IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 15, 2020, accepted March 30, 2020, date of publication April 20, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989050

A Novel Improvement With an Effective Expansion to Enhance the MD5 Hash Function for Verification of a Secure E-Document

AMMAR MOHAMMED ALI AND ALAA KADHIM FARHAN, (Member, IEEE)
Computer Sciences Department, University of Technology, Baghdad 10066, Iraq
Corresponding author: Alaa kadhim Farhan (110030@uotechnology.edu.iq)

ABSTRACT MD5 is a one-way cryptographic function used in various fields for maintaining data integrity. The application of a Hash function can provide much protection and privacy and subsequently reduce data usage. Most users are familiar with validating electronic documents based on a Hash function, such as the MD5 algorithm and other hash functions, to demonstrate the data integrity. There are many weaknesses of the current MD5 algorithm, mainly its failures and weaknesses against varying types of attacks, such as brute force attacks, rainbow table attacks, and Christmas attacks. Therefore, the method proposed in this paper enhances the MD5 algorithm by adding a dynamic variable length and a high efficiency that simulates the highest security available. Whereas the logistic system was used to encode ribonucleic acid (RNA) by generating a random matrix based on a new key that was created using the initial permutation (IP) tables used in the data encryption stander (DES) with the linear-feedback shift register (LFSR), this work proposes several structures to improve the MD5 hash function. The experimental results demonstrate its high resistance to hackers while maintaining a suitable duration. This paper discusses the design of a confident hash algorithm. This algorithm has characteristics that enable it to succeed in the field of digital authentication and data integrity.

INDEX TERMS MD5 hash function, data integrity, Ribonucleic acid (RNA), LFSR, IP, chaotic system.

I. INTRODUCTION

The large interest in information security has centered on the balanced protection of the confidentiality, integrity and availability of data, with the amount of data required to be analyzed and provided for protection continuously varying with every case. Since there is a limited number of trained and experienced personnel that are able to verify the validity of these biases manually, it is necessary to find an efficient and inexpensive method, particularly one that does not require vast experience or an advanced level of training. Hence, it is dependent on computer methods [1]–[6]. It can store and retain sensitive and important information and data securely by eliminating data fraud, enhancing data safety and security. This is done by using the Hash function, which in turn provides high protection and privacy to the user, e.g., the message digest algorithm, which can be abbreviated as ''MD5'' and is a one-way cryptographic function used in various fields for maintenance and to demonstrate data integrity. However, the problem with the digital verification tools of the MD5 function lies in the collision, which many researchers report indicates a weak point of this encryption function [7]–[11]. Researchers have examined the weaknesses of the MD5 algorithm by conducting a large number of studies, research efforts, and analyses. The research has examined some types of attacks that are effective against the current algorithm, such as dictionary attacks, the rainbow table attacks, and brute force attacks. Researchers have discussed the weakness in the solutions that are currently used to solve the weakness of the MD5 algorithm in Web applications [12].

Researchers recommend that more powerful security and elasticity can be achieved by modifying the length of the current MD5 128-bit algorithm, as an output with variable length can achieve high performance and security. The research also proposes the use of a key to eliminate threats that commonly appear in rainbow attacks and dictionary attacks [13]. There are many weaknesses of the current MD5 algorithm, especially its failure and weakness against various types of attacks. Therefore, an enhanced MD5 algorithm with a dynamic variable length and high efficiency that simulates

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

the highest security is proposed. In this paper, a novel and efficient approach is presented to expand the size of the output hash of the MD5 algorithm. This approach adds further protection against known attacks, as a new idea is applied using the chaos theory and produces different fragment sizes to prevent all general hostile attacks. The goal is to generate a chaotic map that provides a random generation of values. As such, the output of the Hash is determined through the Runge-Kutta rules, which generates values based on the initial values that are obtained from the four output registers of the MD5 function, represented by A, B, C, and D, respectively. This provides a high randomness that is difficult for attackers to decipher. In this technique, the MD5 hash function is used to create the digest by modifying the initial conditions of all the chaotic maps used in the enhanced algorithm. Furthermore, the research shows that the development of the implementation of the improved MD5 algorithm plays a major role in maintaining the integrity of data from manipulation and forgery and increases the reliability and authenticity of the data.

Several methods have been proposed in the field of validation, including those presented in [14] where researchers have proposed a SIQ authentication protocol that relies on elliptic curve coding.

II. PREVIOUS WORKS

Due to the development of the present era, the high reliance on data transmission and information on the Internet for communication purposes required the encryption of data and information to identify and restrict malicious intruders from infiltration and access to read messages, and allows the process of encrypting information to users of the systems to transfer information and data with great security. From previous studies, it is possible to record and perceive that in many cases and modifications have been applying to the MD5 algorithm to address their security issues. However, there are several conceptual and methodological weaknesses that expose the chances of high error opportunities. A Message authentication system makes this authentication system more secure Several studies have reported that, in order to counter attacks according to the concept of secure systems assays in the handshake protocol, they need to authenticate all messages using the tick value so that the attacker cannot modify the file the algorithm that used to do this can be other the strongest algorithm or the weaker algorithm Therefore there was a need to pay attention to the structural algorithm of Hash to be stronger by adding strength and complexity while maintaining an acceptable level of execution time. The researchers suggested overcoming the limitations of the MD5 function by relying on parallel coding, as the researchers combined this coding between MD5 and Blowfish function in order to increase and provide sufficient strength and security [15]. Others present an implemented hybrid cryptography that uses both MD5 and the proprieties of an elliptic curve cryptosystem (ECC) to generate key steam [16]. Other strategies to enhancement MD5 done by applying the concept of steganography is combined with cryptography for Increasing security. The message was encrypted by MD5 hash function and applying the Advanced Encryption Standard (AES) [17], also the AES together with message digests (MD5) hash function to increase a security authenticity on cloud computing Proposed by [18]. Thus, this investigation will confer for modifying the round functions of MD5 Incorporating Hirose Compression Function with precise identification of rules and creation of a look-up table to improve the security of the cipher text [19]. This research will then modify the MD5 algorithm, in this paper, a novel technique to increase the retail size of the MD5 algorithm to implement further safeguard upon known retail attacks. The unusual purpose applied a mix of XOR and AND operators to produce a 1280-bit fragmentation size to prevent. Common attacks to increase security level to prevent brute force attacks and rainbow table [20]. Sponge formation inflexion of the MD5 and SHA1 hash functions [8]. The features of the MD5 algorithm were employed in several fields among which was the handling of the URL. This is to overcome, reduce and form the wrong positive actor and enhance the display of allocated crawlers [21]. Researchers have worked to maintain data security and integrity by adopting a proposal that provides a homomorphic partial RSA combination and MD5 hashing algorithm [22]. There is other research dealing with the use of the MD5 hash function, through the preparation of various plans and the adoption of multiple techniques in order to overcome the weaknesses inherent in the MD5 function. Among them, the researchers proposed a new plan, and the confusion and coding techniques are completely different due to the method of sequencing the block code of the jamming used at the bit level after stirring and DNA coding. In this procedure, the MD5 hash function is applied to formulate hash digestion to modify the initial conditions for all used messy maps [23]. The researchers proposed a new algorithm referring to Robust Integrity Verification Algorithm (RIVA). This algorithm helped to cancel the memory mappings of file pages after they were transferred, to improve the integrity of file transfers via checksum calculation tasks to read files directly from the disk [24]. This paper reports the design of a new hash algorithm with key integrity that accepts the conditions of message integrity and sources authentication. In this paper, a new hash algorithm has been introduced that involves a 64-bit key. The time taken by the proposed algorithm increases by 15-20% from the current security level. However, in cases wherever document security is the primary priority, regardless of time [25]. The features of the MD5 algorithm were used to protect devices when connected to the Internet and making it safe to be part of the Internet of Things [26]. A method to ensure Rabin CDC based duplication data reversal technology has been introduced to produce resizable parts and MD5 use in the cloud computing environment [27]. There are a lot of researches to increase diffusion of data one of these approaches based on a new method for rearranging both rows and columns with a fixed shift row depends only on the row to improve diffusion [28].

Different methods used to generate random number one of these depend on Mouse Movement with 3D Chaotic Logistic Maps a random binary sequences generator that produces sequence of bits [29]. In this paper was designed as a new idea for the development of a segmentation algorithm based on different technologies including the generation of a new key with key integration that serves the requirements of message integrity and sources authentication.

We have been able to build the general structure of our scheme that meets the research requirements to enchantment the MD5 algorithm by proposed dynamic methods with variable length and a high efficiency that simulates the highest security available, which exceeds many of the technical and structural weaknesses of many of the studies that have been found.

III. MESSGE DIGEST 5(MD5)HASH FUNCTION

The message-digest algorithm (MD5) as one-way cryptographic hash functions generally provides a 128-bit hash value. MD5 was designed by Ronald Rivest in 1991 to replace a more immediate hash function MD4, and was designated in 1992 as RFC 1321 [30].

A. MAIN STURCTURE OF STANDARD MD5 HASH FUNCTION

The conversion of text to another encrypted text by using the MD5 function, which can be summarized in five basic stages as shown in the following:

1. **Appending the padding bits as preprocess.** This step helped to make the length of the message corresponding to 448, modulo 512 by adding a number of bits between (1…512), so that the resulting message length is a multiple of 512.
   This will extend the message so that it is only 64 bits. The ''1'' bit is added to the message and then a set of 0 is appended so that the final block length matches 448.
2. **Append the length of message** In this step, the extension to the original message length was added, this extension represents the 64-bit length of the original message at the end of the bits in the previous step. If the message length is a k bit, we add the value k mod 2^{64}. The result of the previous two steps is a message of length L * 512. Suppose this letter consists of a collection of words of each word with a length of 32 bits. That is, the letter ranges from (0. . .N-1) such that N = L * 16.
3. **Initialize MD buffer** Recorders are configured as a four-word buffer (A, B, C, and D) to calculate the message digest. Each of these recorders is configured to be the 32-bit length in a hexadecimal byte and low order by using the following initial values:
   Initialize variables: (A = 67452301, B = efcdab89, C = 98badcfe, D = 10325476). The final result of the algorithm is found in these recorders, that 128 bits = 4 * 32.

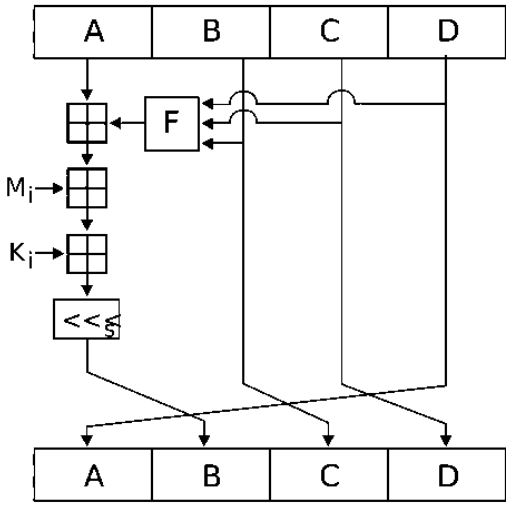

FIGURE 1. The block diagram of the main structure of existing MD5 schema.

4. **Process the message block** The processed in different steps to produce the required output. This step represents the most important step in the algorithm. In this step, 16-word message blocks are processed, which is equal to 512 bits. It consists generally of four cycles; each cycle has its own function:
   F, G, H, I, and in each cycle, there are 16 steps. In each round, 16 steps are performed on the recorders, where the output of words of this step is entered into four rounds in each round is scattered for these words.
5. **Output** After processing of all blocks, plain text is converted into ciphertext or hash form as a message digest, where the final value MD5 hash is 128 bits. See Figure 1, a detailed copy of the [MD5] algorithm can be obtained in[30].

B. WEAKNESS OF THE EXISTING MD5 ALGGORITHM

The research aims to develop a MD5 algorithm that can be used to create an electronic authentication system that makes this system more secure from other authentication system.

This would be an improvement when compared to the current MD5 algorithm, which contains much vulnerability that made it vulnerable to various attacks, including attacks with brute force, rainbow schedule, dictionary, Christmas, etc. Despite this, the current MD5 algorithm is still used in applications, including validation of data; security protocols, data transfer and storage for verification.

The research focuses on eliminating the weaknesses that are inherent in the current MD5 algorithm, thereby ensuring data integrity and secure. The proposed algorithm has focused to improve the MD5 algorithm by dynamically changing the length of its output and using a variety of techniques to aggregate data in its encrypted form.

The Christmas style attack varies from the brute force scenario as the Christmas attack is characterized by the size

of the data, an aspect that is essential. This makes it faster and more reliable than the brute force attack. The consequence of the Christmas attack is consequently, high if the size of the hash is small [31]

Dictionary Attack is another attack that uses a dictionary data list, which is applied as a password through which the message is determined by using all the available possibilities to penetrate the algorithm and to reverse the brute force attack. The speed of hashing with a dictionary attack is much faster than brute force technique [32]. The rainbow attack is considered to be the most accurate when compared to the dictionary attack and brute force. The rainbow attack is a table that is typically utilized to break down and cracking cryptographic hash functions, by reverse password hashes [33].

IV. MODIFICATION AND IMPROVEMENT OF MD5 ALGORITHM

This paper was designed as a new ideation for the development of a segmentation algorithm based on different technologies, including the generation of a new key with key integration that fulfils the requirements of message integrity and source's authentication.

A logistic system was used to encode the Ribonucleic acid (RNA)by generating a random matrix based on a new key that is created using the initial permutation (IP) tables that are used in the data encryption stander (DES) with a linear-feedback shift register (LFSR). The use of these tools helped to make balance between time and complexity.

A. PROPOSED ENHANCEMENT OF MD5

In this paper, a new hash algorithm is proposed. This includes a key, which is created using two different primitive irreducible polynomials to generate a different binary element of the field GF (2^m) from a base finite field with a degree (32) of the list. Next, the LFSR is applied to both polynomials by running the LFSR algorithm 64 times and reduces the output carrier number from each iteration by applying this method to two different sets of 64 bits from both polynomials (sq1, sq2). Subsequently, the XOR function is applied between sq1 and sq2 to obtain the initial key, and then the procedure of the initial permutation of the data encryption stander (DES) is used to create the main key. The main key is repeated as required until the size of the key is the same as that of the block message.

B. METHODOLOGY & KEY GENERATION

The issue of maintaining the privacy and security of data is an important matter in the application of electronic documents across different networks, particularly when dealing with unauthorized infiltration activities that attempt to see the data in the document and manipulate the content of these documents. These activities, if not curtailed, can lead to a creation of a fake copy or stolen data and acts of espionage. The MD5 algorithm used in the proposed system has been improved through these aspects

1. In this way, the explicit text has been converted to binary (0,1).
2. The text size was calculated by calculating the number of bits in order to enter it into the secret key generation algorithm proposed in this paper.
3. From the optimization, the first aspect is to increase the complexity that dependent on a linear-feedback shift register (LFSR).
4. Through using two levels of primitive irreducible polynomials of degree 32 through which a new key advantage can be generated.
5. By using two primitive irreducible polynomials of 32 degree to generate 32-series.
6. First two polynomial equations have been selected

$$\textbf{Primitive polynomial 1} = 1 + x^2 + x^3 + x^4 + x^8 + x^{16} + x^{31}$$

$$\textbf{Primitive polynomial 2} = 1 + x^3 + x^4 + x^5 + x^{15} + x^{20} + x^{31}$$

Second, create two binary vectors of each polynomial equation denoted to (poly1, poly2)
When:
poly1= [1 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
& poly2= [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 0 1]
the LFSR on each binary vector for 64 iteration and save the carry of each iteration in new sequence vector. The two-sequence vector are denoted to (seq1&seq2), that used later in the proposed system, see Figure 2 below that exposed the general structure of key generation.

7. Applying the linear-feedback shift register (LFSR) class on the first serial to output a new serial first bearing the 64-bit length Sq1.
8. Applying the LFSR function on the second serial to output a new serial second bearing the 64-bit length of Sq1.
9. Extracting the initial key through the XOR between (sq1, sq2).
10. Generating the main key by entering the primary key that was generated from the previous step to the initial permutation tables that were used in DES to output 64-bits, where the LFSR 64 roles are applied in each cycle. The load is extracted to be the sum of the other 64-bit outputs, and this would differ from the previous 64-bits.
11. Dividing the message file into several blocks each block has 768 bits instead of 512 bits, for more speed.
12. In the proposed system, RNA coding techniques will be somewhat different because the method of encoding cryptography is used at the bit level after flipping

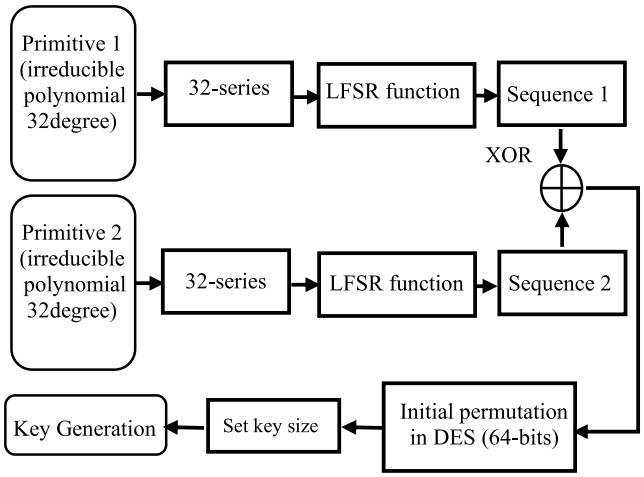

FIGURE 2. Shown the main structure of crating the key generation.

and using multiple permutation methods and RNA encoding. MRNA coding is built by relying on the outputs of the binary text with the outputs of the secret key generated in the previous step, where the construction method was based on taking two bits of the binary text, where it is combined with two bits with the outputs of the secret key and the result is compared with the MRNA table for Coded.

13. Each block is encoded using the product key of the previous step that uses Ribonucleic acid (RNA) Where the RNA chain is transformed to binary codes.
14. It is extracted from MRNA coding the message, which is in the form of decimal numbers, each number is composed of 10 orders, as well as nBits, which represents the number of bits.
    These parameters drawn in the fourth step are the main inputs to the stages of hash formation in MD5 in order to obtain hash.
15. The outputs of these codecs are processed to make them able to be inputted into the MD5 algorithm.
16. The output of four registers (A,B,C,D) of the MD5 algorithm are extracted as input to Fourth-order Runge-Kutta method that used to solve the differential equation in the delay function to obtain high-quality hash value that encrypted by switching of chaotic neural network maps [13] see Figure 3 that shows general flow chart of the proposed system.

The length of the Hash is variable, as it has the dynamic ability to change the length of the Hash depending on the type of Hash required. See Figure 4 that demonstration the general block diagram of the proposed system.

The Output hash size = $32^{\text{hash type}}$; where the hash type= [1,2,3,4 .... n]

That mean it can create different hash size as shown in the following: -

(if hash type=1 that mean hash size = 128 bit(32 hex)
&if hash type=2 that mean hash size = 256 bit(64 hex)

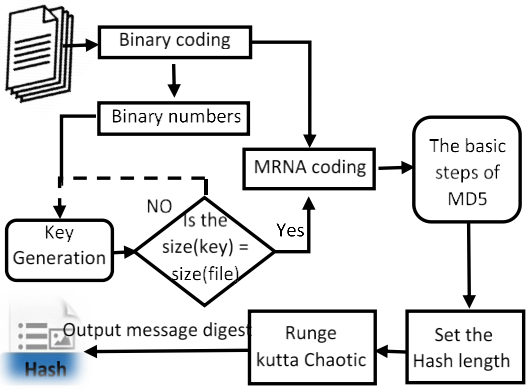

FIGURE 3. The general flow chart of the proposed system.

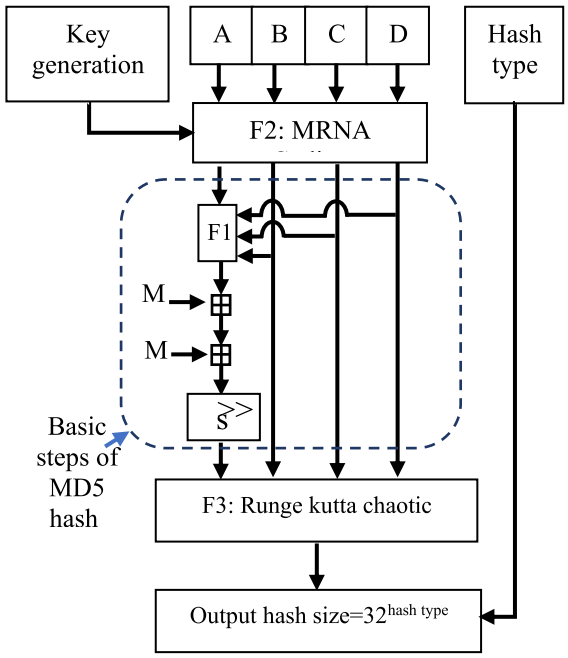

FIGURE 4. Show the general block diagram of the proposed system.

&if hash type=3 that mean hash size = 384 bit(96 hex)
&if hash type=4 that mean hash size = 512 bit(128 hex)
and ... so on)

V. PROSPOSED METHODS

In this work a novel five different methods are used to generate a new different robust hash from MD5 function. The amount of data that required to be analyzed and provided for protection varies continuously from one case to another. it is necessary to find an efficient and inexpensive method, and do not require high experience or advanced training level. that depend on computer methods such as MD5 Hash function. Five novel methods have been proposed in this research paper to improve the performance of an MD5 hash function as shown below: -

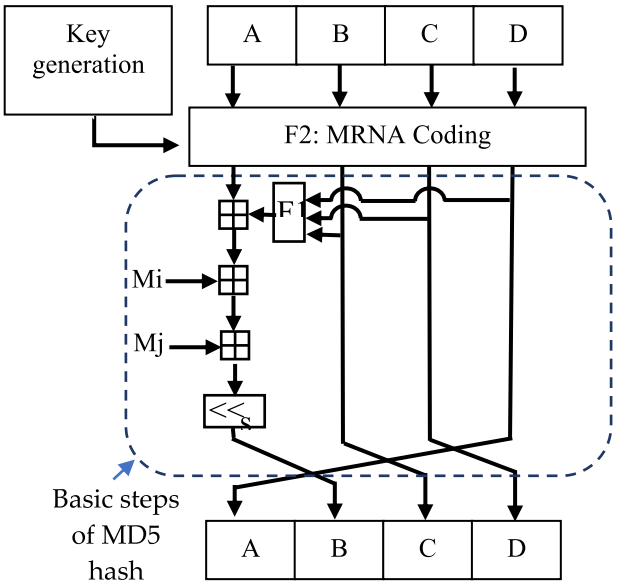

FIGURE 5. Shows the general block diagram of the proposed system (Method 2).

*Method 1:*this method exposed the fusion of MD5 with chaotic theorem (Fourth-order Runge-Kutta method that used to solve the differential equation) & new security key ; the key size that used in this method is the same as the file size the main structure of this method demonstrated in figure 1 above. The output hash length is (128-bit, 256-bit, 384-bit, 512-bit …etc).

Method 2: this method exposed the fusion of MD5 with a new security key that used in method 1 that was explained previously in the section (5. Methodology & Key generation), the length of key is the same of plain text. The output hash length from method 2 is (128 bit) and this length is the same length that created by Basic MD5 hash but with different values, see Figure 5.

*Method 3:*This method creates MD5 hash value by fusion of Basic steps of MD5 hash function with chaotic theorem only. The chaotic plane depends on (Fourth-order Runge-Kutta method that used to solve the differential equation) that used in method1. The output hash length is (128-bit, 256-bit, 384-bit,512-bit …etc). The value of method 3 differs from the value generated by method 1, which can be observed clearly in experimental results.

*Method 4:*This method is showing a light version of method 2 that fusion of MD5 with a new security key that used in method 1 that was explained previously in the section (5. Methodology & Key generation), the length of key is fixed that is predetermined by the user. The key is engaged and injected into the text to be encrypted by performing a function XOR and the insertion key take two form (other fixed steps or fixed ratio from the original file) the same of plain text. But the output hash length from method 4 is (128 bit) and this length is the same length that created by Basic MD5 hash but with different values see figure 6.

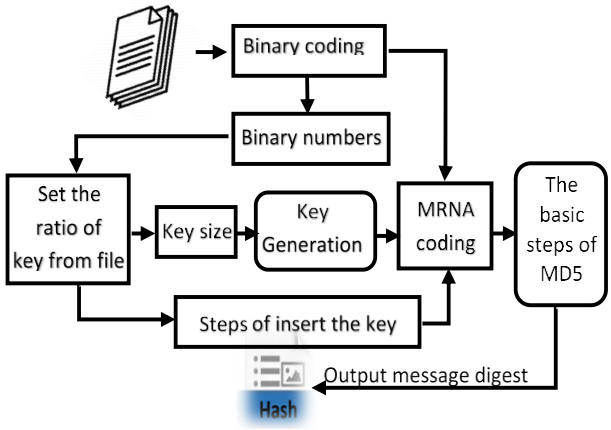

FIGURE 6. Shows the flow chart of general structure of the proposed system (Method 4).

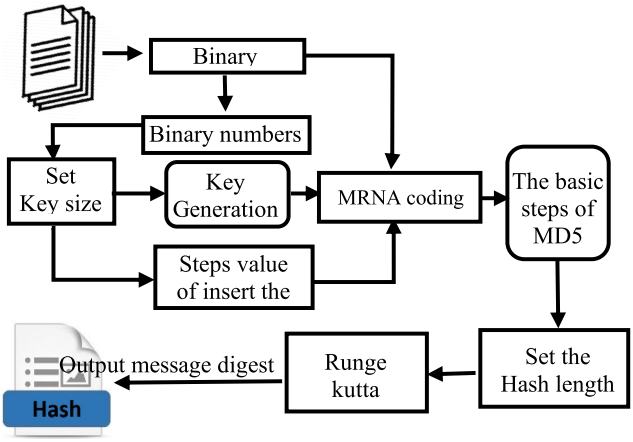

FIGURE 7. Shows the flow chart of the proposed system method 5.

Method 5: This method is showing a light version of method 1 because it used two above method to get the output hash value these methods are Method 3 & method 4 that fusion of MD5 with chaotic theorem & new fixed security key. The output hash length is (128-bit, 256-bit, 384-bit,512-bit …etc). The value of method 3 is differ from value the generated from method 1 and method 3, see Figure 7.

VI. BASIC MEASURES OF PERFORMANCE

This section discusses the confidentiality conditions performed by the suggested approach, which show the robustness of the algorithm as compared to other types of hash algorithms.

A. WEAKNESS OF CONFIDEDTIALITY REQUIREMENTS FOR THE SYSTEM SUCCESSFUL

In order for the system to be confidential, the designer of the system must take into account in its design to meet the basic requirements that achieve the confidentiality of the system. The *Confusion* is an important concept, for designing of the

program, the *Confusion* make sure to fill all the gaps through which the code breaker can analyze the system to know the explicit text through its knowledge of the encrypted text.

At the same time *Diffusion* appear as another important concept, *Diffusion* is When designing a program, you must make sure that all logical relationships are cancelled and the analyst cannot identify the decryption algorithm and find the key. The successful algorithm that prevents the analyst breaks it even if he can know the explicit text. The degree of secrecy the regime has depends on the possibility of breaking it. A highly confidential system is preferred. It is possible that the system is theoretically fracture but difficult to break in practice because the encryption process does not have a known technique for code analysis explicit text or key used.

The proposed new reduction algorithm is based on the MD5 reduction algorithm with better grasp of the excellent functions of the other algorithm. It can be noted that the hash values that produced by our proposed algorithm are different lengths depending on how the data is important, that values to be encrypted (128, 160, 256, 384, 512, 1024, 2024) compared to a 128-bit value that produced only by the MD5 algorithm. Despite the high flexibility and dynamic scalability of the hash values in this algorithm, I have managed to maintain the same execution time for all outputs of lengths per file. This makes our proposed algorithm more difficult to break because of the increased level of security it has achieved.

B. THE SIMPLICITY OF THE KEY

The simplicity of the key and the ease with which it is remembered are the most important characteristics of which the key is preferred. In the proposed algorithm, the choice of keys depends on the diversity of features that give to the key the strength and the simplicity at the same time.

C. THE COMPLICATION OF BOTH THE ENCRYPTION AND DECYPTION METHO

A suitable encryption system should be easy and flexible to decrypt, though, at the same time, the process of analyzing and breaking the code should be difficult and not possible. This is done by making the encoding time and decoding time efficiently (polynomial time), while the time it takes to decode the code is (exponential time). After adding the expansion function in the proposed algorithm to develop a function, MD5, the algorithm needs more further time than the traditional algorithm of MD5, but the difference is not large as it increases and slightly by the length of the message.

However, compared to other reduction algorithms such as SHA 1, for example, the time of implementation of the proposed algorithm is less expensive, and for the complexity of the system, the complexity rate of the algorithm can be calculated from $(2 \wedge 64)$ because the algorithm consists of 4 cycles per cycle Of 16 steps. $16 * 4 = 64$). If compare with other algorithm like SHA1 hash algorithm that has more complexity of $2 \wedge 80$ that consists of 4 cycles per 20-step cycle.

The suggested algorithm consists of 4 cycles in each 16-step cycle, and expansion equations were added to the algorithm to increase the relative complexity of the algorithm. The values of the proposed hash algorithm have a dynamic length such as (128, 160, 256, 384, 512, 1024). Despite the variety of outputs of lengths, the time taken to obtain such lengths for the same file is very close. As for breaking the code, the process is very complex and difficult for the codebreaker. For example, if the output length of the hash value is 160, it is the same as the length of the value produced by the SHA1 algorithm, so they have the same complexity in breaking the code.

VII. EXPERIMENTAL RESULTS

The results of the experiment show that even if the time taken for the proposed algorithm is relatively more in relation to the required level of safety, it will exclude any of the techniques used in the hybridization algorithm, such as the key creation method and use of the coding method that used RNA or the use of chaos theory, such as Brute force and other Attacks. One of the major advantages is that the MD5's improved algorithm can be expanded to any size greater than 64-bit key block length so that the output is safer against any attack. The proposed improvement and development of the algorithm has been tested through several experiments:

The first stage: The first stage was to test the algorithm for collision. The experiment was applied to 1000 text files of different sizes and the result was returned to no collusion.

For the second stage: Several tests were performed on the algorithm by examining the algorithm's resistance to penetration and code breaking attacks. Among these tests, a test was conducted for each of the five methods to novel improve the hash algorithm used in this search to, using online penetration sites where all the sites used to test the strength of the improvement that were performed on the hash algorithm to break the hashtags were failed as among these sites like (Crackstation.net, MD5 decryption.com, md5.web-max.ca/, md5online.org and Hashkiller.co.uk)

Where the hybridization of the encryption algorithm was made using various coding techniques and steps, where the complexity was greatly increased with a relative preservation over the time required for implementation as the hybridization methods used in this paper varied, and this diversity had a great impact in obtaining high dynamics by obtaining hash values. Sub-tests were performed, in which the number of bits extracted from the four registers was changed in order to serve as an introduction to Ren Kota's laws as a chaotic map. This method achieved great success by keeping time relatively, regardless of size of fragmentation. The speed fluctuates near the speed that appears with an unchanged algorithm. Figure 1 shows a comparison of multiple methods used to improve the actual MD5 algorithm. The use of this method had high flexibility in outputting the desired length with different values, as we can obtain the length of 128, 254, 384, 512, 1024 ...etc. from the hash values. If (8 digit) feed to rung kutta chaotic. and we can obtain the length

TABLE 1. Illustration of the different output bits in enhancing the MD5 algorithm that gives the same structure to other.

Type of method	Number of output Bits in new enhancement md5				The output digit of enhance md5 is the same structure to:	
new MD5	8 digits	9 digits	10 digits	11 digits	8 digits	10 digits
1	128	144	160	176	MD4 or MD5	SHA-1
2	256	272	288	304	SHA-2-256 or FSB-256	
3	384	400	416	432	SHA-2-384 or FSB-384 or ECOH-384	
4	512	528	544	2048	SHA-2-512 or FSB-512	
8	1024	1040	1056	1072	VSDL-1024	
16	2048	2064	2080	2096	VSDL-2048	

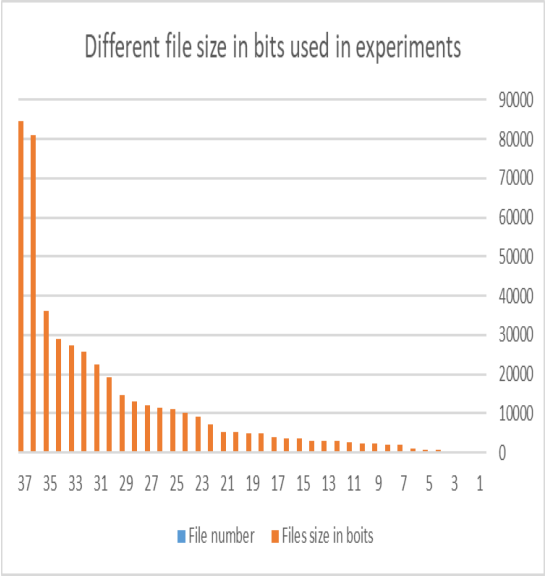

FIGURE 8. Show different file size in bits that used in experiments of all methods.

144,272,400,528,1040,2064) hash length if (9 digit) feed to rung kutta chaotic and 160,288,416,544,1056,2080 hash length if (10 digit) feed to rung kutta chaotic and do on more detail can be seen in table1. Samples of different file size used in experiments of all methods illustration in figure8. This is to prevent the hacker from knowing the algorithm used and trying to break it. And increase the complexity of it by increasing the length MD5 hash value.

The technique proposed in MATLAB R2018b has been implemented and implemented several times on a system containing: -

The processor used in this sheet is the Intel Pentium i76500U processor that runs at 2.5GHz 2.59GHz

- 12 GB RAM. 64-bit
- It was the operating system (Microsoft Window 10).

VIII. EXPERIMENTALA STATISTICAL ANALYSIS OF EXPERMENTAL RESULTS

In order to prevent all malicious and unauthorized access to the secret digital data that data is exchanged and transmitted over the Internet or any internal network, this is done through preventive measures that are achieved to reach the highest degree of confidentiality and security

This paper was introduced in order to share and protect the transmitted information that is sent through the computer and telecommunication network. Therefore, consideration should be given to security as the most important and serious challenges of sharing information through the network.

Security and confidence are a very significant issue that should be available in the case of communication and sharing of data resources between computers.

Consequently, Encryption techniques have been adopted as an important and effective data encryption function as well as authenticated by other users.

The encryption techniques used are the main tool to maintain the security and integrity of information through its methodology, the encrypted messages will be unreadable.

It can be noted that in these days the encryption has become the main savior of the security problem of Internet communications, as it provides important points and features such as (data integrity, non-repudiation in addition to confidentiality and data authentication).in this section a statistical analysis of all our proposed method in this paper are doing.

A. TEST METTHOD-1

Write the hypothesis for T test:

H0 = Time≤ Method 1

H1 = Time≥ Method 1

Writing a test hypothesis for Pearson Correlation:

H0: There is no linear relationship between Time and Method 1 for results 1, 2, 3, 4

H1: There is a linear relationship between Time and Method 1 for results 1, 2, 3, 4

The results of T test and Correlation obtained through applying Statistical Package for the Social Sciences software that denoted to (SPSS) version 23 for editing and analyzing all experiment results in this work as follows:

B. TEST METTHOD-2

Write the hypothesis for T test:

H0 = Time≤ Method 2

H1 = Time≥ Method 2

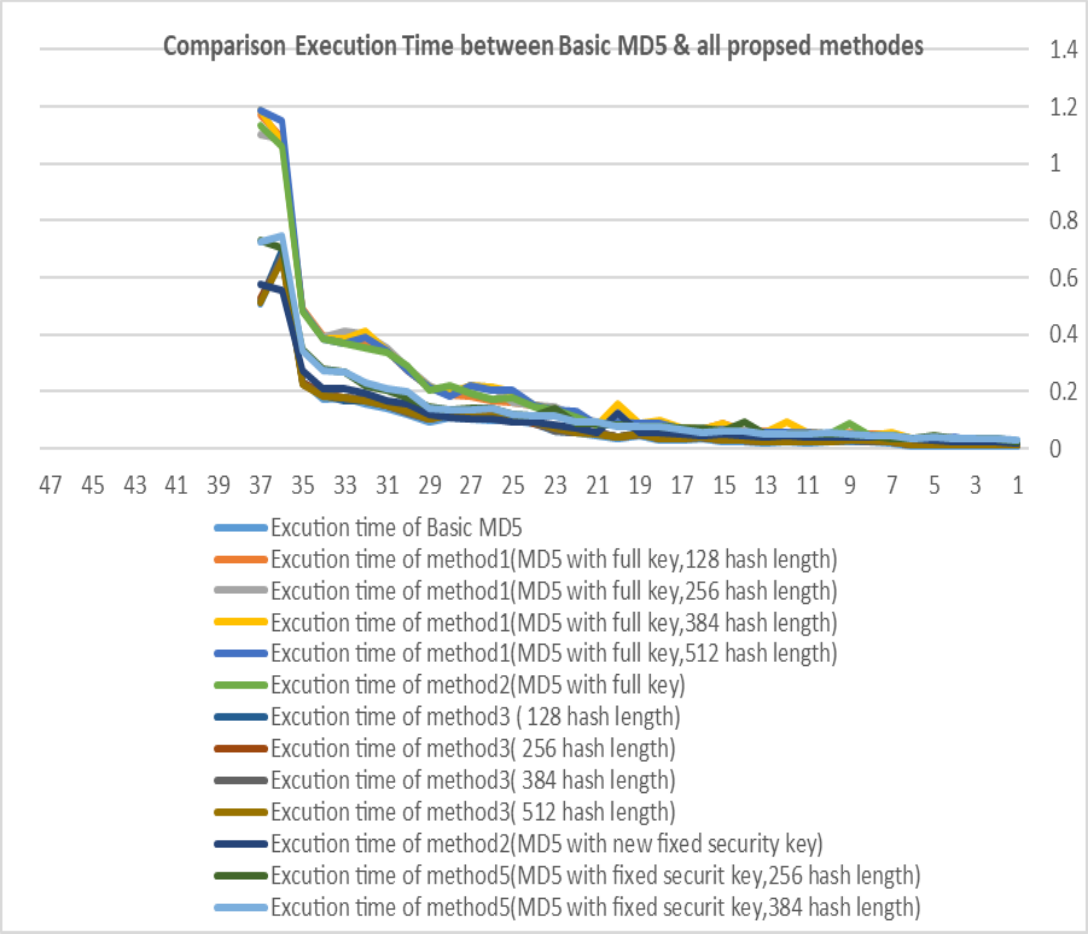

FIGURE 9. Comparison of the run time between the multiple development methods with the actual way.

FIGURE 10. Comparison of the execution time between method 1 and Basic MD5 hash function.

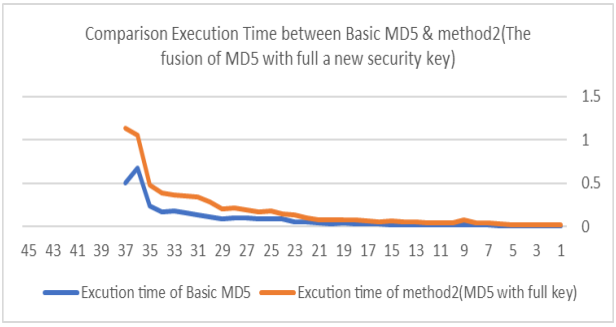

FIGURE 11. Comparison of the execution time between method 2 and Basic MD5 hash function.

Writing a test hypothesis for Pearson Correlation:

H0: There is no linear relationship between Time and Method 2 for results 1

H1: There is a linear relationship between Time and Method 2 for results 1

The results of T test and Correlation obtained through applying the SPSS statistical analysis program for the data are as follows:

C. TEST METTHOD-3

Write the hypothesis for T test:

H0 = Time≤ Method 3

TABLE 2. Samples of experimental results that demonstrate comparison hash value with execution time the evaluation occur between basic MD5 & method1 (The fusion of MD5 with chaotic theorem & new security key).

File size (bits)	Basic MD5	Execution time	Modify MD5							
			128-bit hash length	Execution time	256-bits hash length	Execution time	384-bits hash length	Execution time	512-bits hash length	Execution time
160	e807f1fcf82d132f9bb018ca6738a19f	0.01	91a561a790b9a0e2b1bc31f6c18b8116	0.014117	1a6544a41a7914650ea988b80e2908a91f3c00b01f6b103c118b4484116c148b	0.015028	555a547955671a9a7955619a999e982999a20b9b2999a09bcccfc06bcc361bbb6bcc31bbbbb1b46cbb8618c86cbb81c8	0.014813	9aaa676497a176a491aa11191991aaa79eeea2a892e02ab890ee00090990bee2bfff3630b6f163b0b1ff111b1bb1bff6c1118684c6116884c111111c1cc18116	0.014
2384	ee60ff27ad9b7e8a3787b6cb010d468a	0.0221	db9d8b6e472a47508bdd6bae806380de	0.053828	b68d6696b6edb68d754a5525750475 4aba6d77d7bae8b76d0d8322620de80283	0.046203	ddd6d6eddd8eb9d9eddd8bd9aaa5a504aa40724204aa4742dddad7e8dd6ebd8de8dd6b8d333d32e8338e0686e8338086	0.050036	d6668e86de6be896db66bbbdbddb966e4555404540570425475577747447255 08aaa6e678eabe6d78baabbb8b88bdaae8ddd8e828ed0e86280dd000808806dde	0.046
5112	91fc2e01d4d631a8ffdae68ba2cb842b	0.0346	596df9072a948a381e908ea8e62f26e2	0.084218	90fd2262907592fda384ff9fa382af84ea809999ea81e9806e2f88286e2e682f	0.091784	ddd0d275ddf7965675ddf95644434f824488a92982448a29000a09810088e91981008e19fffef82eff2262e22eff26e2	0.158534	5000f7f257097f625900999595596007233388 8f283a889f2a33aaa2a22a93381aaa888918ae88991eaaeee1e11e9aa8eeee2228e2e62228e6ee666e6ee62ee2	0.013954
27464	9c57423c69aa25a24406290d754173fe	0.1776	bb175baf4a27caa2a27352ac35997533	0.377776	ba578818bafbb857aac7dd2daa24adc72a5322722aca22535379119 1533351 79	0.408888	777a78fb775fb1b1fb775bb1777a7d2477c2a2422477ca42333a32ca335c27a7ca3352a799939133997 35939339975 39	0.385753	baaa5f58bfabf518bbaabbbbbbbb1aaf4aaac2cd42aa2c2d4aaaaaa4a44a2aa2aaaa5c52aca2c572a2aa222a2aa27aac333373713 33537913533555 353359333	0.04595
84432	8ce9b670d109c76a87ccb32063b0db84	0.5063	109440e5800350fcf2df2290c32023a4	1.168899	0e4433930e5103440f53dd0d0fc80d53292f88d8290f282f3a2099293a4c3920	1.100128	444e43514445 09195144401 9333f3dc8335c0080c8335080fff9f80fff202dfd0fff22fd000a094c002432c24c0023c2	1.186662	1eee454315e0549310ee000101109ee58fff5c5d8cf0c50d80ff000808800ffcf9992028f09202d8f299222f2ff2d990caaa2429c4a34229c3aa333c3cc32aa4	0.092991

TABLE 3. Samples of experimental results that demonstrate comparison hash value with execution time the evaluation occur between basic MD5 & method2(The fusion of MD5 with full a new security key).

File size	Basic MD5	Execution time	Modify MD5	Execution time
160	e807f1fcf82d132f9bb018ca6738a19f	0.01	17a694a502ea98b916f3b0bc1618c48b	0.022224
2384	ee60ff27ad9b7e8a3787b6cb010d468a	0.0221	be68d69d7054452abea687dd0ed88263	0.085263
5112	91fc2e01d4d631a8ffdae68ba2cb842b	0.0346	970f526da8382f94e8a8199062e2e82f	0.080533
27464	9c57423c69aa25a24406290d754173fe	0.1776	bfa5b817a2ac4d272ca5a27353373199	0.368277
84432	8ce9b670d109c76a87ccb32063b0db84	0.5063	05e413940cf58d032092f8df34a2c920	1.136157

TABLE 4. Samples of experimental results that demonstrate comparison hash value with execution time the evaluation occur between basic MD5 & method3 (The fusion of MD5 with chaotic theorem).

file size (bits)	Basic MD5	Execution time	Modify MD5							
			128-bit hash length	Execution time	256-bit hash length	Execution time	384-bit hash length	Execution time	512-bit hash length	Execution time
160	e807f1fcf82d132f9bb018ca6738a19f	0.01	fefc7e081f2fdf2819ca09bba69f8637	0.0119973	e07c11f1e08fe17cf2df3323f281f3df9b0a88c89bb1980a638f1191637a618f	0.0138767	ccc0c18fcc78efff8fcc7effff2f381ffd8f21281ffdf12aaaba8b1aa0b9c1cb1aa091cff3f17aff8769a97aff86a9	0.0118255	f0007871f80e87f1fe00eeefeffef0081222d8d3182f8d231f22fff1f11f22281bbb0b081bb9b0c819bb99919119cbbba3338781a7367891a633666a6aa69337	0.0130283
2384	ee60ff27ad9b7e8a3787b6cb010d468a	0.0221	fe270e6e7a8aba9db3cb7387408ad001	0.0263939	e607ff2fe6efef07a9baee8ea9d7aeba387b66c6387b367b00da6686001406da	0.0273017	77767fef770ee2f2ef770ef2aaa9aed7aabda878d7aaba78bbb8b67bbb773cbc7bbb73bcaaa0a614aad1084814aad048	0.0271411	f6660e0ffe6ee02ffe66eeefeffe266e7999bdbe7d9adb8e7a99aaa7a77a899db8887776b78377c6b388333b3bb3c8874000d1d641001d8640000004044 08001	0.0270809
5112	91fc2e01d4d631a8ffdae68ba2cb842b	0.0346	2901c9f13da86dd4ef8bafdf8a2bbac2	0.0388152	9fc1ee0e9f129ec1dd6811a1dd43d168fdab6686fdfef6abacbb4424ac28a4bb	0.0411858	111f1e1211c190201211c920888d81438864da3a43886d3abbbdb6febbaff8e8febbafe8bbbcb428bbb2a28228bbba82	0.0396773	2fffc1ce21f91c0e29ff999292290ff13ddd646134dd46a13dddddd3d33daddd4edddafa6efdffa86efddfffefeef8ddf8cccb2b482ca2b248accaaa8a88a2cc2	0.0414856
27464	9c57423c69aa25a24406290d754173fe	0.1776	493c795c26a2a6a9240d640477fe1745	0.1685247	957c223295c4927c6aa255a56a9265a2406d9909404249 6d741e33f37457731e	0.1780695	ccc5c2c4cc7c9343c4cc7943222a259222a96a2a9222a62addd0d942dd64402042dd6420eee4e357ee157f7f57ee177f	0.1763763	45557c724c59c73249559994944935 5c2aaaa9a529a69aa526aa66626226aaa920006469240446092400444242240004744415137 54751f3774477777777f445	0.1792353
84432	8ce9b670d109c76a87ccb32063b0db84	0.5063	b87098eccd6a9d01b820c8c7d68406b3	0.518698	8e9066768ecb86909d09a77671d01cd79a8cc033238c7b83c06b04bb8b6b3d6b04	0.5299879	000e06cb009c87b7cb0098b7aaa0a71caa91d6c61caa9dc6000c037b00c782b27b00c8b2444b4b3d440368d83d4406d8	0.5123946	beee9c96bce8c976b8ee888b8bb87eecc0009197c10d1967cd00dddcdccd6001bcccc7c3b7c87c23b8cc888b8bb82cc7dbbb030bd3b6308bd6bb666d6dd68bb3	0.512246

TABLE 5. Samples of experimental results that demonstrate comparison hash value with execution time the evaluation occur between basic MD5 & method4(The fusion of MD5 with new security key).

File size	Basic MD5	Execution time	Modify MD5	Execution time
160	e807f1fcf82d132f9bb018ca6738a19f	0.01	427908f85ed463f8a09226a9bc613716	0.016571
2384	ee60ff27ad9b7e8a3787b6cb010d468a	0.0221	78b083638434e9e44a9295055991c892	0.0397616
5112	91fc2e01d4d631a8ffdae68ba2cb842b	0.0346	c996161a73318c035bcdf356e4e298ed	0.1269134
27464	9c57423c69aa25a24406290d754173fe	0.1776	6242fe458ddbb01242c39183cc84b501	0.2110465
84432	8ce9b670d109c76a87ccb32063b0db84	0.5063	c429c4c1e68589fcb6f610ce7bf3de1e	0.5751064

H1 = Time≥ Method 3

Writing a test hypothesis for Pearson Correlation:

H0: There is no linear relationship between Time and Method 3 for results 1, 2, 3, 4

H1: There is a linear relationship between Time and Method 3 for results 1, 2, 3, 4

The results of T test and Correlation obtained through applying the SPSS data analysis program are as follows:

TABLE 6. Samples of experimental results that demonstrate comparison hash value with execution time the evaluation occur between basic MD5 & method5(The fusion of MD5 with chaotic theorem & new fixed security key).

File size	Basic MD5	Execution time	128 bit hash length Modify MD5	Execution time	256 bit hash length Modify MD5	Execution time
160	e807f1fcf82d132f9b b018ca6738a19f	0.01	479888f8472048985d4833f3 5de65348a92966a6a902a629 b6167717b6c3b716	0.026088	8887882088924f0f2088940f888d83e68 84e5f6fe688456f999996029920aa2a029 92a2a666667c3661cb131c3661b31	0.028544
2384	ee60ff27ad9b7e8a37 87b6cb010d468a	0.0221	7b0333637b887303834499e 9834e89444925550549a9452 559128898599c5812	0.048139	333b33883308768688307864443494e 44448eee4e4448ee555955a9552a4090a 95524902229289c221959c99c2215c9	0.051877
5112	91fc2e01d4d631a8ff dae68ba2cb842b	0.0346	c96a6616c991c66a7313cc0c 73387c135cd633535cbf53d6 ee2d88e8ee49e82d	0.082421	aaa9a691aa69c11191aa6c1133333c383 3137080383317806666c63bf66db55f5bf 66d5f5ddded849dd24ee9e49dd2e9e	0.076291
27464	9c57423c69aa25a24 406290d754173fe	0.1776	6425ee4e642f6e258db20010 8ddb80b24c3311814c294133 c8415505c8cbc541	0.265962	55545e2f552264f42f5526f4222d20db22 bd81b1db22b8b1333c31293332489829 333498111815cb114cc0b0cb114cb0	0.269879
84432	8ce9b670d109c76a8 7ccb32063b0db84	0.5063	c29144c4c24cc491e85c99f9e 868e95cbf6e00c0bf61b06e7f 3eee1e7fbd7e3e	0.729230	1112144c1194cccc4c119ccccc8c968cc 56ef8f68cc5e8feeefe061ee66bc1c61ee6 b1ceeefeebdee3b71d1bdee37d1	0.725977

Note : You can find more details of all experimental in the supplementary files of this paper .

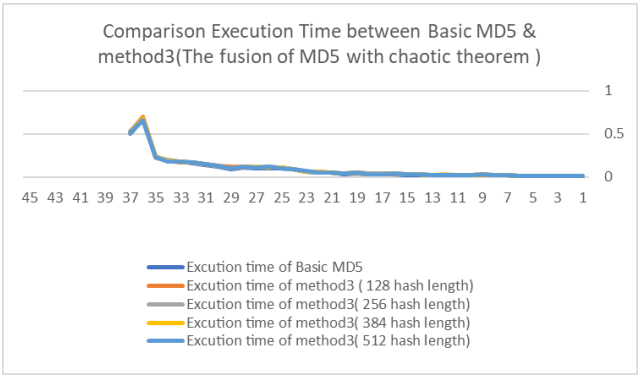

FIGURE 12. Comparison of the execution time between method 3 and Basic MD5 hash function.

TABLE 7. Statistical analysis of method 1.

Proposed Method	METHOD	T test	Pearson Correlation	Sig. (2-tailed)
Method 1	Method (1,1)	-4.572	0.980	0.000
	Method (1,2)	-4.906	0.982	0.000
	Method (1,3)	-4.919	0.975	0.000
	Method (1,4)	-4.624	0.984	0.000

TABLE 8. Statistical analysis of method 2.

Proposed Method	METHOD	T test	Pearson Correlation	Sig. (2-tailed)
Method 2	Method (2,1)	-4.691	0.980	0.000

TABLE 9. Statistical analysis of method 3.

Proposed Method	METHOD	T test	Pearson Correlation	Sig. (2-tailed)
Method 3	Method (3,1)	-6.165	0.999	0.000
	Method (3,2)	-6.562	0.999	0.000
	Method (3,3)	-7.944	0.999	0.000
	Method (3,4)	-6.150	0.999	0.000

The results of T test and Correlation obtained through applying the SPSS data analysis program are as follows:

D. TEST METTHOD-4

Write the hypothesis for T test:

H0 = Time≤ Method 4

H1 = Time≥ Method 4

Writing a test hypothesis for Pearson Correlation:

H0: There is no linear relationship between Time and Method 4 for results 1

H1: There is a linear relationship between Time and Method 4 for results 1

E. TEST METTHOD-5

Write the hypothesis for T test:

H0 = Time≤ Method 5

H1 = Time≥ Method 5

Writing a test hypothesis for Pearson Correlation:

H0: There is no linear relationship between Time and Method 5 for results 1, 2

H1: There is a linear relationship between Time and Method 5 for results 1, 2

TABLE 10. Statistical analysis of method 4.

Proposed Method	METHOD	T test	Pearson Correlation	Sig. (2-tailed)
Method 4	Method (4,1)	-3.956	0. 978	0.000

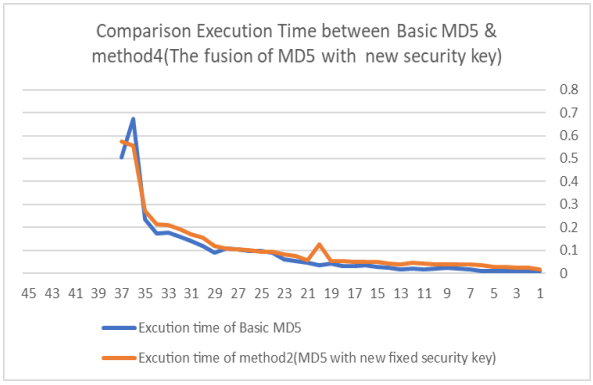

FIGURE 13. Comparison of the execution time between method 4 and Basic MD5 hash function.

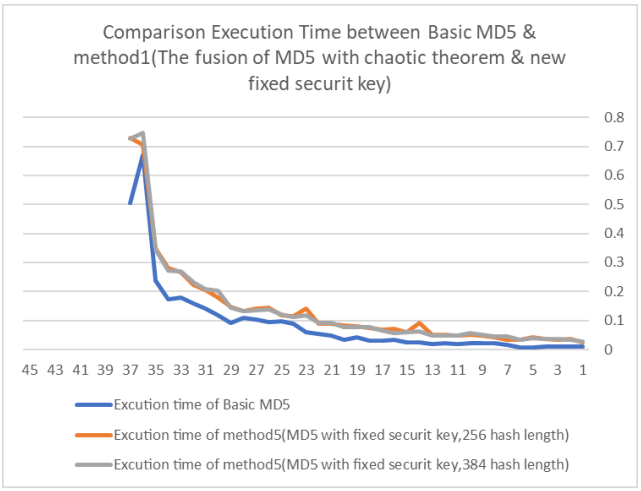

FIGURE 14. Comparison of the execution time between method 1 and Basic MD5 hash function.

The results of T test and Correlation obtained through applying the SPSS statistical analysis program for the data are as follows:

Which includes the results of the statistical tests of the five methods, by setting the tabular value (t) at a significant level (0.05) and since the calculated (t) is greater than the tabular (t) (1.686) and since its value is negative, the H0 hypothesis cannot be rejected, that is, there is no difference Significant between the time taken to implement the original MD5 function and the time taken to implement the MD5 developed function, where the Pearson Correlation values for the five methods were as follows:

The first method recorded the values of the sub-tests for this method, which represent the hash length

TABLE 11. Statistical analysis of method 5.

Proposed Method	METHOD	T test	Pearson Correlation	Sig. (2-tailed)
Method 5	Method (5,1)	-7.766	0.982	0.000
	Method (5,2)	-7.859	0.988	0.000

128,256,384,512 and compared it to the original function test without development, as these values were as follows: (0.980), (0.982), (0.975) and (0.984), respectively.

Whereas, the values of the second method of Pearson Correlation were (0.980).

At the same time, the third method recorded a notable superiority through the Pearson Correlation values for the sub-tests of this method, which represents the hash length of 128,256,384,512 and compared it to the original function test without development which is (0.999), (0.999), (0.999) and (0.999).

While the fourth test was the value of the Pearson Correlation for him (0.978) and finally the fifth method in which two lengths of margin were chosen only 256 and 384 where the values of Pearson Correlation were as follows (0.982) and (0.988)

It has been observed that the value of Sig. (2-tailed) is (0.000) for all values, which leads to rejecting the null hypothesis (H0), meaning that the time of implementation of all the methods proposed in this research is strongly correlated with the values of the original time. This confirms that the correlation coefficient value can be adopted. Accordingly, we can conclude from the above that any of the methods used in this research can be adopted.

More details of the statistical operations can be found with the tables of all the tests for the proposed methods in the attached file, where all the hashtags generated for all the methods have been attached with the time required to implement each method and the details of the statistical relationships associated with it.

IX. CONCLUSION

MD5 is one of the functions or techniques of one-way segmentation that is used in many fields and in different applications to maintain data integrity by converting plain text or data into encrypted text that is generated in the form of unique hash data. This helps prevent any manipulation of it, even if the level of change is just one bit. The development of the MD5 algorithm is conducted through dynamic behavior, and the extraction of the MD5 algorithm is developed by

varying the length of its output (128, 160, 256, 384, 512, 1024, and 2048) and other lengths, as can be clearly observed in Table 1. This is done because the length of the output of the MD5 algorithm is often considered a weak point of the algorithm. Through this dynamic method, the algorithm is made robust against all types of attacks because the algorithm simulates the long lengths of multiple algorithms without affecting the speed of the output of the algorithm. All the length types of the Hash are approximate in a similar way, despite occupying different spaces in the treasury of outputs, such as MD4, SHA-2-256, FSB-256, SHA-2-384, FSB-384, ECOH-384, SHA-2-512, FSB-512, or VSDL-2048, as shown in Table 1. It also has the key technology, which was created to give strength and resilience against attempts at penetration and manipulation. Due to these properties, there have been many modifications to the algorithm to address its pre-existing security issues. Indeed, the MD5 algorithm suffers from the above attacks because of the smaller values of the hash digest. The improvement in the proposed MD5 algorithm results in an improved collision resistance and access to the highest levels of security by providing the distribution and creation of bits such that it is difficult for attackers and hackers to predict and change individual data.

REFERENCES

[1] F. Alaa Kadhim, G. H. Abdul-Majeed, and R. S. Ali, "Enhancement CAST block algorithm to encrypt big data," in *Proc. Annu. Conf. New Trends Inf. Commun. Technol. Appl. (NTICT)*, Mar. 2017, pp. 80–85.

[2] A. M. Ali and A. K. Farhan, "Enhancement of QR code capacity by encrypted lossless compression technology for verification of secure E-Document," *IEEE Access*, vol. 8, pp. 27448–27458, 2020.

[3] A. K. Farhan, R. S. Ali, H. Natiq, and N. M. G. Al-Saidi, "A new S-Box generation algorithm based on multistability behavior of a plasma perturbation model," *IEEE Access*, vol. 7, pp. 124914–124924, 2019.

[4] N. Nehra, R. B. Patel, and V. K. Bhat, "A framework for distributed dynamic load balancing in heterogeneous cluster," *J. Comput. Sci.*, vol. 3, no. 1, pp. 14–24, Jan. 2007.

[5] N. Kumar, M. Kumar, and R. B. Patel, "Capacity and interference aware link scheduling with channel assignment in wireless mesh networks," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 30–38, Jan. 2011.

[6] N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "A multi-tenant cloud-based DC nano grid for self-sustained smart buildings in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 14–21, Mar. 2017.

[7] A. Farhan and M. Ali, "DB protection system depend on modified hash function," in *Proc. 2nd Int. Conf. Cihan University-Erbil Commun. Eng. Comput. Sci.*, Mar. 2017, pp. 1–121.

[8] Z. Al-Odat and S. Khan, "The sponge structure modulation application to overcome the security breaches for the MD5 and SHA-1 hash functions," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 811–816.

[9] M. Gillela, V. Prenosil, and V. R. Ginjala, "Parallelization of brute-force attack on MD5 hash algorithm on FPGA," in *Proc. 32nd Int. Conf. VLSI Design 18th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2019, pp. 2–7.

[10] K. Khan and W. Goodridge, "A survey of network-based security attacks," *Int. J. Adv. Netw. Appl.*, vol. 10, pp. 3981–3989, Mar. 2019.

[11] V. Chiriaco, A. Franzen, R. Thayil, and X. Zhang, "Finding partial hash collisions by brute force parallel programming," in *Proc. IEEE 37th Sarnoff Symp.*, Sep. 2016, pp. 1–6.

[12] C. Ntantogian, S. Malliaros, and C. Xenakis, "Evaluation of password hashing schemes in open source Web platforms," *Comput. Secur.*, vol. 84, pp. 206–224, Jul. 2019.

[13] A. Bhandari, M. Bhuiyan, and P. W. C. Prasad, "Enhancement of MD5 algorithm for secured Web development," *J. Softw.*, vol. 12, no. 4, pp. 240–252, 2017.

[14] M. Azrour, M. Ouanan, and Y. Farhaoui, "A new secure SIP authentication scheme based on elliptic curve cryptography," in *Proc. Int. Conf. Inf. Technol. Commun. Syst.*, Mar. 2017, pp. 155–170.

[15] A. Chauhan and J. Gupta, "A novel technique of cloud security based on hybrid encryption by blowfish and MD5," in *Proc. 4th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Sep. 2017, pp. 349–355.

[16] M. M. Chauhan, "An implemented of hybrid cryptography using elliptic curve cryptosystem (ECC) and MD5," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Aug. 2016, pp. 1–6.

[17] R. Indrayani, H. A. Nugroho, R. Hidayat, and I. Pratama, "Increasing the security of mp3 steganography using AES encryption and MD5 hash function," in *Proc. 2nd Int. Conf. Sci. Technol.-Comput. (ICST)*, Oct. 2016, pp. 16–19.

[18] S. Ojha and V. Rajput, "AES and MD5 based secure authentication in cloud computing," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Feb. 2017, pp. 856–860.

[19] F. J. B. Talirongan, A. M. Sison, and R. P. Medina, "A modified MD5 algorithm incorporating hirose compression function," in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–6.

[20] E. V. Maliberan, A. M. Sison, and R. P. Medina, "A new approach in expanding the hash size of MD5," *Int. J. Commun. Netw. Inf. Secur.*, vol. 10, no. 2, pp. 374–379, 2018.

[21] W. Zhou, P. Wang, X. Chen, and F. Ye, "An improved Bloom filter in distributed crawler," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 1797–1802.

[22] P. Ora and P. R. Pal, "Data security and integrity in cloud computing based on RSA partial homomorphic and MD5 cryptography," in *Proc. Int. Conf. Comput., Commun. Control (IC)*, Sep. 2015, pp. 1–6.

[23] A. ur Rehman, D. Xiao, A. Kulsoom, M. A. Hashmi, and S. A. Abbas, "Block mode image encryption technique using two-fold operations based on chaos, MD5 and DNA rules," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 9355–9382, Apr. 2019.

[24] B. Charyyev and E. Arslan, "RIVA: Robust integrity verification algorithm for high-speed file transfers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1387–1399, Jun. 2020.

[25] A. Shakya and N. Karna, "Enhancing MD5 hash algorithm using symmetric key encryption," in *Proc. 3rd Int. Conf. Cryptogr., Secur. Privacy ICCSP*, 2019, pp. 18–22.

[26] I. A. Landge and H. Satopay, "Secured IoT through hashing using MD5," in *Proc. 4th Int. Conf. Adv. Electr., Electron., Inf., Commun. Bio-Informatics (AEEICB)*, Feb. 2018, pp. 1–5.

[27] H. Kambo and B. Sinha, "Secure data deduplication mechanism based on rabin CDC and MD5 in cloud computing environment," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 17–21.

[28] A. Kadhim and S. Khalaf, "New approach for security chatting in real time," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 4, no. 3, pp. 1–7, 2015.

[29] F. A. Kadhim and M. H. Emad, "Mouse movement with 3D chaotic logistic maps to generate random numbers," *Diyala J. Pure Sci.*, vol. 13, no. 3, pp. 24–39, Jul. 2017.

[30] B. Kaliski, R. Merkle, and D. Chaum, "The MD5 message-digest algorithm," Tech. Rep., 1992.

[31] L. B. de Guzman, A. M. Sison, and R. P. Medina, "MD5 secured cryptographic hash value," in *Proc. Int. Conf. Mach. Learn. Mach. Intell. MLMI*, 2018, pp. 54–59.

[32] B. Khadem, S. Abedi, and I. Sa-adatyar, "An idea to increase the security of EAP-MD5 protocol against dictionary attack," 2018, *arXiv:1812.01533*. [Online]. Available: http://arxiv.org/abs/1812.01533

[33] C. Mcmahon and X. Zhang, "Modern network security practices: Using rainbow tables to solve organizational issues," in *Proc. IEEE 39th Sarnoff Symp.*, Sep. 2018, pp. 1–5.

AMMAR MOHAMMED ALI received the B.S. degree in computer science from the University of Technology, Baghdad, and the M.S. degree in computer science (computer programming) from Harbin Engineering University, China, in 2012. He is currently pursuing the Ph.D. degree in computer science (data security) from the University of Technology, Baghdad. His research interests include privacy, security, biometric techniques, image processing, and pattern recognition applications.

ALAA KADHIM FARHAN (Member, IEEE) received the bachelor's degree in computer Science and the M.Sc. degree in information security from the Department of computer Sciences, University of Technology, Baghdad, Iraq, in 2003 and 2005, respectively, and the Ph.D. degree in information security from the University of Technology, Baghdad, in 2009. In 2005, he joined the Department of Computer Sciences, University of Technology, as an Academic Staff Member. He is currently a Professor with the Department of Computer Sciences, University of Technology, Baghdad. He has authored numerous technical articles since 2008. His research interests include cryptography, programming languages, chaos theory, and cloud computing.