Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Design and Analysis of New Version of Cryptographic Hash Function Based on Improved Chaotic Maps with Induced DNA Sequences

Salwa M. SeragEldin[1,2], Ahmed A. Abd. El-Latif[3,4,*], Samia A. Chelloug[5,*], Musheer Ahmad[6], Ahmed H. Eldeeb[7], Tamer O. Diab[8], Wageda I. Al sobky[9], Hany Nasry Zaky[10]

[1]Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. 11099, Taif, 21944, Saudi Arabia; s.alsaeed@tu.edu.sa
[2]Department of Electronics and electrical communications Engineering, Faculty of Engineering, Tanta Univer-sity, Tanta, Egypt
[3]EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia; aabdellatif@psu.edu.sa
[4]Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, 32511, Egypt
[5]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia; SAChelloug@pnu.edu.sa
[6]Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India; mahmad9@jmi.ac.in
[7]Electronic and Communication Department, Gezira Higher Institute of Engineering and Technology (EGI). Cairo, Egypt; ahmed.eldeib@gmail.com
[8]Electrical Engineering Department, Benha Faculty of Engineering, Benha University, Banha 13511, Egypt; tamer.almarsafawy@bhit.bu.edu.eg
[9]Basic Engineering Sciences Department, Benha Faculty of Engineering, Benha University, Benha 13511, Egypt; wageda.alsobky@bhit.bu.edu.eg
[10]Department of Mathematics, Military Technical College, Egypt; hanynasry@mtc.edu.eg

Corresponding author: Samia A. Chelloug (SAChelloug@pnu.edu.sa) and Ahmed A. Abd El-Latif (aabdellatif@psu.edu.sa)

ABSTRACT One of the first and most used hashing algorithms in blockchains is SHA-256 so the main aim of this paper is how to increase the security level of the blockchain based on the increasing hashing algorithm security. This paper proposes to present a modified SHA-256 like hash algorithm by exploring the design principles of the two hash schemes (SHA-256 and RIPEMD-160). To retrieve the increased security in the proposed algorithm, we modify the SHA-256 after the public key generation. The proposed modification is based on four different chosen types of chaotic maps and DNA sequences which eventually complicate the association between the original message and hash digest i.e., maximizing the security level, and minimizing its vulnerabilities. The proposed hash function efficiency is practically assessed, analyzed, and compared with the well-known SHA-256 with respect to the main properties of the confusion, diffusion and distribution. The security performance is also analyzed using the analysis of collision, which reveals that the new constructed hash function improves SHA-256 with respect to the security and robustness. From the experimental analyses results, the proposed modified hash scheme found to exhibit the better security performance than many state-of-the art existing hash function schemes.

INDEX TERMS Hash Function; Altered Sine-Logistic based Tent map; Cubic Tent map; DNA sequence.

I. INTRODUCTION

Hash functions are algorithms that convert messages of arbitrary size into message digests of a defined size. These functions must meet certain characteristics, such as one-wayness, meaning that the original message cannot be retrieved from the hash and that two distinct messages cannot yield the same hash result (collision). Moreover, identical input messages should always generate unique output digests. Besides that, hash functions must be efficient and simple to compute [1]. Two types of hash functions exist: keyed and unkeyed. The former types require a message and a secret key, whereas unkeyed hash functions require simply an input message. Such functions make it difficult for adversary to create identical hash values to the system deprived of the secret key. Due to developments in processing power and cryptanalysis tools, however, earlier hash algorithms have grown more vulnerable to attack. Maintaining security requirements and defending against attacks necessitates the development of new, more robust cryptographic algorithms [2]. Hash functions are a key component of security protocols and have several uses, including message authentication codes, encryption tools, digital signatures, and the production of pseudo-random numbers.

The vast majority of conventional hash algorithms, including MD4, MD5, SHA, and SHA-1, rely on elementary logical, arithmetic, and algebraic processes. In practice, these hashing algorithms have been proven to be susceptible to collision attacks, as re-ported in references [3-6]. In addition, the use of linear operations in hash functions potentially renders them vulnerable to cryptographic attacks comparable to those employed in the methods, as explained in the aforementioned sources [1, 7-9]. The hash functions have been crucial in recent years for digital signatures [1], integrity protection, message authentication [16], and even blockchain [10–15]. The basic idea in hash 256 is to transform any variable length of input message to fixed output with 256-bit size. SHA (Secure Hash Algorithm) is the most widely used hash function [16]. SHA-0 generates 160-bit hashes, however when it was released, it was not allowed to be used for its "major weakness". SHA-1 was released in 1995 to solve the security problems found in SHA-0. A switch to SHA-2 was made in 2005 as a result of the security vulnerabilities that had been discovered in SHA-1 [17]. SHA-2 consists of powerful cryptographic algorithms that include many algorithms such as SHA-256, SHA-384, and SHA-512. The SHA-2 hashing algorithm family is the most popular ap-plied hash function due to its security, compatibility, and efficiency [18-22]. SHA-256 is used in many authentication and encryption protocols such as SSL, TLS, IPsec, SSH, and PGP. It is also commonly used in the Bitcoin blockchain, for example, to identify transaction hashes and to execute proof-of-work mining by miners.

Based on deoxyribonucleic acid (DNA) and chaotic maps, a new improvement in hash-256 algorithm is presented. The advantages of using DNA sequence are the high parallelism and information density. Also, it has noticed to have low power consumption [23,24]. Chaos based encryption systems have many advantages such as randomness, initial conditions sensitivity, high security, and large key space, and others make these systems suitable for image encryption techniques [25,26]. The chaotic maps are divided into two categories: continuous and discrete. The discrete chaotic map is divided into 1-dimension and n-dimension. The merging between DNA and chaotic maps will increase uncertainty and unpredictability in the system. The main objectives of this papers can be summarized as follows:

1. In the original Hash-256 implementation, the buffer values (a, b, c, d, e, f, g, and h) remain constant. However, in the modified version of Hash-256, these buffer values are dynamically generated using special chaotic maps combined with DNA sequences. This modification significantly enhances the security level of Hash-256.
2. These modifications have made a positive and effective contribution to enhancing performance and augmenting the security properties of Hash-256. Furthermore, the algorithm has been thoroughly tested, demonstrating exceptional performance.
3. Despite introducing four new alternative maps with varying security specifications, the complexity remains unchanged. This is because the obtained values are directly stored in the buffer values without requiring generation at each stage.

This work is divided into eight sections. A brief explanation of chaotic maps is presented in section 2 and introduction of the DNA sequence and complementary rules are presented in section 3 where the proposed modification part in our Hash-256 are in section 4. Then, the scientific outcomes are revealed in section 5. Finally, the hash attacks and cryptanalytic are presented in section 6 and 7 respectively and the conclusion and description of the model improvements are in section 8.

II. CHAOTIC MAPS

Dynamic systems possess chaotic behavior and consequently require a chaotic system analysis. Bifurcation diagram displays how the behavior of a chaotic system changes with a dynamic control parameter. And, the chaotic maps initial parameter, located in the chaotic areas, is selected using the Lyapunov exponent [27-40]. The Lyapunov exponent signs, λ, provide a qualitative picture of a system's dynamics. For 1D map, there are two cases; λ is positive or negative. Firstly, when λ is -ve, the paths which are slightly separated tend to be convergent and therefore the evolution is not chaotic. Secondly, when λ is +ve, nearby paths are divergent, and the evolution is sensitive to initial conditions and consequently chaotic evolution. The complexity of the map increases as the λ value increases (i.e., it becomes less predictable). In case of the chaotic map f is one-dimensional discrete function, the Lyapunov exponent is as follows:

$$\lambda(x_0, n) = \lim_{n\to\infty} \frac{1}{n} \sum ln\left|f'_\mu(x_i)\right| \qquad (1)$$

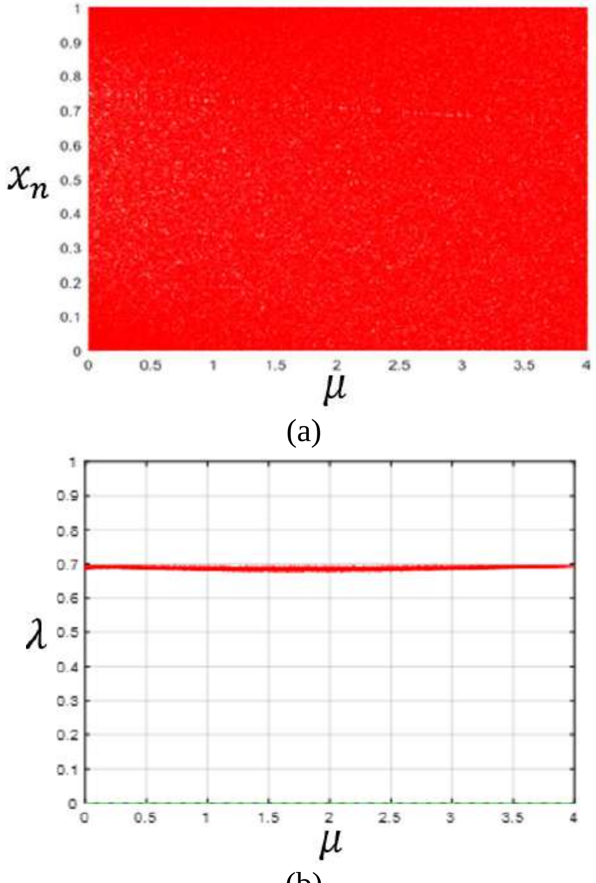
In the previous equation, x_0 is the initial value and n is the iteration. Out of the 150 chaotic Maps available, only 55 were initially utilized. However, upon extensive testing, the majority of them proved to be unsuccessful. Consequently, our focus narrowed down to the four Maps that exhibited promising results during the research. All tests conducted exclusively involved these four Maps, which successfully validated our idea and formed the basis of its implementation in this study. In this section, there are four chaotic maps each of which is one-dimensional map. The Altered Sine-Logistic based Tent map, the Cubic Tent map, 1D-Improved Logistic map, and 1D-Improved Quadratic map. They will be used for the analysis of the proposed system.

A. Altered Sine-Logistic based Tent map ASLT [31]

This map is a hybrid combination of the logistic, sine and tent maps and de-fined mathematically as follows.

$$x_{n+1} = \begin{cases} \frac{4-\mu}{4}\sin(\pi x_n) + \frac{\mu}{2}x_n & x_n < 0.5 \\ (4-\mu)x_n(1-x_n) + \frac{\mu}{2}(1-x_n) & x_n \geq 0.5 \end{cases} \tag{2}$$

Where parameter $\mu \in [0, 4]$ and n is the number of iterations. As shown in Figure 1 the ASLT has excellent chaotic behavior. This could be performed according to the results demonstrated in Figure 1. In the complete range [0, 4] of the parameter settings, the Lyapunov Exponent, μ, of the ASLT is greater than zero.



(a)

(b)

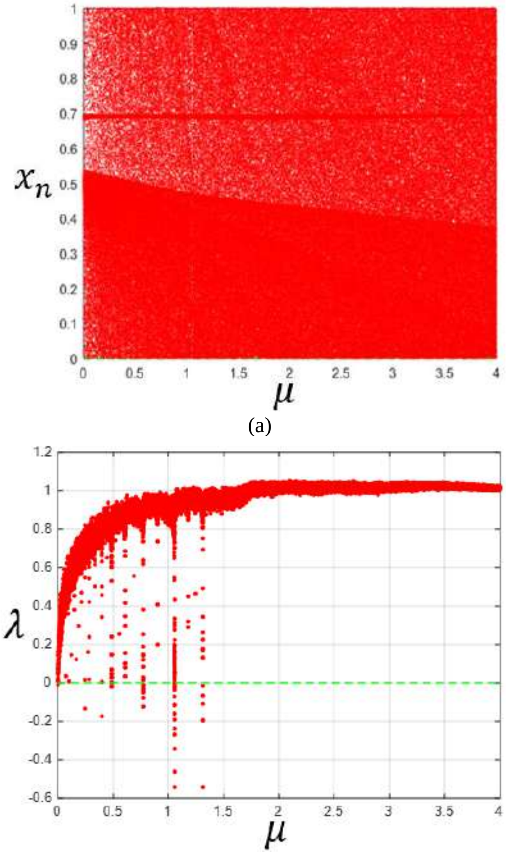
FIGURE 1. (a) Bifurcation diagram (b) Lyapunov plot of ASLT

B. CUBIC TENT MAP

It is considered a hybrid map as it is originated from the Tent map and the Cubic map. The principal purpose of this map is to improve its chaotic nature. It is de-fined mathematically as follows:

$$x_{n+1} = \begin{cases} \mathrm{mod}\left(\left(4-\frac{3}{4}\mu\right)x_n(1-x_{\mathrm{n}}^2) + \frac{\mu}{2}x_n, 1\right) & x_n < 0.5 \\ \mathrm{mod}\left(\left(4-\frac{3}{4}\mu\right)x_n(1-x_{\mathrm{n}}^2) + \frac{\mu}{2}(1-x_n), 1\right) & x_n \geq 0.5 \end{cases} \tag{3}$$

The Cubic Tent map chaotic range is significantly higher than the original chaotic maps, the Cubic, and the Tent maps. The bifurcation diagram and the Lyapunov exponent of the Cubic Tent map are displayed in Figure 2, respectively. Also, in the entire range $\mu \in [0,4]$, the chaotic behavior has some interruptions as shown in the Figure 2.



(a)

(b)

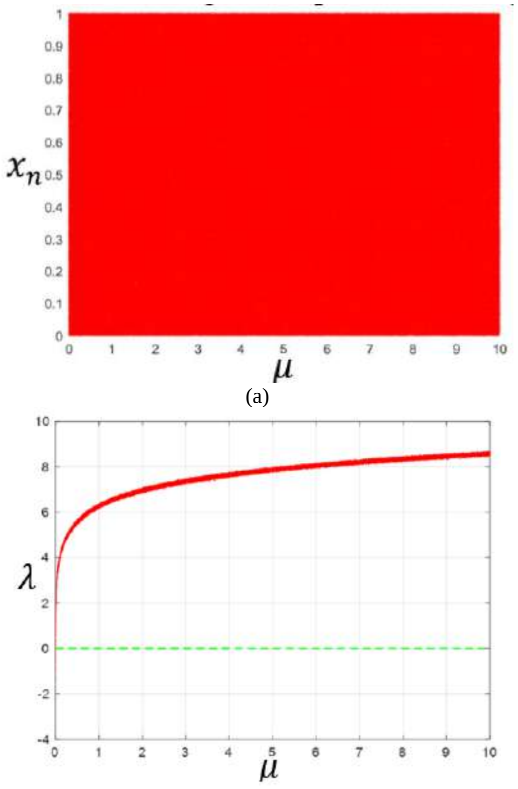
FIGURE 2. (a) Bifurcation diagram (b) Lyapunov plot of Cubic Tent map

C. 1D-Improved Logistic Map ILM

The improved version from the logistic and sine map is presented as follows:

$$\mathrm{x_{n+1}} = \mathrm{mod}(\mu \mathrm{x_n}(1-\mathrm{x_n}) \times 2^{\mathrm{K}}/\sin(\mathrm{x_n})^{\mathrm{R}}, 1) \tag{4}$$

Where μ, K and R represent the control parameters, and x_n represents the initial value of the map within (0, 1]. $K \in$ [2, 26] and $R \in$ [1, 3]. As shown in Figure 3. The chaotic sequences of the ILM have a uniform-distribution within the range [0, 10]. Means, the improved ILM chaotic map covers whole range of parameter range nicely. Hence, the ILM found to possess improved chaotic phenomenon than the conventional chaotic Logistic and the Sine maps.



(a)

(b)

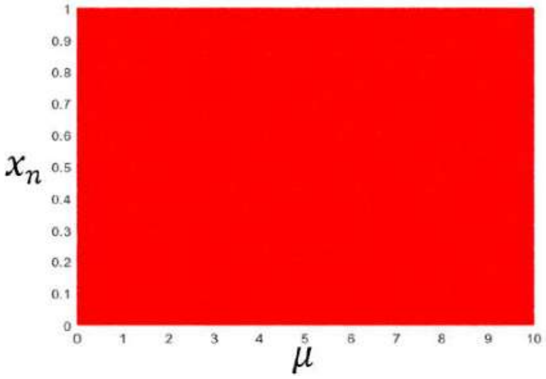
FIGURE 3. (a) Bifurcation diagram (b) Lyapunov plot of ILM map.

D. 1D-Improved Quadratic Map (1D-IQM)

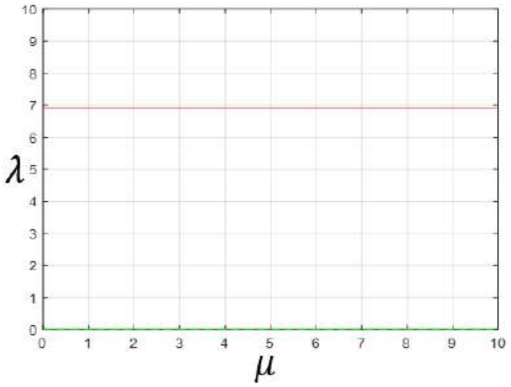
The improved version from the cubic and sine map is presented as follows:

$$x_{n+1} = mod\left(\left((\mu - x_n^2) \times 2^K\right)/\, sin(x_n)^R\,, 1\right) \qquad (5)$$

As shown in Figure 4, the 1D-IQM has excellent chaotic behavior in the whole range of the parameter setting.



(a)



(b)

FIGURE 4. (a) Bifurcation diagram (b) Lyapunov plot of 1D-IQM map.

III. DNA COMPLEMENTARY RULES

At the end of section 2, the output of four different chaotic maps with different critical initial conditions is obtained to increase the randomness and strength of the dynamics of (A, B, C, …, H) values. Moreover, the o/p is expanded by using DNA Sequence [41-44]. Adenine (A), Guanine (G), Cytosine (C) and Thymine (T) are four categories of DNA nitrogen bases and (A, T) and (C, G) are complementary pairs. There are 6 suitable complementary rule which each nucleotide (Ni) DNA complementary rule, The group 3(A → C) (C → T) (T → G) (G → A) and so on up to group 6 as shown in Table 1.

Table 1. *DNA rules*

n_i\ rule	1	2	3	4	5	6
A	T	T	C	C	G	G
T	C	G	G	A	C	A
C	G	A	T	G	A	T
G	A	C	A	T	T	C

IV. NOVEL MODIFIED HASH-256

The complete process is divided into nine different segments, as follows:

4. Convert the given message into an equivalent ASCII stream.
5. Pad the message adds extra bits to your message, such that the length is a 512 multiple which is exactly 64 bits. Note during the addition, the first bit only be one, otherwise be filled with zeroes.
6. Select suitable chaotic maps for both the hash buffers and for the additive constant from 55 discrete chaotic maps.
7. All chaotic maps used has domain for initial value belongs to $[0, 1]$, two symbols are used; x_0 as initial value to get all buffers values and y_0 as initial value for additive constant key (k_0).
8. All output values of x_{n+1} that uses the initial x_0 also have values $\in [0, 1]$. So, multiply all the values

x_{n+1} by 256 then converting to hexadecimal then another transformation to the DNA sequence.

9. Use a suitable complementary rule for each DNA sequence and return back to hexadecimal value put this value in the last 8-bits in buffer A.
10. Repeat again the iterative processes in x_{n+1} the next 8-bits are obtained of buffer hash and keep repeating until 64 bits are reached.
11. The last value of x_{n+1} is the last 8 bits in buffer a repeat the previous steps to obtain the values of all buffers of hash C, D, E, F, G and H.
12. Repeat all the previous steps to get the first additive constant.

The block diagram of hash generation process using chaotic map and DNA sequences is shown in Figure 5.

In this paper, we apply more than 55 different types of chaotic maps to both the hash buffer and additive constants of SHA-256. It is found the most effective maps in improving the performance of SHA-256 are the only four following maps; Altered Si-ne-Logistic based Tent Map ASLT and the Cubic Tent Map, CT and LTS. And consequently, four structures are designed as presented in Table2.

V. MEASURING EFFICIENCY

The benefit of the suggested hash function is disclosed by a performance study that considers hash value distribution, diffusion, confusion, message sensitivity, and collision confrontation. The suggested hash algorithm can work with messages up to 2128 characters long. Three important points will be mentioned about hash functions.

A. Hash value distribution

This paper conducts an experimental simulation to assess the security of the Hash 256 algorithm, which utilizes modified performance to distribute the hash value evenly and randomly across the ciphertext space using the entire plaintext space. As a result, it becomes difficult to discern the relationship between the hash value and the input message. The experimental simulation focuses on analyzing the security of the hash algorithm when applied to the following sentence as an example:

"Cryptography is a Greek word that means "secret writing. However, we use cryptography to transform messages to make them immune and secure against attacks. In the past,

Hashing 1000 completely different messages is used to appraise the scattering of the hash value, The distribution for each test structures based on totaling the rate of hexadecimal digits (0-F) is shown in Figure 6 (a-d) and a total comparison among all structures is shown in Figure 6 (e).

B. Linear complexity

The differences between 0s and 1s in the hash result are measured by the metric of the linear complexity. For different input message, this metric can be evaluated by counting the 0's number in value of the hash function [42]. Figure 7 presents the linear complexity of hash values for 1000 dissimilar message for each design alone in Figure 7 (a, b, c, and d) and by comparing among all designs in Figure 7 (e)

C. Sensitivity analysis

The proposed hash function sensitivity analysis is studied based on the original message change according to the five different conditions or change (C) types in the original message as follows:

C1: The main plaintext message.

C2: Insert 3 after the end of the classes.

C3: Convert the uppercase letter of the letter c in the word "Cryptography" to lower-case.

C4: Convert all letter (o) in sentence to (a).

C5: Change '.' At the end of sentence to '/'.

The hash values and number of the bits changed for all structures are presented in hexadecimal notation as shown in Table 3. The simulation results presented in Figure 8 demonstrates that the suggested hash function structure is highly sensitive to even minor changes in the message's bits.

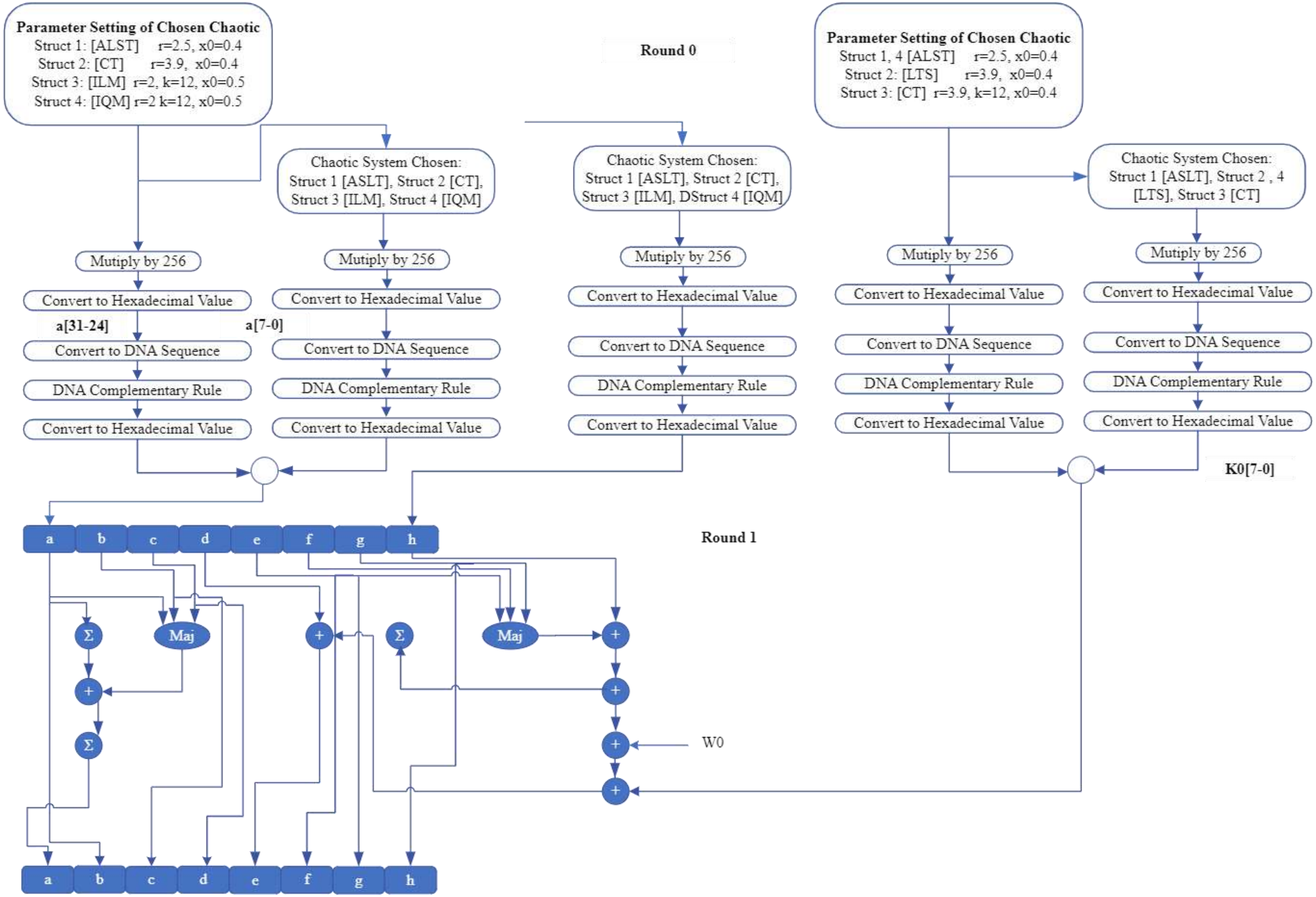
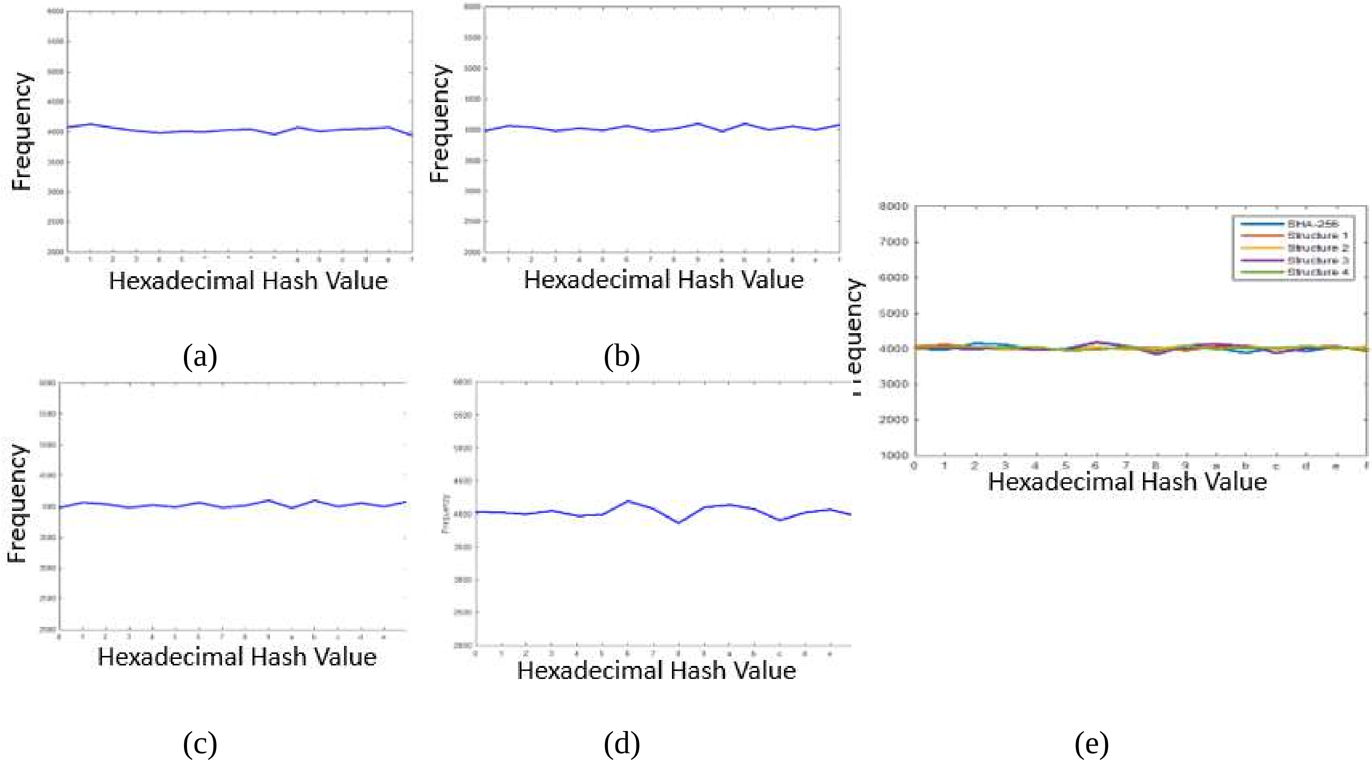

Parameter Setting of Chosen Chaotic
Struct 1: [ALST] r=2.5, x0=0.4
Struct 2: [CT] r=3.9, x0=0.4
Struct 3: [ILM] r=2, k=12, x0=0.5
Struct 4: [IQM] r=2 k=12, x0=0.5
Round 0
Parameter Setting of Chosen Chaotic
Struct 1, 4 [ALST] r=2.5, x0=0.4
Struct 2: [LTS] r=3.9, x0=0.4
Struct 3: [CT] r=3.9, k=12, x0=0.4
Chaotic System Chosen: Struct 1 [ASLT], Struct 2 [CT], Struct 3 [ILM], Struct 4 [IQM]
Chaotic System Chosen: Struct 1 [ASLT], Struct 2 [CT], Struct 3 [ILM], DStruct 4 [IQM]
Chaotic System Chosen: Struct 1 [ASLT], Struct 2 , 4 [LTS], Struct 3 [CT]
Mutiply by 256
Convert to Hexadecimal Value
a[31-24]
a[7-0]
Convert to DNA Sequence
DNA Complementary Rule
Convert to Hexadecimal Value
K0[7-0]
a
b
c
d
e
f
g
h
Round 1
Σ
Maj
+
Σ
Maj
+
W0
a
b
c
d
e
f
g
h

FIGURE 5. Proposed modified Hash-256 algorithm.

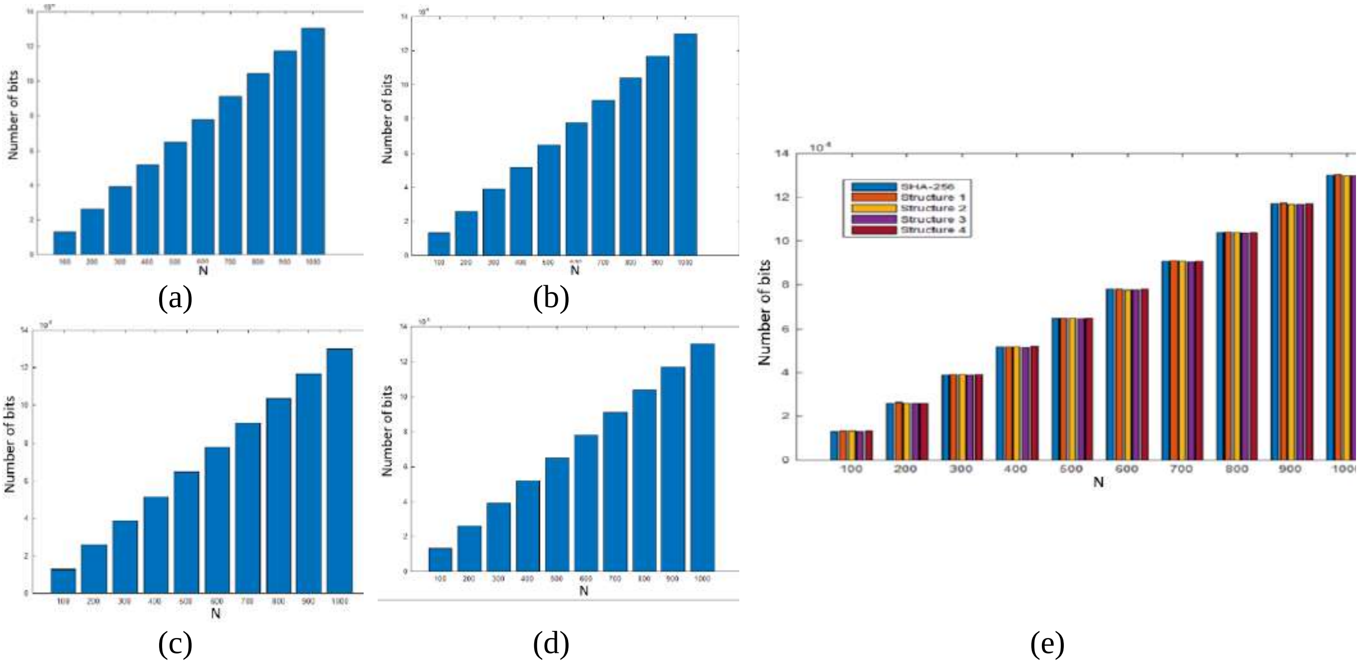
***Table 2*. DNA rules**

Design	**Buffer**	**Adaptive Constant k_t**	**DNA Encoding Rule**	**Complementary Rule**
Design 1	ASLT	CLM	0	G6
	$a = 33ce4703$	$k_1 = 28f5d1f1$		
	$b = 2fc74234$	$k_2 = ee52870f$		
	$c = ff6ac145$	$k_3 = 22a12ada$		
	$d = f8651eff$	$k_4 = a2819985$		
	$e = 107827a2$	$k_{61} = 88a9bfbc$		
	$f = d4c4403c$	$k_{62} = 262ec07b$		
	$g = 8a20ab01$	$k_{63} = a28c9ef2$		
	$h = 0728f7dc$	$k_{64} = 9da327ab$		
Design 2	CT	LTS	5	G1
	$a = 43b23699$	$k_1 = 822d49c8$		
	$b = a8a38b91$	$k_2 = 57a08027$		
	$c = c8286a40$	$k_3 = 9cda4cea$		
	$d = 2f4fe109$	$k_4 = a59cda4f$		
	$e = 8f25710d$	$k_{61} = fa742b62$		
	$f = 25710dae$	$k_{62} = e055abad$		
	$g = 1b2b6967$	$k_{63} = 68d663d8$		
	$h = 6b65fdad$	$k_{64} = b7c153b2$		
Design 3	ILM	CT	5	G3
	$a = c0bec4d6$	$k_1 = 1f4abc5d$		
	$b = 87b78ced$	$k_2 = 20e327f$		
	$c = ee7c9a44$	$k_3 = 6649b37a$		
	$d = 502b52e0$	$k_4 = 7d3ec159$		
	$e = ac2ca425$	$k_{61} = 6b5b0b39$		
	$f = bf14f674$	$k_{62} = 1d4e996b$		
	$g = 51a51f9f$	$k_{63} = f31a5b10$		
	$h = b288bda5$	$k_{64} = 55ad1f4a$		
Design 4	IQM	ASLT	2	G5
	$a = d5c7c83e$	$k_1 = bbed23ab$		
	$b = 62879f17$	$k_2 = 9fe329b2$		
	$c = 20b12cc6$	$k_3 = ff15e820$		
	$d = 96f5b857$	$k_4 = f6108dff$		
	$e = 793b2c04$	$k_{61} = 66547f7e$		
	$f = 8301eb6$	$k_{62} = 919dea37$		
	$g = 686ad30a$	$k_{63} = 596e4df9$		
	$h = 5e51e0d1$	$k_{64} = 4c5b9357$		



(a) (b) (c) (d) (e)

FIGURE 6. Hash value distribution for 1000 message. (a) structure-1 (b) structure-2 (c) structure-3 (d) structure-4(e) combined

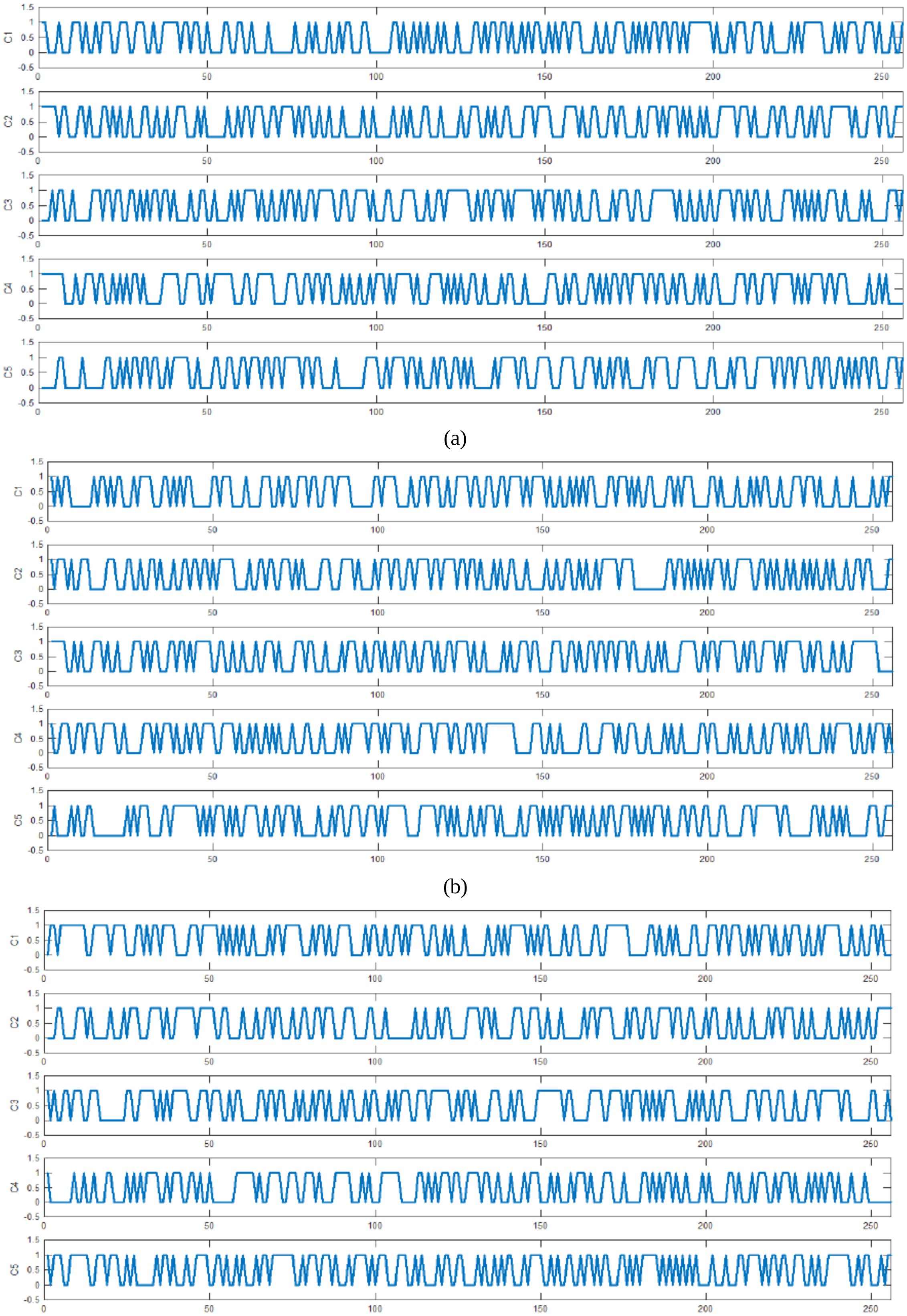


(a) (b) (c) (d) (e)

FIGURE 7. Linear complexity for 1000 messages. (a) structure-1 (b) structure-2 (c) structure-3 (d) structure-4(e) combined

Table 3. The hash values and number of the bits changed for all structures.

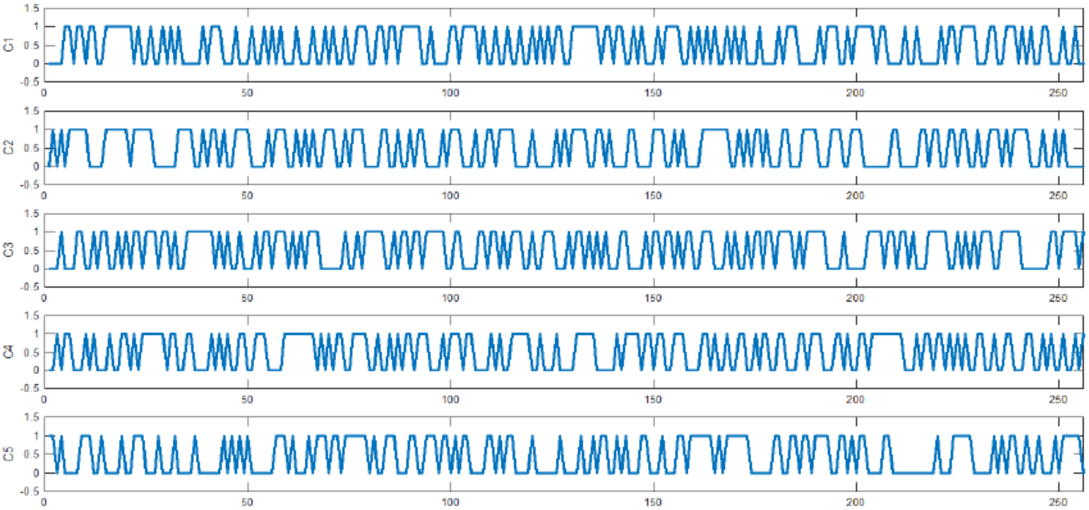
C	*Hash_ASLT_ASLT*	Changed bits
C1	c474b9ce4fb64332101144b180d2a510d6d2b2b7093956dafe8b9c82bf15ce89	--
C2	fb1a352324e2812dcdeb244120946209ca61de1e98d84eba947dd0d8cb3f4767	127
C3	1641db356d0990af4969be67a31e26fe5bafd759185cc7f4927b590eaa72141d	136
C4	fe27655a0f9e5f8cf8736c525d7d1ea990b10396356ded962983b3beaf6e04a0	134
C5	0608195b65f2199bb6fb6100f2f68b3502fb1e3c74b41bc7c30b8fb6e6cd5b1d	129
	Hash_CT_LTS	
C1	ac05ac5e35606e10e59d9de03784dc9eccdebd25586ea61ae0d08ac47742104b	--
C2	ba70719344db5f09b3a838f42ecdededa298824ca5f7801b5574f9a94a944d83	132
C3	f94748758daf9922743b10b934b274de9053b0e69b6ec4a1f2df2cf5f2c49fe0	130
C4	9cdcf20eba5bcf4a5446215f75e8f7b36ff8392061e9c42906c2889d2c2be4ba	136
C5	4298015c37faba9e4dd0459cebf0fb512e82755d496755d1b3183bfb01a540e7	129
	Hash_ILM_ CT	
C1	6ff3ef1ade1df55216f8b4f0b26bcca204afab09867f034a18edcae9725f1268	--
C2	18741171e9fbe61136266e38620122c1ed0739102de1667b3162442dc908a49f	139
C3	a6e700cfabe3b960db14b458b15cefcb438b0ff60f3d755c2a4e0ee623bf6062	129
C4	809230abcb9a407e9e73a1f147e0d5d9ec88b8a2e8e20d5a428345cba2ad1100	136
C5	b9f9dca05a279a516fe5fa2f5c72859852b1794cb58d5f555088df5b394fe58b	129
	Hash_IQM_ASLT	
C1	0db3fa4a82e225444a4e6df21bd249567f6ca2f2a9510b82e5f6120b78cea624	--
C2	57c3f7c0f2d3c2e54e5c32585c4de10579a187287f2b463c7380c1b364ebc4a0	129
C3	119656ed3fa916eae04be5b7cc36b938b54616d6714b6f5f083db47cab9f01bd	131
C4	2c5135fb30a99c3fd59e4adc5d853e443e0bdb386932cc6e34dfe2daa64c6525	128
C5	d0e42711081540e8bb7e9c676b0d610212ca311bf5f819b79ac30011f05648be	134



(a)

(b)

(c)



(d)

FIGURE 8. **Hashes for different conditions (a) structure-1 (b) structure-2 (c) structure-3 (d) Structure-4.**

D. Confusion and diffusion analysis

Claude Shannon established confusion and diffusion [44] as two aspects of the procedure of a secure cipher. The confusion and diffusion of hashing techniques may be computed as follows:

1. For the original input message, determine the hash function ate hash value for an input message.
2. Invert the single bit in the message and determine the hash value again.
3. Compare the difference between hash evaluated in step 1 and step 2.
4. Do N times the steps 1 to 3.

The six analysis is done with different messages lengths L = 256, 512, 1024, 2048 and 10000 as shown in Tables 4

Table 4. Analysis of different messages lengths with four different structures.

Message Length	Structure 1						Structure 2					
	B_{MIN}	B_{MAX}	MEAN	P%	ΔB	ΔP	B_{MIN}	B_{MAX}	MEAN	P%	ΔB	ΔP
256	106	150	127.7	49.87	7.905	3.088	104	161	127.6	49.86	8.721	3.406
512	106	155	127.8	49.92	8.132	3.176	101	161	128.3	50.11	8.486	3.315
1024	106	155	128	50.01	8.075	3.154	101	161	127.9	49.98	8.377	3.272
2048	100	155	128.1	50.02	8.116	3.17	101	161	128.1	50.02	8.214	3.208
10000	97	159	128	50.01	7.921	3.094	100	161	128.1	50.04	8.016	3.131
	Structure 3						Structure 4					
256	104	161	127.6	49.98	8.721	3.406	105	149	128.2	50.08	7.91	3.09
512	101	161	128	50	8.488	3.315	103	151	128.1	50.02	7.822	3.055
1024	101	161	127.9	49.97	8.012	3.272	103	153	127.7	49.89	8.132	3.176
2048	101	161	128.1	50.02	8.006	3.17	98	155	128	49.99	8.139	3.179
10000	100	161	128	50	7.911	3.091	95	153	128	50.02	8.131	3.176

Table 5. Comparative analysis of different hashes.

Hash Scheme	B_{*MIN*}	B_{*MAX*}	Mean	P%	ΔB	ΔP
Proposed (Str-1)	97	159	128	50.01	7.921	3.094
Proposed (Str-2)	100	161	128.1	50.04	8.016	3.131
Proposed (Str-3)	100	161	128	50	7.911	3.091
Proposed (Str-4)	95	153	128	50.02	8.131	3.176
SHA-256	104	154	128	50	7.94	3.1

SHA3-256 [45]	101	153	128.1	50.02	8.01	3.13
SHA2-256 [46]	104	154	128	50	7.94	3.1
Hash in [47]	-	-	63.85	49.88	5.78	4.52
Hash in [48]	-	-	63.85	49.88	5.79	4.52
Hash in [49]	46	80	63.91	49.92	5.58	4.36
Hash in [50]	-	-	63.98	49.98	5.53	4.33
Hash in [51]	-	-	64.01	50.01	5.72	4.47
Hash in [52]	-	-	63.91	49.92	5.58	4.36
Hash in [53]	-	-	63.84	49.88	5.88	4.59
Hash in [54]	-	-	64.14	50.11	5.55	4.33
Hash in [55]	-	-	64.09	50.07	5.48	4.28
Hash in [56]	-	-	63.84	49.88	5.58	4.37
Hash in [57]	45	81	63.88	49.9	5.37	4.2
Hash in [58]	-	-	63.9	49.93	5.64	4.41
Hash in [59]	42	83	63.91	49.92	5.69	4.45
Hash in [60]	-	63.88	49.91	5.75	4.5	-
Hash in [61]	-	-	63.8	49.84	5.75	4.49
Hash in [62]	44	82	64.15	50.11	5.76	4.5
Hash in [63]	-	-	63.56	49.66	7.42	5.8
Hash in [64]	-	-	63.97	49.98	5.84	4.56
Hash in [65]	45	83	64.03	50.02	5.6	4.4
Hash in [66]	43	81	62.84	49.09	5.63	4.4
Hash in [67]	-	-	64.01	50.01	5.61	4.38
Hash in [68]	-	-	64.12	50.09	5.63	4.41
Hash in [69]	43	82	63.89	49.91	5.77	4.5
Hash in [70]	-	-	64.08	50.06	5.72	4.72
SHA3-256 [71]	101	153	128.1	50.02	8.01	3.13
SHA2-256 [72]	104	154	128	50	7.94	3.1
Hash in [73]	-	-	63.85	49.88	5.78	4.52
Hash in [74]	-	-	63.85	49.88	5.79	4.52
Hash in [75]	46	80	63.91	49.92	5.58	4.36
Hash in [76]	-	-	63.98	49.98	5.53	4.33
Hash in [77]	47	83	63.92	49.94	5.62	4.39
Hash in [78]	-	-	64.18	50.14	5.59	4.36
Hash in [79]	-	-	64.07	50.06	5.74	4.48
Hash in [80]	-	-	63.89	49.91	5.64	4.41
Hash in [81]	-	-	63.92	49.94	5.78	4.52
Hash in [82]	-	-	63.4	49.53	7.13	6.35
Hash in [83]	45.6	81.8	63.98	49.98	5.73	4.47
Hash in [84]	-	-	64.43	49.46	5.57	4.51

Where B_{min}, $B_{\max}$, $\bar{B}$, P, and ΔB shown in Table 5 can be defined as follows:

“Minimum inverted bit numbers” is defined by.

$$B_{min} = \text{MIN}\ (B_i)_1^L$$

“Maximum inverted bit numbers” is defined by.

$$B_{\max} = \max\ (B_i)_1^L$$

“Mean inverted bit numbers” is defined by

$$\bar{B} = \sum_1^L \frac{B_i}{L}$$

“Mean changed probability” is defined by

$$P = \frac{\bar{B}}{256} \times 100\%$$

“Standard variance of the inverted bit numbers” is defined by

$$\Delta B = \sqrt{\frac{1}{L-1}\sum_1^L (Bi - \bar{B})^2}$$

“Standard variance of probability” is defined by

$$\Delta P = \sqrt{\frac{1}{L-1}\sum_1^L \left(\frac{B_i}{L} - P\right)^2}$$

VI. HASH ATTACKS

If any of the properties of the hash function can be compromised in any manner, this is known as an attack on the hash function [85]. There are two basic categories of attacks: brute-force assaults and cryptanalytical attacks.

A. Brute force attacks applied on hash functions

The main purpose of the Brute-force attack is to gain unauthorized access to data, systems, or networks. As a result of the successful implementation of such an attack, an attacker can bypass authentication and authorization mechanisms, discover hidden web application resources (directories, files, website sections, etc.) and perform other malicious actions [86].

One method for trying to break a cryptographic primitive is by brute force attacks. Through a thorough "trial-and-error" search in the cypher key space, they are utilized to discover a cypher secret key [87]. These techniques simply focus on the speed at which the cryptographic primitive is computed rather than how it is computed in any way. By assuming that an adversary can compute the output of the primitive for any suitable input, it is possible to analyses their effectiveness (and performance) with relative ease. What's more, the analysis done above only depends on the combinatorial characteristics of the cryptographic primitive in question.

B. Preimage and second preimage attacks

1. **Preimage** [88]. Given a hash function $h: Y \rightarrow Z$, a message $z = h(y)$ computed from a chosen y ∈ Y randomly, where Counter1 … the maximum number of calculating of hash function. Counter2 … the number of the evaluating of the existing preimage values. The structure of the preimage is presented in Figure 9 and the results are shown in Figure 10.

2. **2nd Preimage** can be defined as: chosen y uniformly at random in Y, and computed $h(y)$, find $y_0 \in Y - \{y\}$ such that $h(y_0) = h(y)$.

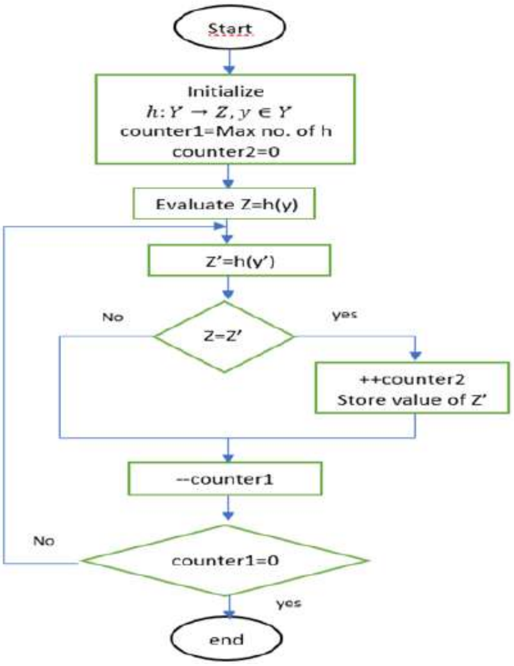
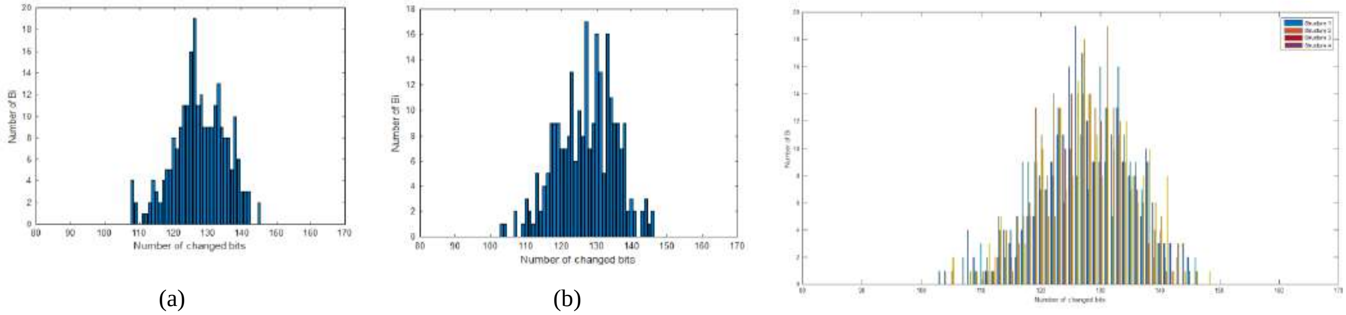
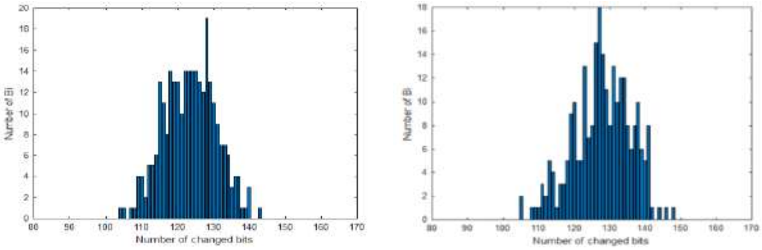


FIGURE 9. Preimage block diagram.



(a) (b)



(c) (d) (e)

FIGURE 10. Statistics and histograms of the average number Bi. (a) structure-1 (b) structure-2 (c) structure-3 (d) structure-4 (e) combined

C. Collision resist attack

Given a hash function $h\colon Y \rightarrow Z$ and the maximum number of calculating of hash function defined in counter1, if 2 messages $y_0 \neq y$ were found after the maximum number stored in counter1 steps such that $h(y_0) = h(y)$ [89]. The number of hits distribution is presented in Figure 11. The number of hits that happened in N tests is defined as follows:

$$WN(\omega) = N \times prob(\omega)$$

$$= N\frac{s!}{\omega!\,(s-\omega)!}(\frac{1}{2^8})^{\omega}(1-\frac{1}{2^8})^{s-\omega} \qquad (6)$$

Where, N … tests numbers and $s = Length(h)/8$, ω is the number of identical bytes.

The theoretical values of $WN(\omega)$ are given in Table 6. An absolute value difference calculated as:

$$\mathrm{D-hash} = \sum_{i=1}^{N}(|\mathrm{t}(\mathrm{a_i}) - \mathrm{t}(\mathrm{b_i})|) \qquad (7)$$

Where, a_i and b_i are the ASCII characters, and *t*(.) converts the ASCII character into its decimal value. The following steps are used to evaluate the absolute value difference.

a) Represent the hash in decimal format from ASCII format.
b) The differences between the two representations were calculated.
c) 1000 messages were repeated, and the resulting comparison values are tabulated in Table. 6.

Table 6. Number of hits for the proposed hash function.

	Hits numbers (ω)					
	0	**1**	**2**	**3**	**4**	**5**
Theoretical value	1806.91	226.75	13.78	0.54	-	1.75 × 10-74
Proposed (Str-1)	1819	219	10	0	0	0
Proposed (Str-2)	1806	222	20	0	0	0
Proposed (Str-3)	1800	240	8	0	0	0
Proposed (Str-4)	1799	235	14	0	0	0
Ref[45]-structure 1	1806	229	13	0	0	0
Ref[45]-structure 2	1787	244	17	0	0	0
Ref[45]-structure 3	1824	213	11	0	0	0
Ref[46]-structure 1	1931	114	3	0	0	0
Ref[46]-structure 2	1929	114	5	0	0	0
Ref[46]-structure 3	1942	106	0	0	0	0

VII. CRYPTANALYTIC ATTACKS

To guarantee the effectiveness and security of cryptographic hash function, the weakness in the hash structure must be covered. The confrontation of a hash function is measured to cryptanalysis for comparing its forte with the brute force attack effort [90]. The hash function is to be ideal if the brute force attack effort must be less than or equal the cryptanalytic hash function. The Meet-in-the-middle is a general crypto-graphic attack that can be used on any block cipher cryptographic techniques.

A. Meet-in-the-middle attack.

This kind of attack always stands against the hash function and is considered as a variation of birthday attack that utilizes the compression function f which becomes invertible to the chaining variable Hi or the message block Xi. And consequently, it allows messages that correspond to certain digest to be created by the attackers. [90]. The Algorithm to calculate the Meet-in-the-middle attacks is considered in the following steps:

1. The original message can be defined as M (M1, M2, M3, ..., Mn)
2. The expected contradicted message can be defined as ME = (ME1, ME2, ME3, ..., MEn)
3. Calculate the hash value for M and ME.
4. Xor (M, ME) to calculate the difference bits number between M and ME.
5. The ideal theoretical value of differing bits ≅128 bits, as in Table 7 and the distribution of the number hits is shown in Figure 11.

Table 7. Different bits in Meet-in-the-middle attack.

Proposed (S-1)	**132**
Proposed (S-2)	138
Proposed (S-3)	139
Proposed (S-4)	131

Finally, the tests done based on the modified hash256 prove the quality and efficiency of the modified algorithm as presented in Table 8.

Table 8. All the requirements for a secure cryptographic hash function

Requirements	Achieved
Variable input message size	Done
Fixed output digested message length=256	Done
Efficiency	Tested
Preimage resistant	Tested
Second preimage resistant	Tested
Collision resistant	Tested
Pseud randomness	Tested

VIII. CONCLUSIONS

In this paper, four different 1-D chaotic maps selected from 55 discrete 1-D chaotic maps which are concluded with the using of DNA sequence to improve the SHA-256 performance. The four most suitable chaotic maps concluded from experimentation are improved logistic, TSS, CLM, LSS and hybrid. The modified SHA-256 algorithm provides better robustness in confusion and diffusion, strong distribution, robustness and sensitivity to defend and stand against all known attacks.

REFERENCES

[1] Ahmad M, Singh S, Khurana S "Cryptographic one-way hash function generation using twelve-terms 4D nonlinear system". Int J Inf Tecnol 13:2295–2303, 2021.

[2] Menezes AJ, Oorschot PCV, Vanstone SA "Handbook of applied cryptography". CRC Press, Boca Raton, 1997.

[3] Wang X, Feng D, Lai X, Yu H (2004) "Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD". Cryptology ePrint Archive, report 2004/199.

[4] Liang J, Lai XJ "Improved collision attack on hash function MD5". J Comput Sci Technol 22(1):79–87, 2007.

[5] Biham E, Chen R, Joux A, Carribault P, Lemuet C, Jalby W "Collisions of SHA-0 and reduced SHA-. Lect Notes Comput Sci 3494:36–57, 2005.

[6] Wang XY, Yin YQ, Yu HB Finding collisions in the full SHA-1. Lect Notes Comput Sci 3621:17–36, 2005.

[7] Xiao D, Liao X, Deng S Chaos based hash function: chaos-based cryptography. Springer, Berlinm 2011.

[8] Ahmad M, Khurana S, Singh S, AlSharari HD "A simple secure hash function scheme using multiple chaotic maps". 3D Res 8(2):1–13., 2017

[9] Wadhwa, S.; Ahmad, M.; Vijay, H. "Chaotic hash function based plain-image dependent block ciphering technique." In Proceedings of the 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Jaipur, India, 21–24 September 2016; pp. 633–637.

[10] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr., no. October, pp. 557–564, 2017, doi: 10.1109/BigDataCongress.2017.85.

[11] J.-P. Delahaye, “Cryptocurrencies and Blockchains,” Inference Int. Rev. Sci., vol. 2, no. 4, 2016, doi: 10.37282/991819.16.38.

[12] J. Wang, G. Liu, Y. Chen, and S. Wang, “Construction and Analysis of SHA-256 Compression Function Based on Chaos S-Box,” IEEE Access, vol. 9, pp. 61768–61777, 2021, doi: 10.1109/ACCESS.2021.3071501.

[13] I. Priyadarshini, Introduction to Blockchain Technology, no. March. 2019.

[14] Basha, H. A. M. A., Mohra, A. S. S., Diab, T. O. M., & El Sobky, W. I. (2022). Efficient image encryption based on new substitution box using DNA coding and bent function. IEEE Access, 10, 66409-66429.

[15] Wageda I. El Sobky Sherif Hamdy Gomaa and A. Y.Hassan, “A-Survey-of-Blockchain-from-the-Viewpoints-of-Applications,-Challenges-and-Chances,” Int. J. Sci. Eng., vol. 12, no. 4, p. 11, 2021.

[16] S. H. Standard, “FIPS Pub 180-1,” Natl. Inst. Stand. Technol., vol. 17, no. 180, p. 15, 1995.

[17] E. Edition, O. Systems, S. Edition, and B. D. Communications, the William Stallings Books on Computer Data and Computer Communications , Eighth Edition, vol. 139, no. 3. 2011.

[18] A. F. S. Ibrahim, “New secure solutions for privacy and access control in Health Information Exchange,” ProQuest Diss. Theses, p. 171, 2016, [Online]. Available: https://search.proquest.com/docview/1819468453?accountid=13460%0Ahttp://zp2yn2et6f.search.serialssolutions.com/directLink?&atitle=New+secure+solutions+for+privacy+and+access+control+in+Health+Information+Exchange&author=Ibrahim%2C+Ahmed+Fouad+Shedeed&issn.

[19] J. Park and J. H. Park, “applied sciences A Lightweight Hash-Based Blockchain Architecture for Industrial IoT,” 2019, doi: 10.3390/app9183740.

[20] R. Martino and A. Cilardo, “Designing a SHA-256 processor for blockchain-based IoT applications,” Internet of Things, vol. 11, p. 100254, Sep. 2020, doi: 10.1016/J.IOT.2020.100254.

[21] Hala Saeed, Hossam E.Ahmed, Tamer O.Diab, Hossam L.Zayed, Hany Nasry Zaky, and Wageda I.Elsobky, “Evaluation of the Most Suitable Hyperchaotic Map in S-Box Design Used in Image Encryption,” International Journal of Multidisciplinary Research

and Publications (IJMRAP), Volume 5, Issue 4, pp. 176-182, 2022.,

[22] K. Ljupco and L. Shiguo, “Chaos-Based Cryptography,” vol. 354, 2011, doi: 10.1007/978-3-642-20542-2.

[23] Y. Liu, Y. Luo, S. Song, L. Cao, J. Liu, and J. Harkin, “Counteracting Dynamical Degradation of Digital Chaotic Chebyshev Map via Perturbation,” http://dx.doi.org/10.1142/S021812741750033X, vol. 27, no. 3, Apr. 2017, doi: 10.1142/S021812741750033X.

[24] Alawida, M.; Samsudin, A.; Alajarmeh, N.; Teh, J.S.; Ahmad, M.; Alshoura,W.H. A Novel Hash Function Based on a Chaotic Sponge and DNA Sequence. IEEE Access 2021, 9, 17882–17897.

[25] R. Martino and A. Cilardo, “SHA-2 Acceleration Meeting the Needs of Emerging Applications: A Comparative Survey,” IEEE Access, vol. 8, pp. 28415–28436, 2020, doi: 10.1109/ACCESS.2020.2972265.

[26] A. Mohammed Ali and A. Kadhim Farhan, “A novel improvement with an effective expansion to enhance the MD5 hash function for verification of a secure E-Document,” IEEE Access, vol. 8, pp. 80290–80304, 2020, doi: 10.1109/ACCESS.2020.2989050.

[27] M. Alawida, A. Samsudin, and J. Sen Teh, “Enhanced digital chaotic maps based on bit reversal with applications in random bit generators,” Inf. Sci. (Ny)., vol. 512, no. November, pp. 1155–1169, 2020, doi: 10.1016/j.ins.2019.10.055.

[28] A. Ahmed Mohammed and A. O. Ibadi, “A Proposed Non Feistel Block Cipher Algorithm,” Qalaai Zanist Sci. J., vol. 2, no. 2, pp. 64–71, 2017, doi: 10.25212/lfu.qzj.2.2.08.

[29] Nasry, H., Abdallah, A. A., Farhan, A. K., Ahmed, H. E., & El Sobky, W. I. (2022, August). Multi Chaotic System to Generate Novel S-Box for Image Encryption. In Journal of Physics: Conference Series (Vol. 2304, No. 1, p. 012007).

[30] A. T. Sadiq, A. K. Farhan, and S. A. Hassan, “A proposal to improve RC4 algorithm based on hybrid chaotic maps,” J. Adv. Comput. Sci. Technol. Res, vol. 6, no. 4, pp. 74–81, 2016.

[31] Aouissaoui, Ichraf, et al. "Improved One-Dimensional Piecewise Chaotic Maps for Information Security." J. Commun. 17.1 (2022): 11-16. cubic tent map

[32] Alawida, M.; Teh, J.S.; Oyinloye, D.P.; Alshoura, W.H.; Ahmad, M.; Alkhawaldeh, R.S. A New Hash Function Based on Chaotic Maps and Deterministic Finite State Automata. IEEE Access 2020, 8, 113163–113174.

[33] Wageda Ibrahim Alsobky ,Abdelkader Esmail ,Ashraf S. Mohra, Ayman Abdelaziem “Design and Implementation of Advanced Encryption Standard by New Substitution Box in Galois Field (2^8)” International Journal of Telecommunications, IJT’2022, Vol.02, Issue 01

[34]. Nada E. El-Meligy , Tamer O. Diab , Ashraf S. Mohra , Ashraf Y. Hassan and Wageda I. El-Sobky"A Novel Dynamic Mathematical Model Applied in Hash Function Based on DNA Algorithm and Chaotic Maps". Mathematics 2022, 10, 1333. https://doi.org/10.3390/ math10081333

[35] H. Liu, A. Kadir, and J. Liu, ‘‘Keyed hash function using hyper chaotic system with time-varying parameters perturbation,’’ IEEE Access, vol. 7,pp. 37211–37219, 2019.

[36] A. Kumar, A. Fatima, and N. K. Nishchal, ‘‘An optical Hash function construction based on equal modulus decomposition for authentication verification,’’ Opt. Commun., vol. 428, pp. 7–14, Dec. 2018.

[37] A. Kanso, H. Yahyaoui, and M. Almulla, ‘‘Keyed hash function based on a chaotic map,’’ Inf. Sci., vol. 186, no. 1, pp. 249–264, 2012.

[38] Designing Two Secure Keyed Hash Functions Based on Sponge Construction and the Chaotic Neural Network.

[39] Abdoun, N.; El Assad, S.; Deforges, O.; Assaf, R.; Khalil, M. Design and security analysis of two robust keyed hash functions based on chaotic neural networks. J. Ambient Intell. Humaniz. Comput. 2020, 11, 2137–2161.

[40] M. Todorova, B. Stoyanov, K. Szczypiorski, and K. Kordov, ‘‘SHAH?: Hash function based on irregularly,’’ INTL J. Electron. Telecommun., vol. 64, no. 4, pp. 457–465, Oct. 2018.

[41] Maolood, Abeer Tariq, Alaa Kadhim Farhan, Wageda I. El-Sobky, Hany Nasry Zaky, Hossam L. Zayed, Hossam E. Ahmed, and Tamer O. Diab. "Fast Novel Efficient S-Boxes with Expanded DNA Codes." Security and Communication Networks 2023 (2023).

[42] Zheng, Jieyu, and LingFeng Liu. "Novel image encryption by combining dynamic DNA sequence encryption and the improved 2D logistic sine map." IET Image Processing 14.11 (2020): 2310-2320.

[43] El-Meligy, Nada E., Tamer O. Diab, Ashraf S. Mohra, Ashraf Y. Hassan, and Wageda I. El-Sobky. "A novel dynamic mathematical model applied in hash function based on dna algorithm and chaotic maps." Mathematics 10, no. 8 (2022): 1333.

[44] Shannon, C. E.(1949), “The Mathematical theory of communication,” In: TheMathematical Theory of Communication, edited by C. E. Shannon and W. Weaver,University of Illinois Press, Urbana. *(6) (PDF) Generalization of Shannon's information theory*. Available from:

[45] Dworkin, M.J. SHA-3. Standard: Permutation-Based Hash and Extendable-Output Functions; PUB FIPS 202; Information Technology Laboratory National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015, doi:10.6028/NIST.FIPS.202.

[46] Standard, S.H.; FIPS, P. 180-2. August 2002, 1, 72.

[47] Xiao, D.; Liao, X.; Deng, S. One-way Hash function construction based on the chaotic map with changeable-parameter. Chaos Solitons Fractals 2005, 24, 65–71.

[48] Lian, S.; Sun, J.; Wang, Z. Secure hash function based on neural network. Neurocomputing 2006, 69, 2346–2350.

[49] Zhang, J.; Wang, X.; Zhang, W. Chaotic keyed hash function based on feedforward–feedback nonlinear digital filter. Phys. Lett. A 2007, 362, 439–448.

[50] Wang, Y.; Liao, X.; Xiao, D.; Wong, K.W. One-way hash function construction based on 2D coupled map lattices. Inf. Sci. 2008, 178, 1391–1406.

[51] Xiao, D.; Liao, X.; Wang, Y. Parallel keyed hash function construction based on chaotic neural network. Neurocomputing 2009, 72, 2288–2296.

[52] Deng, S.; Xiao, D.; Li, Y.; Peng, W. A novel combined cryptographic and hash algorithm based on chaotic control character. Commun. Nonlinear Sci. Numer. Simul. 2009, 14, 3889–3900.

[53]Deng, S.; Li, Y.; Xiao, D. Analysis and improvement of a chaos-based Hash function construction. Commun. Nonlinear Sci. Numer. Simul. 2010, 15, 1338–1347.

[54] Yang, H.; Wong, K.W.; Liao, X.; Wang, Y.; Yang, D. One-way hash function construction based on chaotic map network. Chaos Solitons Fractals 2009, 41, 2566–2574.

[55] Xiao, D.; Liao, X.; Wang, Y. Improving the security of a parallel keyed hash function based on chaotic maps.Phys. Lett. A 2009, 373, 4346–4353.

[56] Amin, M.; Faragallah, O.S.; El-Latif, A.A.A. Chaos-based hash function (CBHF) for cryptographic applications. Chaos Solitons Fractals 2009, 42, 767–772.

[57] Li, Y.; Xiao, D.; Deng, S. Secure hash function based on chaotic tent map with changeable parameter. High Technol. Lett 2012, 18, 7–12.

[58] Wang, Y.; Du, M.; Yang, D.; Yang, H. One-way hash function construction based on iterating a chaotic map. In Proceedings of the International Conference on Computational Intelligence and Security Workshops 2007,Heilongjiang, China, 15–19 December 2007; pp. 791–794.

[59] Akhavan, A.; Samsudin, A.; Akhshani, A. Hash function based on piecewise nonlinear chaotic map. Chaos Solitons Fractals 2009, 42, 1046–1053.

[60] Huang, Z. A more secure parallel keyed hash function based on chaotic neural network. Commun. Nonlinear Sci. Numer. Simul. 2011, 16, 3245–3256.

[61] Li, Y.; Deng, S.; Xiao, D. A novel Hash algorithm construction based on chaotic neural network.Neural Comput. Appl. 2011, 20, 133–141.

[62] Wang, Y.; Wong, K.W.; Xiao, D. Parallel hash function construction based on coupled map lattices. Commun. Nonlinear Sci. Numer. Simul. 2011, 16, 2810–2821.

[63] Li, Y.; Xiao, D.; Deng, S.; Han, Q.; Zhou, G. Parallel Hash function construction based on chaotic maps with changeable parameters. Neural Comput. Appl. 2011, 20, 1305–1312.

[64] Li, Y.; Xiao, D.; Deng, S.; Zhou, G. Improvement and performance analysis of a novel hash function based on chaotic neural network. Neural Comput. Appl. 2013, 22, 391–402.

[65] He, B.; Lei, P.; Pu, Q.; Liu, Z. A method for designing hash function based on chaotic neural network.In Proceedings of the International Workshop on Cloud Computing and Information

Security (CCIS),Shanhai, China, 9–11 November 2013; pp. 229-233.

[66] Jiteurtragool, N.; Ketthong, P.; Wannaboon, C.; San-Um, W. A topologically simple keyed hash function based on circular chaotic sinusoidal map network. In Proceedings of the 2013 15th International Conference on Advanced Communications Technology (ICACT), Pyeong Chang, South Korea, 27–30 January 2013; pp. 1089–1094. Entropy 2020, 22, 1012 34 of 34

[67] Teh, J.S.; Samsudin, A.; Akhavan, A. Parallel chaotic hash function based on the shuffle-exchange network.Nonlinear Dyn. 2015, 81, 1067–1079.

[68] Chenaghlu, M.A.; Jamali, S.; Khasmakhi, N.N. A novel keyed parallel hashing scheme based on a new chaotic system. Chaos Solitons Fractals 2016, 87, 216–225.

[69] Akhavan, A.; Samsudin, A.; Akhshani, A. A novel parallel hash function based on 3D chaotic map. EURASIP J. Adv. Signal Process. 2013, 2013, 126.

[70] Nouri, M.; Khezeli, A.; Ramezani, A.; Ebrahimi, A. A dynamic chaotic hash function based upon circle chord methods. In Proceedings of the 6th International Symposium on Telecommunications (IST), Tehran, Iran,6–8 November 2012; pp. 1044–1049.

[71] Dworkin, M.J. SHA-3. Standard: Permutation-Based Hash and Extendable-Output Functions; PUB FIPS 202; Information Technology Laboratory National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015, doi:10.6028/NIST.FIPS.202

[72] Standard, S.H.; FIPS, P. 180-2. August 2002, 1, 72

[73] Xiao, D.; Liao, X.; Deng, S. One-way Hash function construction based on the chaotic map with changeable-parameter. Chaos Solitons Fractals 2005, 24, 65–71.

[74] Lian, S.; Sun, J.; Wang, Z. Secure hash function based on neural network. Neurocomputing 2006, 69, 2346–2350.

[75] Zhang, J.; Wang, X.; Zhang, W. Chaotic keyed hash function based on feedforward–feedback nonlinear digital filter. Phys. Lett. A 2007, 362, 439–448

[76] Wang, Y.; Liao, X.; Xiao, D.; Wong, K.W. One-way hash function construction based on 2D coupled map lattices. Inf. Sci. 2008, 178, 1391–1406.

[77] Xiao, D.; Liao, X.; Deng, S. Parallel keyed hash function construction based on chaotic maps. Phys. Lett. A 2008, 372, 4682–4688.

[78] Yu-Ling, L.; Ming-Hui, D. One-way hash function construction based on the spatiotemporal chaotic system. Chin. Phys. B 2012, 21, 060503.

[79] Li, Y.; Xiao, D.; Li, H.; Deng, S. Parallel chaotic Hash function construction based on cellular neural network. Neural Comput. Appl. 2012, 21, 1563–1573.

[80] Li, Y.; Xiao, D.; Deng, S. Keyed hash function based on a dynamic lookup table of functions. Inf. Sci. 2012 214, 56–75.

[81] Ren, H.; Wang, Y.; Xie, Q.; Yang, H. A novel method for one-way hash function construction based on spatiotemporal chaos. Chaos Solitons Fractals 2009, 42, 2014–2022.

[82] Guo, X.F.; Zhang, J.S. Keyed one-way Hash function construction based on the chaotic dynamic S-Box. Acta Phys. Sin. 2006, 55, 4442–4449.

[83] Yu, H.; Lu, Y.F.; Yang, X.; Zhu, Z.L. One-way hash function construction based on chaotic coupled map network. In Proceedings of the 2011 Fourth International Workshop on Chaos-Fractals Theories and Applications, Hangzhou, China, 19–22 October 2011; pp. 193–197.

[84] Zhang, H.; Wang, X.F.; Li, Z.H.; Liu, D.H. One way hash function construction based on spatiotemporal chaos. Acta Phys. Sin. 2005, 54, 4006-4011.

[85] R. Sobti and G. Geetha, "Cryptographic Hash functions - a review," IJCSI Int. J. Comput. Sci. Issues, vol. 9, no. 2, pp. 461–479, 2012.

[86] M. Todorova, B. Stoyanov, K. Szczypiorski, and K. Kordov, "SHAH: Hash function based on irregularly decimated chaotic map," Int. J. Electron. Telecommun., vol. 64, no. 4, pp. 457–465, 2018, doi: 10.24425/123546.

[89] J. Zhang, X. Wang, and W. Zhang, "Chaotic keyed hash function based on feedforward-feedback nonlinear digital filter," Phys. Lett. Sect. A Gen. At. Solid State Phys., vol. 362, no. 5–6, pp. 439–448, 2007, doi: 10.1016/j.physleta.2006.10.052.

[90] Y. Li, D. Xiao, S. Deng, Q. Han, and G. Zhou, "Parallel Hash function construction based on chaotic maps with changeable parameters," Neural Comput. Appl., vol. 20, no. 8, pp. 1305–1312, 2011, doi: 10.1007/s00521-011-0543-4.