oplus\mathrm{D} \\ \mathrm{I}(\mathrm{B},\mathrm{C},\mathrm{D}) = \mathrm{C}\oplus(\mathrm{B} \vee \neg\mathrm{D}) \end{gathered} \tag{1}$$

Where ⊕, ∧, ∨, and ¬ denote the respectively as XOR, AND, OR, and NOT. The MD5 function used to encrypted text from original to encrypted text is done by using the following five steps:

Appending Step: Its responsible about forcing a message length be equivalent to 448 modulo 512, resulting in a message length that is a multiple of 512, by combining a lot of bits (1...512). As a consequence, the message size is decreased to only 64 bits. The "1' bit is appended to the message, followed by a series of 0s, resulting in a block length of 448 characters.

Add a length message step: Its phase included an addition to an original message's length, which indicates an original message's 64-bit length at a conclusion of the bits from the previous step. If a message is k bits long, The k mod 264 value is appended to the equation. A preceding two phases output in a message with a length of L 512. Assume this letter is made up of 32-bit words. The letter has a value ranging from 0 to N-1, with N D L equaling 16.

Initialize MD5 buffer step: To find a message digest, recorders are set up as a 4 buffer (D, C, B, and A). Use a following basic settings to give each of these recorders a 32-bit length in a hexa byte and lower arrange: (A=67452301, B=efcdab89, C=98badcfe, D=10325476) A=67452301, B=efcdab89, C=98badcfe, D=10325476) (A=67452301, B=efcdab89, C=98badcfe, D=10325476) A=67452301, B=efcdab89, C=98badcfe, D=10325476) A=67452301, B=efcdab89, C=98badcfe, D=10325476) A=67452301, B= an algorithm's ultimate result, 128 bits D 4 32, This information is saved on these recorders.

Process the message block step: To get a desired result, the material is treated in several processes. The most crucial phase in the algorithm is this one. This stage processes 512-bit 16-word message blocks. It usually has four cycles, each with a different function: F, G, H, and I, each with 16 steps. Each round has 16 steps on the recorders, with the words from each step being sent into four rounds in each round.

Output step: Once all blocks have been processed, the plain data are converted to ciphertext or hash format as a message digest, with the final MD5 hash value being 128 bits. The [MD5] approach is depicted in full in Figure 1, which may be found in [23].

3. Feebleness in the Existing MD5 Algorithm

The MD5 algorithm has the number of flaws, including vulnerabilities to rainbow table, dictionary, birthday, and other attacks [24]. The MD5 algorithm's flaws have been studied extensively. The focus of a literature study is on many attacks that may be carried out on a MD5 algorithm [25]. Many papers focus on many types of attacks that may be used to break or hijack MD5.

3.1 Birthday attack

A birthday attack is a sort of attack that depends on the operation's unpredictability.. As with the birthday paradox, the attack is based on a somewhat random procedure and chance. It jeopardizes the security of data and the integrity of messages. When there are a lot of people in a room, the chances of two persons having birthdays on the same day are very high. a birthday paradox is employed to attack the cryptographic functions' flaw [18]. It differs from the brute-force situation, in which all conceivable data is examined until the hashed result is equal. Compared to brute force attacks, birthday attacks are more efficient and faster. In the birthday assault, the data length is quite important. If a hashed output is relatively short in length, the birthday an attack is more likely to succeed [26].

3.2 A dictionary attack

The dictionary attack is a type of an attack in which an original message is deciphered using a list of dictionaries' data, which is commonly used as a password. It differs from a brute force attack, which employs all available methods to compromise the cryptographic function. In contrast to brute force, it employs the guess to hijack the process through the usage of a list of options. The assault is more likely to be hijacked than brute force since it relies on a guess. The hijacking speed of a dictionary assault, on the other hand, is much faster than a brute force attack. A hashed data of the dictionary files is pre-computed in a dictionary attack. The data to be cracked or hijacked is compared to the hashed data created [27].

3.3 Rainbow table

Dictionary assault is the more advanced variation of a dictionary attack. In general, the tiny password combined with a small domain makes data hijacking very simple using a dictionary attack [28]. The rainbow-themed table. The rainbow table attack is a method of discovering an original message by computing the hashed result for a large number of inputs using the same operations in a table. A rainbow table is much quicker because a result is already saved for speedier the algorithm a hijacking. For speedier and more efficient hijacking of cryptographic algorithms like MD5, the hashed outcomes for plain text is previous calculated and saved in a database. The rainbow table is a previous calculated table that contains hashed values that include a hashed output of the original message. In addition, a rainbow table refer to the sophisticated variant of a dictionary attack in which the hashed result are data with a higher likelihood of being hijacked. Because a hashed output for the same original data is the same in all circumstances, a time it takes to find the required hashed result is greatly reduced. As a result, One of the most serious problems of the MD5 method is that the hashed output for the same input is always the same.

4. Enhance MD5 hash function

Many academic articles and publications have been published in an attempt to find a solution to the present MD5 algorithm's flaws. The literature study outlines all of the benefits and drawbacks of each phase in the current MD5 algorithm. As a result, a flaws have been eliminated, and the strengths or benefits have been examined. For the same input value or data, the hashed output is always the same, which is one of MD5's primary flaws. The use of a key value that distinguishes the output for the same input helps counteract this strategy. Furthermore, a brute force attack demonstrates that the technique may be readily exploited if the hashed result is short. As a result, this flaw may be avoided by simply lengthening the MD5 algorithm. As a result, the use of a key and variable output length provides a solution to the MD5 algorithm's flaws. As a result, a research suggests and details the creation of a hybrid model that includes both the key value and variable length as a remedy to the present MD5 algorithm's flaws. The Henon map was used to create an arbitrary matrix using a new key generated using the data encryption standard's initial permutation (IP) tables (DES) [29]. These technologies assisted in achieving a balance between time and complexity. The enhanced MD5 hash algorithm is shown in Figure 2.

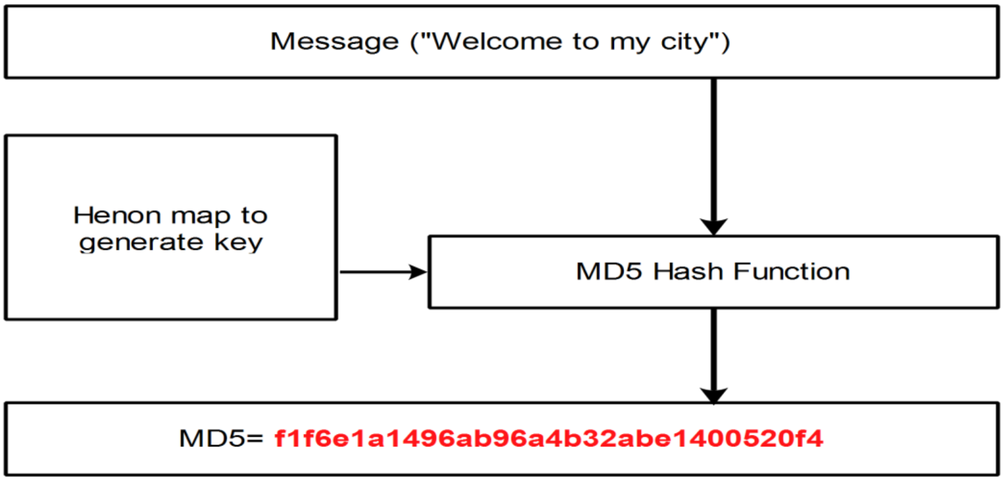
Message ("Welcome to my city")
Henon map to generate key
MD5 Hash Function
MD5= f1f6e1a1496ab96a4b32abe1400520f4

Figure 2: Proposed model

4.1. Improved MD5

Because the conventional MD5 method has several flaws and inadequacies, By employing a hybrid technique with changeable outcome length and key value, these can be decreased. A modified MD5 method improves on an original MD5 algorithm by altering a length of the key. As a result, the improved method employs a Henon map with changeable output length and key. Because each user's key is unique, the output differs depending on the key, yet the same key produces the same result. The introduction of a key in the MD5 algorithm totally eliminates the possibility of a dictionary attack, rainbow table, birthday or because an output will change depending on the key. A variable length will improve the hashing algorithm since its length makes it subject to various assaults. An improved version of the existing MD5 algorithm is depicted in the diagram. Michel Henon designed the Henon map, which is a 2D chaotic map. It is a dynamical system with discrete time. As illustrated in an equation (2), the henon map starts with a point (xn, yn).

$$\begin{gathered} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{gathered} \tag{2}$$

Where equal 1 and b equals 0.3. When utilizing electronic documents across several networks, the problem of data privacy and security is crucial, particularly when dealing with unwanted infiltration efforts to read data in the documents and modify their content. If not controlled, these actions can lead to the development of the false copy or stolen material, as well as acts of espionage. Through the following procedures, MD5 algorithm, which is used in the proposed system, has been enhanced:

Step 1: An explicit text was translated to a binary in this manner (0,1).

Step 2: The text size was established by calculating the number of bits, which was then fed into the suggested secret key production procedure.

Step 3: An initial step in an improvement is to increase the system's complexities, which is accomplished using a shift register with linear feed.

Step 4: An equation (2) of degree 32 is used to solve a problem, a new important advantage may be produced. First two values selected from equation (2). Second, given equation, make two binary vectors (2) where it is indicated to (value 1, value 2) When: value 1[1 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 000 0 0 0 0 0 0 1] and value 2 [1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1]. The two-sequence vector is known as (value 1 and value 2), It will be employed later in the proposed system and will reveal the overall structure of a key generation.

Step 5: By using the linear-feedback shift register on values 1 and 2, a new value with a 64-bit length for values 1 and 2 is produced.

Step 6: Using the XOR between the first and second keys to get the first key (value 1, value 2).

Step 7: The major key is generated by applying the LFSR 64 roles on the primary key obtained in the preceding step in the DES initial permutation tables to yield 64-bits. The load is computed by multiplying the extra 64-bit outcomes by the predecessor 64-bit outputs.

Step 8: The message file is separated into multiple blocks, each with 512 bits, to improve efficiency.

Step 9: To give a high-quality hash value, the output of four MD5 algorithm registers (A, B, C, and D) is retrieved as input to the Fourth-order technique, which is utilized to solve the differential equation in the delay function.

Results

The results of simulations employing the Dictionary Attack, Avalanche Effect and Random Test are shown below.

The avalanche Effect

The avalanche effect was assessed to validate the updated MD5 hash result by analyzing the severe binary bit change for a little change in the input and must be able to prove a significant degree of change in order to achieve an optimum value of fifty percent (50 percent). This is a critical characteristic of any cryptographic hash algorithm since it determines whether the hash result can forecast the plaintext or not. The plaintext, associated message digest hash value, binary value of hash, and percent of bit altered when the original and updated MD5 hash methods were used are shown in Tables 1 and Table 2.

Table 1: An original MD5 hash function

Input string	MD5	A hash binary	Bit changed %

Welcome to	1b803bfbc0b3c00a9b7ccbfb5e82ae29	11111001111010001001100100011…	
welcome to	1b803bfbc0b3c00a9b7ccbfb5e82aa21	11011100000000001111011111110…	

The outcome of applying the new MD5 hash algorithm is shown in Table 2. The avalanche result in percentage of bits altered illustrates that deploying the new method has a significant impact.

Table 2: MD5 modified

Input string	MD5	Hash binary	Bit changed %
Welcome to	f9e899180e19eebe5871a8fe903790e9		
welcome to	1b803bfbc0b3c00a9b7ccbfb5e82ae29		

Table 3 show the results with different size of file and time running.

Table 3: Files of various sizes and lengths of time are executed.

File size (k)	MD5	Time execution
150	dd1b4db6273e67ea5a60fa228357a335	0.02
300	aa7cbaaa1511b5ebe97929d39ab34ae3	0.032
500	678a071d0fde6fbb56ab661fafdcbf0d	0.043
700	159e859b31a421f6683db890efefbfe6	0.058
1000	07ec57a6cb9e2b47bd347d50ba56ba32	0.087
1500	ca450a1cffe23f66ffffeffc7223a2ca	0.12

Dictionary Attack

One of the most common cryptographic tactics used to extract plaintext messages is dictionary attacks, which attempt to decode messages using commonly used phrases, passwords, and short keys stored in the attacker's database. The researchers also investigated putting a hash value generated by a modified MD5 to the test and seeing if it could be decrypted using a dictionary attack and free tools for cracking hash values. Table 4 shows that with the updated padding scheme and extra security procedures, the original plain text message could not be recovered, and these cracking tools failed.

Table 4: Dictionary attack for check the proposed model

Input string	MD5 value	Result
crackstation.net	5c5e9c59db319b4f330273dbc288f257	Not found
hashkiller.co.uk	ca57fa6c46d5c69c674977ea07e1f7e7	Not found

Conclusion

The experts have performed an outstanding research study in order to improve MD5's collision sensitivity The addition of 1024 bits to the message input block, accompanied by the installation of a novel padding method, not just to allows you a variety of ways to create a message block, However, a message length increase to 64-bit sub blocks is also possible. An inclusion of processes that result in the updated MD5 hash function's security being strengthened added complexity finally, the message digestibility computation. The criteria employing the increased padding scheme technique and the additional security operations yield a percentage output score of 57.03, which exceeds the optimum value of fifty (50) percent rate, based on the avalanche effect test, which assesses whether prediction of the original plaintext is possible.

Finally, a randomization test was carried out to make certain that no duplicate dissolve values were discovered. The mean value test was calculated using the arithmetic mean, and using the criterion of adopting the updated padding scheme and extra security methods, it was able to achieve a required central value of at least fifty (50) percent, resulting in scores of 50.45 and 51.93, respectively. To determine if the extra security was secure from dictionary assaults, internet cracking programs (hashkiller.co and crackstation.net) were used. The results of a test indicate that retrieving the plaintext message that was sent is impossible. In a future, further cryptographic approaches might be added to give a message digest value that is more secure and collision-free.

Reference

[1] J. L. Naidu, A. C. Gorakala, and S. S. Amiripalli, "Hash functions and its security for Snags," 2020.

[2] J. Deepakumara, H. M. Heys, and R. Venkatesan, "FPGA implementation of MD5 hash algorithm," in Canadian Conference on Electrical and Computer Engineering 2001. Conference Proceedings (Cat. No. 01TH8555), 2001, vol. 2, pp. 919-924: IEEE.

[3] X. Wang and H. Yu, "How to break MD5 and other hash functions," in Annual international conference on the theory and applications of cryptographic techniques, 2005, pp. 19-35: Springer.

[4] P. Metzger and W. Simpson, "IP authentication using keyed MD5," RFC 1828, August1995.

[5] J. Liang and X.-J. Lai, "Improved collision attack on hash function MD5," Journal of Computer Science and Technology, vol. 22, no. 1, pp. 79-87, 2007.

[6] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "BLAKE2: simpler, smaller, fast as MD5," in International Conference on Applied Cryptography and Network Security, 2013, pp. 119-135: Springer.

[7] Z. Yong-Xia and Z. Ge, "MD5 research," in 2010 second international conference on multimedia and information technology, 2010, vol. 2, pp. 271-273: IEEE.

[8] A. Sotirov et al., "MD5 considered harmful today, creating a rogue CA certificate," in 25th Annual Chaos Communication Congress, 2008, no. CONF.

[9] M.-J. Wang and Y.-Z. Li, "Hash function with variable output length," in 2015 International Conference on Network and Information Systems for Computers, 2015, pp. 190-193: IEEE.

[10] Z.-Y. Wang, H.-G. Zhang, Z.-P. Qin, and Q.-S. Meng, "A Fast Attack Algorithm on the MD5 Hash Function," Journal of Shanghai Jiaotong University (Science), vol. 11, no. 2, pp. 140-145, 2006.

[11] D. Kaminsky, "MD5 to be considered harmful someday," in Aggressive Network Self-Defense: Elsevier, 2005, pp. 323-337.

[12] M. Stevens, "On collisions for MD5," ed: Citeseer, 2007.

[13] M. Stevens, A. K. Lenstra, and B. De Weger, "Chosen-prefix collisions for MD5 and applications," International Journal of Applied Cryptography, vol. 2, no. 4, pp. 322-359, 2012.

[14] I. A. Landge and H. Satopay, "Secured IoT through hashing using MD5," in 2018 fourth international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB), 2018, pp. 1-5: IEEE.

[15] I. A. Landge and B. Mishra, "Hardware based MD5 implementation using VHDL for secured embedded and VLSI based designs," in 2016 International Conference on Communication and Electronics Systems (ICCES), 2016, pp. 1-6: IEEE.

[16] X. Wang, X. Lai, D. Feng, H. Chen, and X. Yu, "Cryptanalysis of the Hash Functions MD4 and RIPEMD," in Annual international conference on the theory and applications of cryptographic techniques, 2005, pp. 1-18: Springer.

[17] J.-P. Aumasson, W. Meier, and F. Mendel, "Preimage attacks on 3-pass HAVAL and step-reduced MD5," in International Workshop on Selected Areas in Cryptography, 2008, pp. 120-135: Springer.

[18] C.-W. Ng, T.-S. Ng, and K.-W. Yip, "A unified architecture of MD5 and RIPEMD-160 hash algorithms," in 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512), 2004, vol. 2, pp. II-889: IEEE.

[19] M. C. Ah Kioon, Z. S. Wang, and S. Deb Das, "Security analysis of MD5 algorithm in password storage," in Applied Mechanics and Materials, 2013, vol. 347, pp. 2706-2711: Trans Tech Publ.

[20] P.-A. Fouque, G. Leurent, and P. Q. Nguyen, "Full key-recovery attacks on HMAC/NMAC-MD4 and NMAC-MD5," in Annual International Cryptology Conference, 2007, pp. 13-30: Springer.

[21] A. K. Kasgar, M. K. Dhariwal, N. Tantubay, and H. Malviya, "A review paper of message digest 5 (MD5)," International Journal of Modern Engineering & Management Research, vol. 1, no. 4, pp. 29-35, 2013.

[22] B. P. Gajendra, V. K. Singh, and M. Sujeet, "Achieving cloud security using third party auditor, MD5 and identity-based encryption," in 2016 International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 1304-1309: IEEE.

[23] K. Jarvinen, M. Tommiska, and J. Skytta, "Hardware implementation analysis of the MD5 hash algorithm," in Proceedings of the 38th annual Hawaii international conference on system sciences, 2005, pp. 298a-298a: IEEE.

[24] T. Xie, F. Liu, and D. Feng, "Fast Collision Attack on MD5," IACR Cryptol. ePrint Arch., vol. 2013, p. 170, 2013.

[25] E. Thompson, "MD5 collisions and the impact on computer forensics," Digital investigation, vol. 2, no. 1, pp. 36-40, 2005.

[26] G. Gupta, "What is Birthday attack??," ed: February, 2015.

[27] J. Majumder, "Dictionary Attack on MD5 hash," International Journal of Engineering Research and Applications, vol. 2, no. 3, pp. 721-724, 2012.

[28] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in Proceedings of the 9th ACM Conference on Computer and Communications Security, 2002, pp. 161-170.

[29] T. H. Obaida and D. H. Abd, "A Robust Approach for Mixed Technique of Data Encryption Between DES and RC4 Algorithm," Journal of Kufa for Mathematics and Computer Vol, vol. 3, no. 2, pp. 48-54, 2016.

[30] E. N. Lorenz, "Compound windows of the Hénon-map," Physica D: Nonlinear Phenomena, vol. 237, no. 13, pp. 1689-1704, 2008.

[31] Ali MH, Jaber MM, Abd SK, Rehman A, Awan MJ, Vitkutė-Adžgauskienė D, Damaševičius R, Bahaj SA. Harris Hawks Sparse Auto-Encoder Networks for Automatic Speech Recognition System. Applied Sciences. 2022; 12(3):1091. https://doi.org/10.3390/app12031091

[32] Jaber, M. M., Abd, S. K., Shakeel, P. M., Burhanuddin, M. A., Mohammed, M. A., & Yussof, S. (2020). A telemedicine tool framework for lung sounds classification using ensemble classifier algorithms. Measurement: Journal of the International Measurement Confederation, 162, 107883. https://doi.org/10.1016/j.measurement.2020.107883