DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

A Lightweight Hash Function Based on Enhanced Chaotic Map Algorithm(Keccak)

Yusra Ahmed Ghareeb[1], Ekhlas Khalaf Gbashi[2]
[1,2]Department of Computer Science, University of Technology, Baghdad, Iraq
[1]cs.19.11@grad.uotechnology.edu.iq, [2]110026@uotechnology.edu.iq

Abstract— *Cryptography is a security strategy that prevents disclosure of the information while it is transit, in storage, or both. There are a variety of methods for maintaining data security, including utilizing light weight speed algorithms for encryption and parameter validation. Many algorithms have originated in the area of information protection, helping to assure the validity of the information generated. These include the following algorithms: SHA-1, SHA-2, SHA-3, AES, RC5, RSA, and more. In order to secure the legitimacy of the information and monitoring data , the speed of encryption and authentication must be critical. . Due to the necessity of fast and secure algorithms, these features are required. In this work, modification of the SHA-3 algorithm by introducing a new function called (the keccak function), which has an extremely quick execution time and a high level of security, also versatile cryptographic function. This change is implemented via the 2D chaotic system, which is geared towards generating random values for constants for the SHA3 algorithm. These constants values are generated by the SHA3 algorithm, and so are random and unguessable by the intrude. Statistical tests conducted by the National Institute of Standards and Technology (NIST) effectively outperformed the randomness of a proposed approach .The proposed algorithm shows lower execution time compared to previous studies, which is 0.041616sec for 1MB.*

key words— *Cryptography, Hash function, keccak Function, SHA-3, Chaotic system*

I. INTRODUCTION

A deterministic process (or function) that is input from every data block and returns a fixed (cryptographic) result is known as a hash cryptographic function. These security and privacy features were first implemented for specific reasons [1]. it was employed The hash algorithm is commonly used to verify data integrity in cryptography. Using the hash value is exceedingly difficult if the input message is unknown. Even if no successful attacks have been made against a weakened variation, an assault may succeed and result in the downfall of the entire hashing method [2]. MD5, SHA1, SHA2, and SHA3 are four of the widely used cryptographic hash methods. As a result, NIST selected a new cryptographic hash algorithm in 2007, initiating a new one in the process (SHA-3). The Keccak algorithm won a five-round examination in 2012. Keccak refers to a family of sponge functions that can be used as stream ciphers, hash functions, Message Authentication Codes (MACs), and pseudo-random number generators. In general, these functions work by mapping a variable-length input to a variable-length output using a fixed-length transformation and a padding rule for the input. The SHA-3 standard for cryptographic hash functions now includes a subset of the Keccak sponge functions[3]. Important for implementing information security and to ensure data integrity in digital transactions, SHA-3 is a popular cryptographic algorithm. For over half of the output bits, any change to the input message will trigger an

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

avalanche effect [4]. Many modern designs and designs that have been modified for SHA-3 are now being used in SHA-3 research.

This study is organized in the following way. Section 2 is providing the overview of cryptography hash function. Section 3 is providing the standard secure hash function algorithm version3. Section 4 providing describe keccak function. Section 5 explain about chaotic system. Section 6 Shows the proposed algorithm. Section 7 It shows the results and discussion of the proposed algorithm with a comparison with previous studies.

There are many researchers who have made modern designs as well as modifying some designs for the SHA-3 algorithm that works in a specific environment to provide the best results.

1. Magnus and Ricardo [5] proposed a SHA-3 unfolded structure, which they felt would serve as a benchmark in relation to state-of-the-art approaches. A pipeline structure that consists of a pipeline register connecting the round function and the step mapping has been modified from a basic structure with an internal pipeline register.

2. Ming et al. [6] proposed the new and simplified round constant (RC) generator, which uses pipelining, unrolling, and sub pipelining for improved hashing functionality. This has so generated a total of five different SHA-3 implementations.

3. Haider et al. [7] offered an alternative to the SHA-3 algorithm using a new technique called SPECK that replaces the Keccak function with another very fast algorithm. Another important feature of the expanded logistic system is that it will be utilized to generate the initial values used by the SHA3 algorithm. The values generated by the system will be of no utility to an intruder and so will remain unknown to him.

4. Hayder et al. [8] advocated using a high-speed, lower-memory, and lower-processing algorithm, known as Salsa20, as the fundamental feature of their newly proposed adaptation of the Secure Hash Algorithm 3 (SHA-3). Also, expanding the logistic technique would result in uncertain and not recognizable initial values for the Secure Hash Algorithm 3 (SHA-3) algorithm.

In the present paper, the lightweight sha-3 algorithm is used in providing security is a modification on sha-3, such as round constants, rotation offsets and also some modification on the keccak function.

II. CRYPTOGRAPHIC HASH FUNCTION(CHF)

A hash function (HF) is a cryptographic process that transforms an input message of variable length to a fixed message known as a hash message or message digest. Using the capabilities of hash functions, we may build authentication systems or random generators that assist with data integrity and digital signatures [9].

As a tool to help achieve a wide range of safety purposes, including authenticity, numerical signatures, generation of pseudo numbers, digital steganography, and digital time stamping, one of the essential cryptographic hash functions (CHF) is an essential component.[10]

Data integrity and authentication has been preserved using numerous hashing methods that have been employed throughout the data's lifecycle. For this reason, agree on a form of encrypted data between the two parties, which may be delivered and received between many computers, using one of the stable hash algorithms as a method of ensuring confidentiality [11].

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

III. STANDARD SECURE HASH ALGORITHM-3 (SHA-3)

Keccak, a popular hashing algorithm, was selected as the new default method for SHA-3 hashing. the new Secure Hash Algorithm-3 has been created by Keccak It comprises several SHA-3 versions, including the ones with a hash size of 224 bits, 512 bits, 384 bits, and 256 bits [12]. It comprises various rounds, which allows for more logical operations on each round. In order to absorb the input first, and then squeeze out the desired output [13], the sponge has been constructed. Initializing, absorbing, and squeezing the input message "M" results in three Sponge functions: Initializing, absorbing, and squeezing. Sponge functions illustrated in *Fig. 1* and in the output side of the padding module are designated as Z and begin with the character "Z"[14].

• To initialize the input matrix, the zero input step is performed, and to generate 1600 bits of arbitrary input blocks, the input padding is carried out.

• The XOR method is done on the input matrix and then all 24 rounds are run.

• In order to meet the required output length, the input matrix is truncated during the squeezing process.

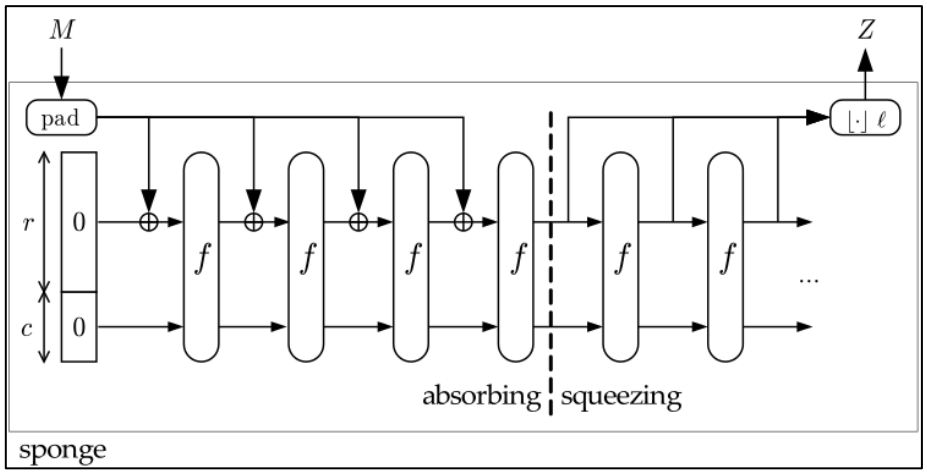

FIG. 1. SPONGE CONSTRUCTION [13].

IV. KECCAK FUNCTION

The National Institute of Standards and Technology launched the process for selecting a new secure cryptographic hash algorithm, in order to increase the security performance of hash functions. In order to meet different security needs such as digital signatures and message authentication codes, SHA-3 was developed to be the most secure [15]. In every cycle, each step is made up of five independent phases, such as Theta (Θ), Rho (Ƥ), Pi (π), Chi (X) and Iota (ℓ). These steps are outlined in the formulas mentioned at the bottom of the page, shown as equations (1), (2), (3), and (4) , A,B,C,D

Which the Theta (Θ) steps are:

$$C[x] = A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus A[x,3] \oplus A[x,4]$$

$$D[x] = C[x \oplus 1] \oplus rot(C[x + 1], 1)$$

$$A[x,y] = A[x,y] \oplus D[x] \qquad (1)$$

$$0 \leq x, y \leq 4$$

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

Rho (P) and Pi (π) steps:

$$B[y, 2x + 3y] = rot(A[x, y], r[x, y]) \quad (2)$$

$0 \leq x, y \leq 4$

Chi (χ) step:

$$A[x, y] = B[x, y] \oplus ((\_B[x + 1, y] \wedge B[x + 2, y]) \quad (3)$$

$$0 \leq x, y \leq 4$$

Iota (ι) step:

$$A[0; 0] = A[0; 0] \oplus RC[i] \quad (4)$$

The element $A[x, y]$ denotes the specific word in the state array, while $B[x, y]$, $C[x]$ and $D[x]$signify the intermediate variables.

V. CHAOTIC SYSTEMS

In contrast to the laws of nature that are known with certainty, chaos theory is based on nonlinear and probabilistic behavior [16]. Because of this, minor changes in the beginning values or control settings will result in dramatic fluctuations in the chaotic outputs [17]. The term "chaos theory" was originally coined to explain the seeming unpredictability of complex systems. The chaotic systems feature chaotic and random output signals that are very unpredictable and random in nature [18]. According to chaos theory, unpredictable output is a must for every type of method [19], which means that chaos theory may be applied to encrypt images, random number generators, hash functions, block ciphers, stream ciphers, steganography, and watermarking [20]. Table I in [21] displays two different classifications of chaotic maps: continuous time and discrete time chaotic maps.

TABLE I. THESE CHAOTIC SYSTEMS [21] CAN BE COMPARED

Chaotic Map Name	A Domain	Dimension
Logistic map	A Discrete	1
Piecewise linear chaotic	A Discrete	1
Tent map	A Discrete	1
Gaussian map	A Discrete	1
Standard map	A Discrete	2
Hénon map	A Discrete	2
Cat map	A Discrete	2
Baker map	A Discrete	2
Lorenz system	A Continuous	3
Rössler attractor	A Continuous	3
Chen system	A Continuous	3
Jerk equation	A Con0tinuous	3

In Hénon map, basic nonlinear dynamical equations can give birth to extremely complex, seemingly chaotic behavior. Two is the number of values the beginning conditions can have. Michel Hénon (a geographer from France) released the map in 1947. According to mathematics, the Hénon map is defined as [22]:

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

$X(n+1) = 1 - a * x(n)^2 + y(n)$…… .(5)

$Y(n+1) = b * X(n)$…….... (6)

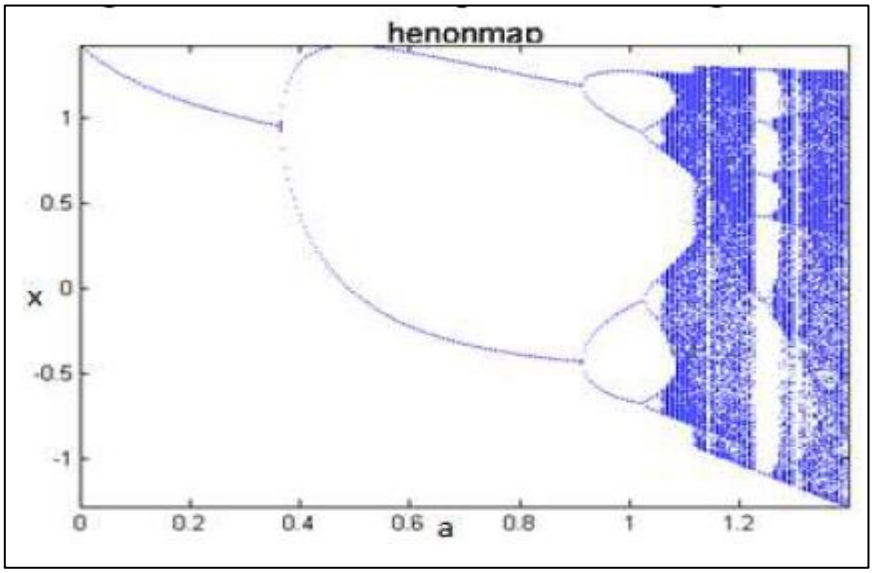

FIG. 2. BIFURCATION DIAGRAM OF THE HÉNON MAP [22].

A "cat map" is a map in mathematics whose matrix has a determinant of 1, and so has integer entries for its inverse. The cat map is described as follows [23]:

X=2x+y mod 1…… (7)

Y=x+y mod 1 ……..…….....…. (8)

VI. PROPOSED SHA-3 ALGORITHM

In this new effort, an entirely new method of generating the SHA-3 algorithm has been developed to make the algorithm work more rapidly and effectively. To make it more suitable for the different types of data that are entered, make the round constants and rotation offsets float values rather than constant values. Chaos is implemented in the generation of values to ensure greater security. Using a Xor operation, the Hénon map (1,2,3) and the cat map (4) are implemented in the development of the proposed Lightweight SHA-3. The SHA-3 algorithm's design It is made up of multiple stages: the creation of all essential hash tools was dependent on each other, implementing a mixture of chaotic systems, and this system incorporates Algorithm 1. Rounds are reduced from 24 to 10, and the left shift rotation is implemented instead of the Xor. These adjustments are made to reduce the time it takes to compute SHA-3 while still providing a solid security layer to protect against most attacks. Sha-3 presented the following modified algorithm in the following:

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

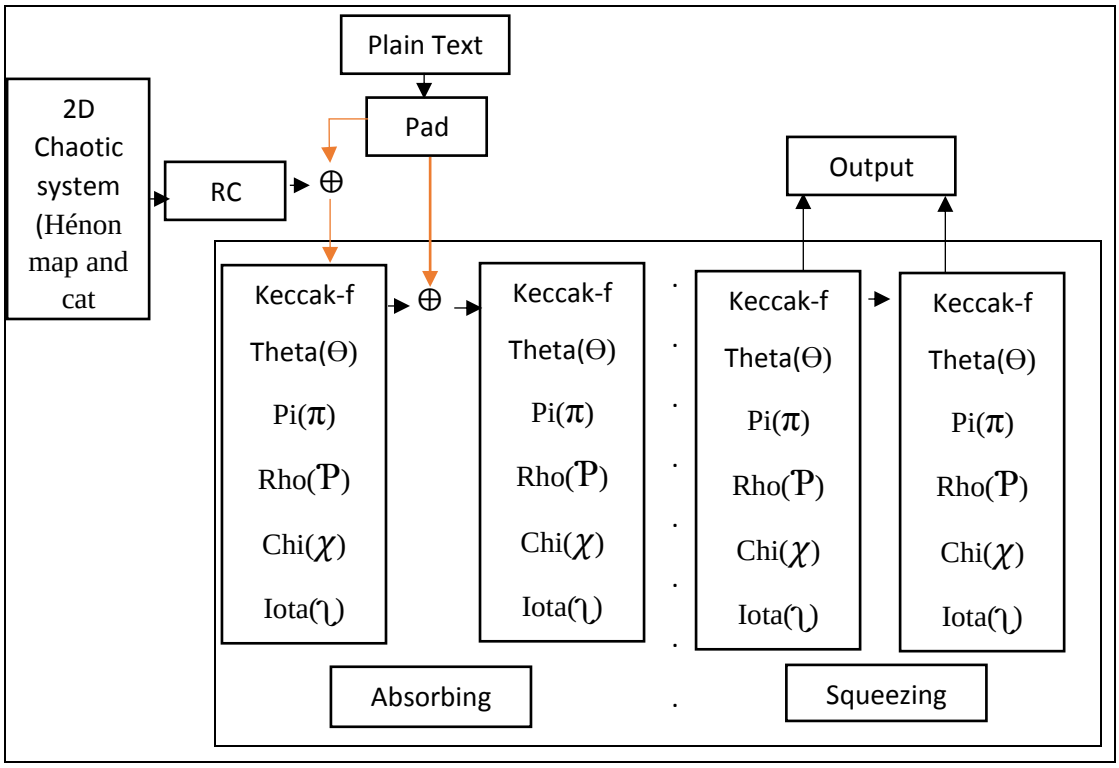

FIG. 3. BLOCK DIAGRAM OF PROPOSED SHA-3.

Algorithm 1: Key generation of System
Input: 2D chaotic map initial conditions (X0, Y0, L, B)
Output: Round constants (10), rotation offsets (5*5).
Begin
Step1: Read initial values.
Step2: Generate chaotic floating values using Hénon map and cat map
Step3: Take the value after the sorter of floating numbers.
Step4: Xor operation between the resulting values of equations
X←X Cat Xor X Hénon , Y←Y Cat Xor Y Hénon
Key ← X Xor Y to construct Round constants, rotation offsets.
End

Received 17/September/2021; Accepted 17/November/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

Algorithm 2: The Proposed SHA-3 Algorithm

Input: Plaintext, chaotic map initial value
Output: hash value (512 bit)
Begin
Step1: Pad the input using the padding function and denote the result as P. We will pad the input to the length where we get the length of P1600 bit
Step2: break the P into n consecutive pieces. Denote these Strings as P (0), P (1), …, P (N-1).
Step3: Xor between pi and Round constants that generated from the 2D chaotic map. Set of string S
In the absorbing phase.
Step4: for each P(i)
4.1: Add number of '0' bits to P(i) so that the final length of P(i) is b.
4.2: XOR (P(i), S).
4.3: Apply block permutation function to step 4.2. The result is S(new).
Moving to the squeezing phase:
Step5: Initialize an empty string Z.
Step6: While length (Z< d) , d the length of output string:
6.1: Append the first of S to Z.
6.2: If (Z< d) bits then apply f to S. The result is now S new.
Step7: Truncate Z to d bits.
In permeation Block
Step8: For i=0 to 9 // i here to represent the amount of rounds it takes to fight.
8.1: Using 1600 bits as input length S.
8.2: Apply the five steps are θ (theta), ρ (rho), π (pi), χ (chi) and ι (iota) // permeation Block value
Step9: Save hash value and repeat steps on a new block of plaintext.
End

A. The new round constant construction

In this suggestion, a new method for creating a suitable Round constant was introduced, and it generates a message of bewilderment when utilized in the proposed LSHA-3 algorithm. Thus, it offers much of the security in LSHA-3. Based on the improved 2D chaotic map equation, the proposed round constants are constructed. For generating the constants, we use a round-robin approach. In the first stage, the floating numbers are generated by the cat map, and in the next stage, the Hénon and Xor between cat map and Hénon map values are utilized. mathematical equations and formulas creating enhanced, fresh, and secure results. The Iota equation always provides a new set of unique keys each time the round is applied. Therefore, this also cuts down on the amount of arithmetic needed in Iota (ι), as the constant multiplier from 24 to 10 is sufficient The setup presented in Algorithm 1 will be utilized to create the round constants for the LSHA-3 algorithm.

B. The new rotation offsets construction

Distributing slides among lanes rotates each lane to a different offset. To ensure good level of diffusion, a rotation compensation plan must be created each time the initial values are entered. This new rotation offsets are created using Hénon and cat maps. During the Xor procedure, previously discussed in section 4.1 and algorithm 1. the algorithmically generated sites are generated if a duplicate takes 25 positions. Formulas that are used to generate change values when run have their initial values applied.

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

C. The Left Shift rotate

The third stage in this proposal is to use the left shift rotate operation to instead of operation Xor in equation chi (χ). In left shift rotate, the bits that fall off at left end are put back at right end. At this stage, the operations are left Shift rotate instead of Xor to increase the randomness and complexity of the Sha-3 algorithm and make it more security. Table II represents an example of an instead operation.

TABLE II. EXAMPLE OF INSTEAD OPERATION

Xor operation	**Left Shift rotate**
0110 =6	0110=6
0011=3	0011=3
Xor(0110,0011)=0101=5	0011<<6=1100=12

D. The summation process

Fourth, apply the summation process for the goal of handling the diffusion process by causing and amplifying the load development and spread. Instead of using the Xor method, use a summation on equation theta (Θ). Using this alteration, each bit of the message is equally and visibly distributed to hide it.

VII. THE RESULTS & DISCSSION

To increase the difficulty of implementing the proposed SHA-3 algorithm, the developers put the algorithm within a chaotic system. Results of the long-term experiment in which SHA-3 was used are displayed in Table III as well as the results of the NIST experiments, which appear in Table IV.

Table III presents the results of a comparison between the original SHA-3 algorithm, the reference, and the modified SHA-3 algorithm. The modified SHA-3 method was initially used to validate data information at a fast speed. Thus, the improved SHA-3 algorithm in terms of execution speed has been achieved.

TABLE III. IMPLEMENTATION TIME

File Size	Standard Sha-3 (sec)	SSHA [6] (sec)	Proposed SHA-3 (sec)
1KB	0.023	0.014	0.000127
10KB	0.284	0.17	0.000198
100KB	1.14	0.92	0.001703
1MB	11.66	6.49	0.041616

The proposed SHA-3 algorithm is faster than others (0.001703 sec for hash 100KB), while the original SHA-3 algorithm takes 11.66 sec and reference in [6] takes 6.46 sec to hash the same text file size, in proposed SHA-3 0.000198 sec for hash 10KB while the original SHA-3 algorithm takes 0.284 sec and reference in [6] takes 0.17 sec.

Table IV indicates that if data is modified with different modifications in numbers and letters, the modified algorithm adds even more confusion to the clear text, while the

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

probability of verifying the data after being applied to it is higher. This change to the SHA-3 algorithm will alter the overall output.

TABLE IV. COMPARISON OF NIST'S TESTING FINDINGS

Category of results used in the NIST statistical tests	original SHA-3	Ref[7]	Proposed SHA-3
Frequency (Mono bit) test	0.342	0.510	0.423
Gives a test run	0.261	0.587	0.947
FFT test	0.011	0.651	0.358
Measuring the frequency of an action inside a block of a test.	0.929	0.498	0.988
The longest running test	0.002	0.614	0.969
self-test	0.508	0.595	0.717
Maurrer's (Generalized) Universal Statistical Test	0.052	0.411	0.131
single template test overlap	0.005	0.990	0.189
Line of best fit	0.546	0.600	0.642
template overlap test that doesn't overlap	0.991	0.779	0.999
randomly-delimited exploration test	0.366	0.801	0.503
random exploratory outings	0.522	0.649	0.642
Cumulative sums test	0.652	0.518	0.854

The updated algorithm was tested, and the original algorithm can be implemented based on the requirements, leading to acceptability of the results from the proposed algorithm. Using this randomization process offers additional security when working with larger files, as their reliance on data heterogeneity is stronger. In order to achieve overall system reliability.

VIII. CONCLUSION AND FUTURE WORK

Based on the findings of the modified SHA-3 method, we may conclude that this algorithm is fast and strong. for the sake of simplicity, we will be replacing the SHA-3 algorithm and the processes that are based on it with the modified version that replaces the Xor operation with the Shift operation. More specifically, in equation chi, instead of the Xor operation, the summation is performed in equation theta, changing the number of round, the process of constructing rotation tables and round constants, depending on the 2D optimized chaotic system. These modifications enabled the SHA-3 algorithm to finish processing within only 0.000127 milliseconds per 1 kilobyte (as opposed to the original SHA-3 algorithm, which took 0.023 milliseconds). This leads to a higher level of security and a less complex algorithm when compared to the original SHA-3 algorithm.
The work can be developed and/or expanded with the following suggestions: -

1. Incorporating an additional chaotic system into the encryption process will help to boost security.
2. To improve security, an artificial intelligence approach is added to processes that occur within of the rounds.

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.2.5

3. A tool should be created to analyze the algorithm randomness signature and identify all of the algorithm's possible weaknesses, including any problems with randomness.

REFERENCES

[1] X. Wang,D. Feng,X.Lai and H. Yu "Collisions for hash functions md4, md5, haval-128 and ripemd," *IACR*, August 2004.

[2] M. J. Schiereck "Sha-3 Standard: Permutation-Based Hash and Extendable-Output Functions" Federal Inf. Process. Stds. (NIST FIPS)-202, Gaithersburg, MD, USA, 2015.

[3] "Keccak Team", *Sponge.noekeon.org*, 2021. [Online]. Available: http://sponge.noekeon.org/. [Accessed: 19- Oct- 2021].

[4] S.B.Sarmadi, M. M.Kermani and A Aziz "E_cient and Concurrent Reliable Realization of the Secure Cryptographic SHA-3 Algorithm" *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 33, no. 7, 2014.

[5] S.Magnus, and R.Chaves "Efficient FPGA implementation of the SHA-3 hash function " *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*,p. 86-91, 2017.

[6] M. M. Wong, J. Haj-Yahya, S. Sau and A. Chattopadhyay"A New High Throughput and Area Efficient SHA-3 Implementation" *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5,2018.

[7] H. K. Hoomod , J.R.Naif and I. S. Ahmed "Modify Speck-SHA3 (SSHA) for data integrity in WoT networking based on 4-D chaotic system" *Periodicals of Engineering and Natural Sciences (PEN)* ,vol. 8, no. 4, pp 2379-2388, 2020.

[8] H. Najm,R. Hassan and H. K. Hoomod"Data Authentication for Web of Things (WoT) by Using Modified Secure Hash Algorithm-3 (SHA-3) and Salsa20 Algorithm"*Turkish Journal of Computer and Mathematics Education (TURCOMAT)*,vol.12 ,no.10,pp 2541-2551,2021.

[9]R. Chaves et al., "Secure hashing: SHA-1, SHA-2, and SHA-3," Circuits Syst. Secur. Priv., pp. 81–107, 2017, doi: 10.1201/b19499.

[10] R. Sobti and G. Geetha "Cryptographic Hash functions - a review" *International Journal of Computer Science Issues (IJCSI)*,vol. 9,no.2, pp 461,2012.

[11] A. K. Farhan and M. Ali, "Database protection system depend on modified hash function," *in Conference of Cihan University-Erbil on Communication Engineering and Computer Science*, 2017, p. 84.

[12] L. Henzen, P. Gendotti, P. Guillet, E. Pargaetzi, M. Zoller, and F. K. Gürkaynak "Developing a hardware evaluation method for SHA-3 candidates " *in International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 248–263,2010.

[13] J. Sharma and D. Koppad "Low power and pipelined secure hashing algorithm-3(SHA-3)" *IEEE Annu. India Conf. INDICON 2016*, vol. 3, pp. 0–4, 2017, doi: 10.1109/INDICON.2016.7839059.

[14]J. James, R. Karthika, and R. Nandakumar "Design & Characterization of SHA 3- 256 Bit IP Core "*Procedia Technology* , vol.24,pp.918-924,2016

[15]G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, Keccak. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 313–314.

[16] A. K. Farhan and H. Emad " Mouse Movement with 3D Chaotic Logistic Maps to Generate Random Numbers" *Diyala Journal for Pure Sciences* ,vol. 13, no.3,2017.

[17] H. N. Kadhim,N. Al-Saidi, M. Said and A. Kilicman "A new hyperchaotic map and its application for image encryption" ,*Springer*,*The European Physical Journal Plus*,vol.133,no.1 ,pp.1-14,2018.

[18] E. K.Gbashi and A. K. Farhan "Chaotic System and DNA Computing operations for Image Encryption Based on Pixels Shuffling", *Al-Qadisiyah Journal Of Pure Science*,vol.26,no.2 ,pp.. 1–14,2021.

[19] G. Jakimoski and L. Kocarev"Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps",*IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS*,vol. 48, no. 2 ,pp.163-169,2001.

[20]O. Jallouli "Chaos-based security under real-time and energy constraints for the Internet of Things" ,Thesis, Signal and Image processing, Universite de Nantes, English, ,2017.

[21] A. K. Farhan and R. S.Ali " Enhancement AES based on 3D Chaos Theory and DNA Operations Addition" *Karbala International Journal of Modern Science*, vol.5 ,no.2,2019.

[22] Dr. H. B.Abdul Wahab and S. I. Mahdi "Speech Encryption Based on Wavelet Transformation and Chaotic Map"*Eng. &Tech.Journal*,vol.34,no.5,2016

[23] N. A.Abbas "Image encryption based on Independent Component Analysis and Arnold's Cat Map", *Egyptian Informatics Journal*,vol.17,no.1,pp.139-146,2016.

Received 17/September/2021; Accepted 17/November/2021
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*