Journal of Al-Qadisiyah for Computer Science and Mathematics Vol.11(3) 2019 , pp Comp.69–85

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Multikey Image Encryption Algorithm Based on a High-Complexity Hyperchaotic System

Ahmed Sh. Ahmed [a] ***, Hussein A. Salah*** [b] ***, Jalal Q.Jameel*** [c]

[a] *Department of Basic Sciences, College of Nursing, University of Baghdad, Baghdad, Iraq. Email: ahmedshihabinfo@conursing.uobaghdad.edu.iq*

[b] *Department of Computer Systems, Technical Institute- Suwaira, Middle technical university, Baghdad, Iraq. Email: husseinalali6@gmail.com*

[c] *College of Medicine, Al-Mustansiriyah University, Baghdad ,Iraq. Email: jalalalqaisy1@gmail.com*

ARTICLE INFO

Article history:
Received: 05 /05/2019
Rrevised form: 16 /06/2019
Accepted : 16 /07/2019
Available online: 01 /09/2019

Keywords:

Hyperchaotic, Image encryption, Chaotic encryption.

ABSTRACT

In life, a chaotic system has many applications in different fields, including physics, biology, communication, and cryptography. In this study, a new hyperchaotic system is introduced. This hyperchaotic system is a two-dimensional system that is based on three maps-namely, logistic, iterative chaotic, and Henon maps. The dynamics of this system are investigated using maximal Lyapunov exponents, bifurcation diagrams, phase portraits, basin of attraction, and complexity via entropy. This system shows highly complicated dynamics. On the basis of the proposed system, a new algorithm for image encryption is also introduced. Confusion and diffusion can be achieved with this algorithm, which are fundamental demands. The stochastic behavior of this system is used to reinforce the security of the encrypted image. The image is divided into four parts, each of which uses a different random key established by the proposed chaotic system. The security of this cryptosystem is validated on the basis of key security parameters and common attacks.

MSC.

1. Introduction

A few years ago, the communications industry was established as an integral part of our life because of its importance in the transmission of information. In the processes of data transmission, the confidentiality of the information being sent and received is an important issue. However, traditional methods of encryption, such as the Advanced Encryption Standard, Data Encryption Standard, and Rivest Cipher 4 (RC4), are largely considered unsafe for securing images; their distinguishing features include bulk data capacity, high redundancy, and strong adjacent pixel correlation [1]. Many researchers have therefore introduced several image encryption algorithms, such as compressive sensing [2–4], wavelet transmission [5,6], DNA coding [7], affine transformation [8], neural network [9], blowfish algorithm [10], RC4 [11], and chaotic mapping [1,12–16]. Virtually, a chaotic map exhibits better results versus the other methods because of its complexity, mixing, and randomness, which are similar to the characteristics of the diffusion and confusion principles of cryptography.

Corresponding author Ahmed Sh. Ahmed

Email addresses: ahmedshihabinfo@conursing.uobaghdad.edu.iq

Communicated by Dr. Mustafa Jawad Radif

Chaotic maps used for the purpose of image encryption are more effective because of their high security as well as fast speed. There are two main types of chaotic maps: one-dimensional (1D) chaotic maps, which are dependent on one variable, and high-dimensional (HD) chaotic maps. Image encryption based on 1D chaotic maps is considered risky [1], whereas algorithms based on HD chaotic maps are considered more safe and suitable for encryption. HD systems should satisfy two requirements: first, the system must be discrete, or the properties of the dynamical system must be discretized, and, second, the system must be as simple as possible so as to increase the encryption speed. Many researchers in recent years have developed algorithms on the basis of chaotic maps. In 2010, Liu et al. [17] introduced a new algorithm for image encryption on the basis of robust chaotic maps. In 2011, Ye et al. [18] proposed a new image algorithm based on a chaos system with an efficient permutation–diffusion mechanism. In 2013, Sheng et al. [19] presented a novel bit-level image encryption protocol developed on the basis of hyperchaotic systems. Separately, Wang et al. [20] developed an algorithm by employing dynamic S-boxes and two 1D chaotic maps, while Xiaoling et al. [21] showcased a new algorithm for image encryption on the basis of hyperchaos and deoxyribonucleic acid (DNA) sequences. In 2015, Rasul et al. [22] proposed a novel algorithm for image encryption on the basis of a hybrid model of DNA and cellular automata. In the same year, Wang et al. [23] introduced an algorithm for image encryption by utilizing the chaotic shuffling diffusion method. Hua et al. [15] presented a new algorithm for image encryption on the basis of a two-dimensional (2D) sine logistic modulation map, while Wang et al. [24] used DNA sequencing operations and chaotic systems for image encryption and Koppu et al. [25] employed hybrid chaotic magic transformation for image encryption. In addition, Wenhao et al. [13] relied on a new 2D system based on sine mapping and iterative chaotic map to design a novel bit-level image encryption algorithm. Furthermore, in 2016, Liu et al. [26] proposed a novel algorithm that used a logistic chaotic map to encrypt images. Li et al. [27] offered a new algorithm that incorporated pixel-level and bit-level permutations to encrypt images on the basis of hyperchaotic maps. Hayder et al. [16] suggested a new hyperchaotic map based on three maps called 2D-SHAM and offered a new image encryption algorithm based on the proposed system. Finally, in 2018, Ca et al. [1] adopted new 2D hyperchaotic maps and used them as a basis for a new algorithm for image encryption called 2D logistic iterative chaotic map with infinite collapse (ICMIC) cascade mapping (2D-LICM).

To vanquish the weaknesses of the other encryption algorithm, this article proposes a new two-dimensional (2D) hyperchaotic system which is derived from three maps-namely, logistic, iterative chaotic, and Henon maps. Performance analysis of this system shows highly complicated dynamics, hyperchaotic properties and better ergodicity. Used this chaotic system security applications to generate a novel image encryption algorithm. this algorithm mainly depending on divided the plaintext and generate four different key (multi-key) generated from the hyperchaotic system to increase complicate and decrease the time. The encryption process mainly depending on row encryption and column encryption. The plain-image is divided into blocks to generate four different key (multi-key) based on the proposed hyperchaotic system to increase the complexity and reduce the computation time. Finally, to show the efficiency of the encryption image, some performance analysis tests are performed such as; histogram, NPCR, correlation, and entropy. The proposed image encryption algorithm is compared with some other encryption algorithms. The efficiency and analysis of security of this algorithm showed a reasonable improvement over them. The present paper is organized as follows: we first introduce a 2D version of the hyperchaotic map in section 2. Section 3 presents a performance evaluation of the aforementioned novel 2D hyperchaotic map. Section 4 proposes simulation results of the algorithm of image encryption depending on four keys generated from the system of the new 2D hyperchaotic map, and Section 5 includes the conclusion details of this paper.

2. 2D Novel Hyperchaotic Map

2.1. Definition of Existing Chaotic Maps

Henon map [28] is 2D discrete time system defined as

$$x_{i+1} = 1 - ax^2 + y_i$$
$$y_{i+1} = by_i \tag{1}$$

where $a \in [0, 1.4]$, $b = 0.3$ are system parameters.

A logistic map [29] is a 1D discrete time system defined as

$$x_{i+1} = rx_i(1 - x_i) \tag{2}$$

where r is a system parameter, $r \epsilon (0, +\infty)$.

An ICMIC [30] is 1D similar to the logistic map, mathematically, and is defined as

$$x_{i+1} = \sin({}^{c}/_{x_i}) \tag{3}$$

where c is a system parameter, c ϵ $(0, +\infty)$.

In a dynamical system, the bifurcation diagram refers to a system phenomenon that introduces a new behavior as variable parameters. The bifurcation of preliminary chaotic maps (i.e., Henon map, logistics map, and ICMIC) is illustrated in Figure 1.

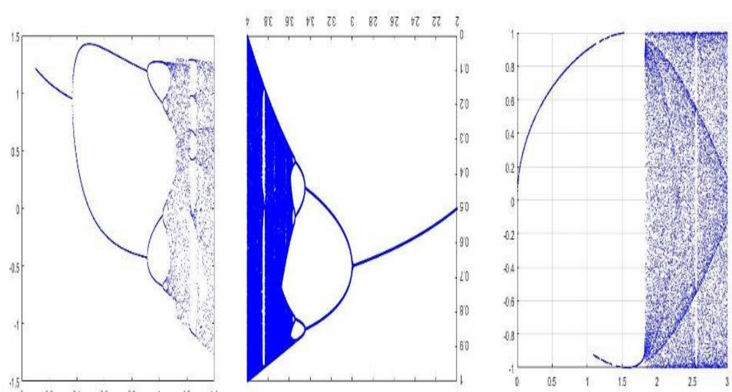

Figure 1: Bifurcation of a (a) Henon map, (b) logistics map, and (c) ICMIC.

2.2. Definition of A Novel 2D Hyperchaotic Map

Based on the aforementioned maps, we present a novel 2D hyperchaotic map, mathematically defined as:

$$x_{i+1} = 2\sin(2y_i(1 - y_i)) + \sin(21/(2x_i + (k/2\pi)\sin(x_i)))$$
$$y_{i+1} = 21x^3 + \sin(21/(r + (ky_i + 3)y_i(1 - y_i))) \tag{4}$$

Where k and r are system parameters and $k \in (0{,}100)$ and $r \in (0{,}10)$. From the realized Lyapunov exponents, the system can be said to be a new 2D hyperchaotic map.

2.3. Presentation and Performance of The Novel 2D Hyperchaotic Map

In this section, we evaluate the performance of chaotic systems (e.g., phase diagram, Lyapunov exponents, and permutation entropy). We also compare the novel 2D hyperchaotic map with other chaotic maps, such as a 2D-LICM and a 2D sine ICMIC modulation map (2D-SIMM).

2.3.1 Phase Diagram

The dynamical system trajectory is a series of values that show the movement track of the output of a system. We set the parameters k and r to ensure that the maximum range spreads in the phase space. With these settings, we can safeguard the perfect property of the ergodic dynamical system and conform to the structure of the new hyperchaotic map. Attractors of 2D-LICM [1], 2D-SIMM [15], and our chaotic maps are shown in Figure 2. The diagrams show that the distribution of the new hyperchaotic map is greater than those of 2D-LICM and 2D-SIMM. Hence, the randomness and ergodicity properties of the former are more superior than those of the two latter.

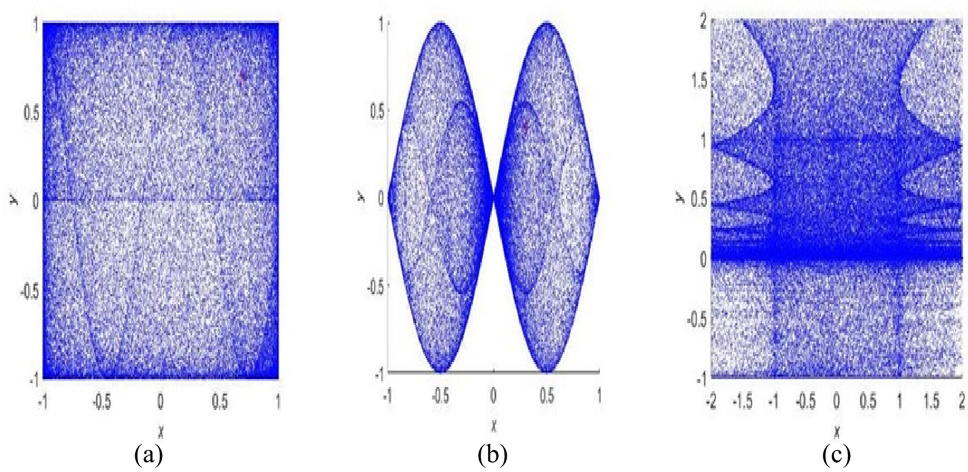

Figure 2: Attractors of (a) 2D-LICM, where $a = 0.6,\ k = 0.8$; (b) 2D-SIMM, where $a = 1,\ b = 5$; and (c) the new 2D hyperchaotic map, where $(r, k) = (0.8, 0.6)$.

2.3.2 Lyapunov Exponents Spectrum

A Lyapunov exponent (LE) is a measure of the rate between the neighbouring trajectories to where convergence or divergence occurs and can be defined as [31].

$$\lambda \cong \frac{1}{t} \ln \frac{\|\delta x(t)\|}{\|\delta x(0)\|} \tag{5}$$

where $\frac{\|\delta x(t)\|}{\|\delta x(0)\|}$ is the distance between two trajectories, or can be defined as [32]

$$\lambda = \lim \frac{1}{t} \sum t_i \lambda_i . \tag{6}$$

On the other hand, the Qs decomposition algorithm [33] calculated LEs and defined them as follows:

$$LE = \frac{1}{t} \sum_{i=1}^{N} |R_i(v, v)| \tag{7}$$

Where $v = 1,2, \dots$ and N is the number of iterations. The LEs λ_1 and λ_{12}, which have a distribution in the new hyperchaotic map with reference to r and k parameters, are illustrated in Figure 3. In Figures 3a and 3b, the system is hyperchaotic, where $r = 0.6, r = 2.6$, and $k = (0{,}100)$. In Figures 2c and 2d, the system also is hyperchaotic, where $k = 0.6$, k= 2.6, and $r = (0{,}100)$.

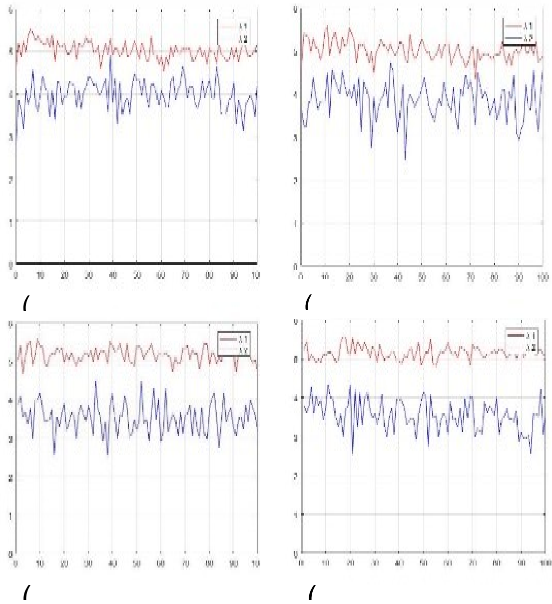

Figure 3: (a,b) LE spectrum of the new hyperchaotic map, where r = 0.6, r = 2.6, and k = (0,100).

(c,d) LE spectrum of the new hyperchaotic map, where k = 0.6, k = 2.6, and r = (0,100).

2.3.3 Approximate Entropy

Approximate entropy (ApEn) is a type of entropy that explains the quantitative complexity of a signal. ApEn is used to measure information that is necessary to know in order to predict a dynamical system. ApEn can be mathematically expressed as follows:

$$ApEn(m, r, n) = \Phi^{m}(r) - \Phi^{m+1}(r) \qquad (8)$$

Where m is the embedding dimension and r is the tolerance. Additionally, it can also be expressed as

$$\Phi^{m}(r) = [n - (m-1)\tau]^{-1} \sum_{i=1}^{n-(m-1)\tau} ln \frac{B_i}{n-(m-1)\tau} \qquad (9)$$

Where $m = 2$ and time delay $\tau = 1$.

Figure 4 shows the ApEn for several different chaotic maps such as 2D-LICM [1], 2D-SIMM [15], and 2D-HGSM [33]. We show that the new hyperchaotic map and 2D-LICM are close to some of the other maps. Thus, the new hyperchaotic map can be used to encrypt images that exhibit randomness and large chaotic sequences.

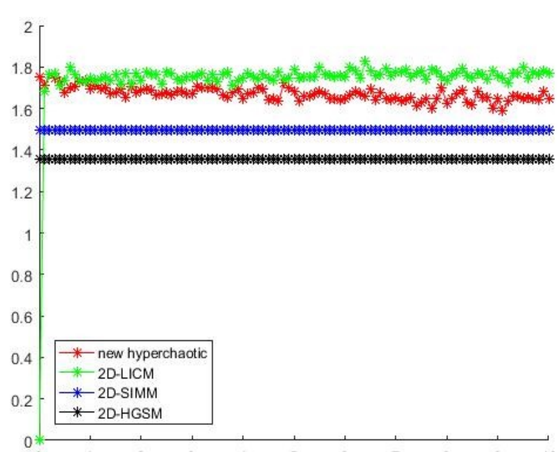

Figure 4: (ApEn) of several chaotic maps.

3. Simulation Results of Image Encryption Algorithm Based on the New 2D Hyperchaotic Map

In this section, a new algorithm for image encryption based on the new hyperchaotic map is introduced. This new algorithm consists of five steps to obtain a cipher image. The first step involves changing the location of the pixels. The second step includes dividing the image into four parts, with each part having a different key. The third and fourth steps involve confusion and diffusion operations, respectively. Confusion involves randomly shuffling the position of the pixels, while diffusion involves altering the values of the pixels. These operations are repeated twice. Eventually, the various parts of the image are merged in order to obtain the cipher image. The structure of the algorithm is illustrated in Figure 5.

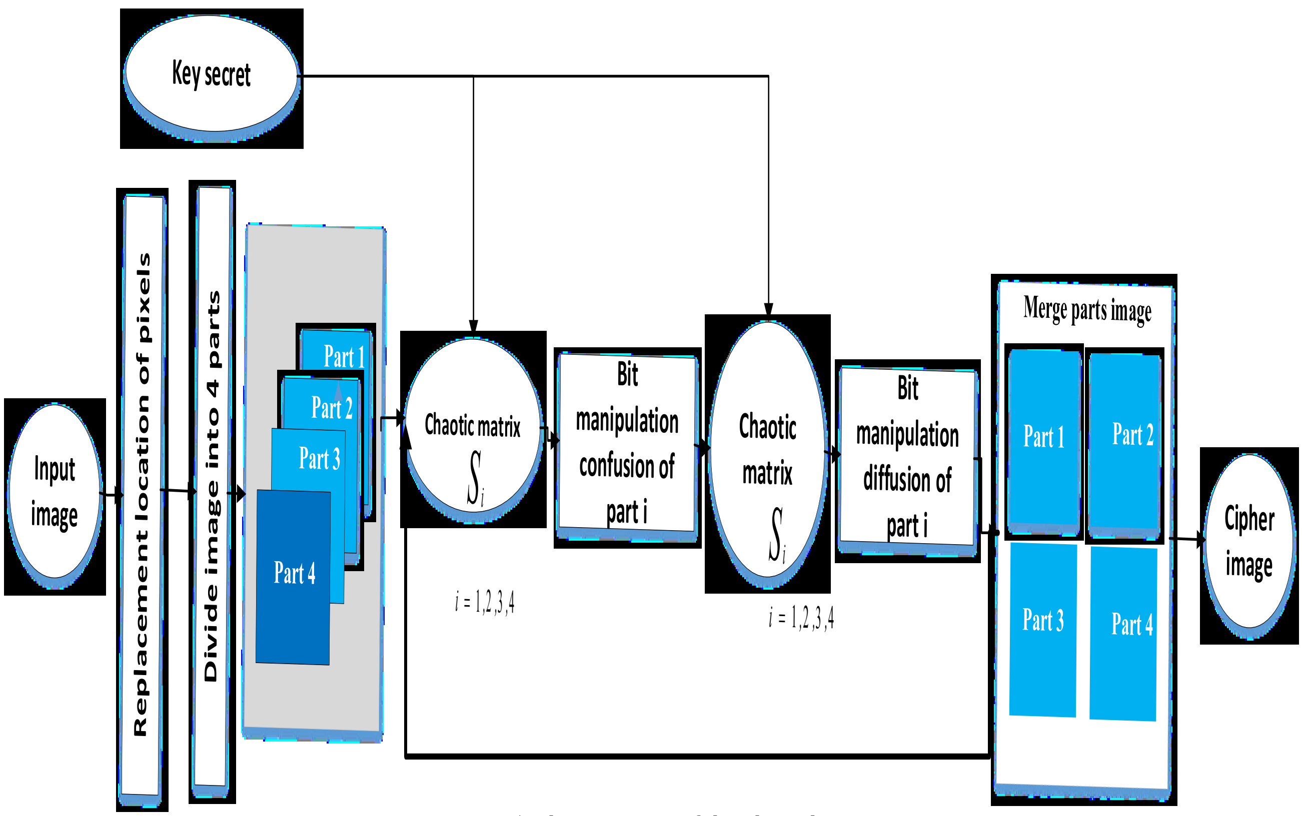

Figure 5: The structure of the algorithm

3.1 Changing the Location of the Pixels

The first step of our algorithm depends on the creation of a matrix of all elements, starting from the integer 1 and going to the maximum row corresponding to the dimension of the rows of the plain image. However, these rows are scattered. An example is shown in Table 1.

Table 1: Create a row and change elements' location of it

Old row	1	2	3	... MR
New row	25	10	19	... N_1

The same idea applies to the columns and is shown in Table 2.

Table 2: Create a column and change elements' location of it

Old column	1	2	3	... MC
New column	23	11	15	... N_2

Where MR and MC are the maximum row and maximum column, respectively. So, we created a matrix $N_1 \times N_2$ based on the new location of new rows and new columns, such that a change in location of the pixels of a plain image can be based on the new matrix $N_1 \times N_2$. For example, shown in Figure 6.

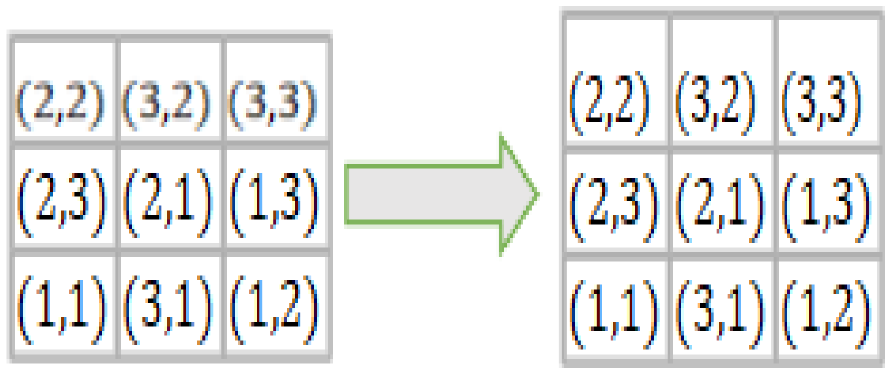

Figure 6: Changed locations of pixels of a plain image.

Then, the changed locations of the pixels of the plain image are shown in Figure 7.

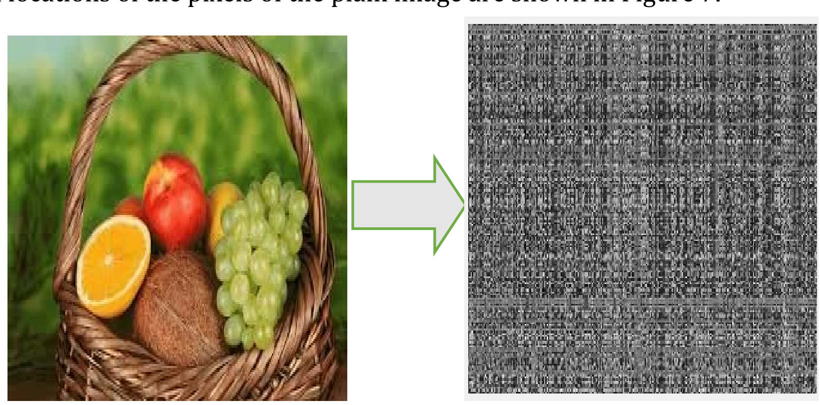

Figure 7: (a) Plain image and (b) changes in the pixels' locations of a plain image.

3.2 Dividing the Image into Four Parts

The second step of our algorithm involves dividing the resulting image generated after changing the locations of the image's pixels into four parts such that each part passes through the rest of the parts alone and becomes integrated into those other parts to obtain the cipher image. An example of this concept is shown in Figure 8.

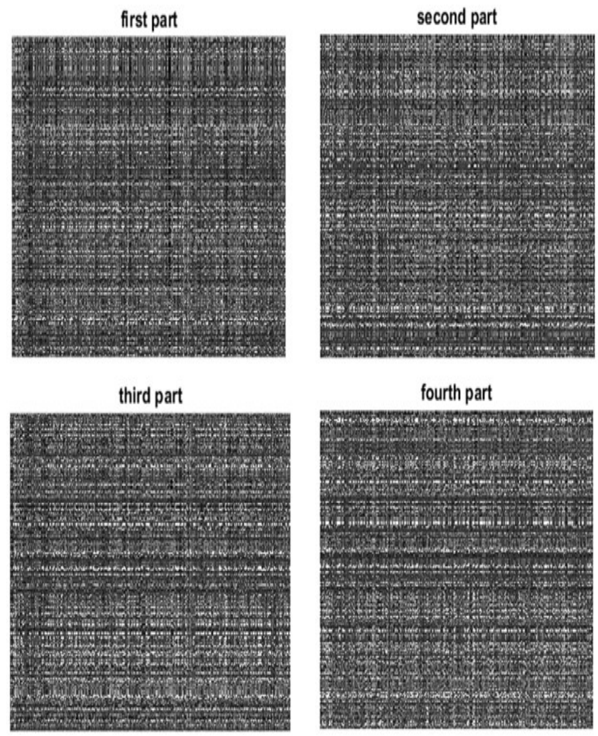

Figure 8: An example of dividing an image resulting from changes in the locations of pixels of an image into four parts.

3.3 Generation Keys

The key space should be larger than 2^{100} so as to avoid any attack on a chaotic encryption system [13]. Our algorithm has four different keys, such that each 256-bit key $K = \{x_0^k, y_0^k, a_0^k, w_1^k, w_2^k, G_1^k, G_2^k\}, k = 1,2,3,4 \ldots$, where (x_0^k, y_0^k, a_0^k) are the initial conditions of the new hyperchaotic system and $(w_1^i, w_2^i, G_1^i, G_2^i)$ are the involvement parameters. The algorithm for generating secret keys according to the new hyperchaotic map is shown in Algorithm 1.

Algorithm 1 The generation of initial states for the new hyperchaotic map.

Input: Secret key K with the length of 232 bits.

Output: Initial states (x_0^k, y_0^k, a_0^k) where $k = 1,2,3,4$.

$$x_0 = {\left(\sum_{i=1}^{52} K[i] \times 2^{52-i}\right)}/{2^{52}};$$

$$y_0 = {\left(\sum_{i=53}^{104} K[i] \times 2^{104-i}\right)}/{2^{52}};$$

$$a_0 = {\left(\sum_{i=105}^{156} K[i] \times 2^{156-i}\right)}/{2^{52}};$$

$$w_1 = {\left(\sum_{i=157}^{180} K[i] \times 2^{180-i}\right)}/{2^{24}};$$

$$w_2 = {\left(\sum_{i=181}^{204} K[i] \times 2^{204-i}\right)}/{2^{24}};$$

$G_1 = {\left(\sum_{i=205}^{228} K[i] \times 2^{228-i}\right)}/{2^{24}};$

$G_2 = {\left(\sum_{i=229}^{252} K[i] \times 2^{252-i}\right)}/{2^{24}};$

For $i = 1$ to 4

$x_0^k = (x_0 + w_1 \times G_1) mod\ 1;$

$y_0^k = (y_0 + w_2 \times G_2)\ mod\ 1;$

if $x_0^k \& y_0^k = 0$ **then**

$x_0^k = 0.7271;$

$y_0^k = 0.7271;$

end if

$a_0^k = (x_0/y_0) + (x_0 + w_1 \times G_1)\ mod\ 1;$

end for

From this algorithm, we generated four secret keys that depended on initial conditions(x_0^k, y_0^k, a_0^k). The four generated keys are shown below:

$K_1 = D0DA73E21A30D089C2A04B06040C545C508800C48308428B03260204C0402C2$

$K_2 = 097147A4988C438A430040D1522314000E3800C07080556362144150854O8C2$

$K_3 = 20501722604AC0AB35B43820B752581E8A0000830203BE08140AC601AC08208$

$K_4 = 11924E890E3A0AE2880804B1288302000C308506901019E88AAC01589C5C920$

where each key has a 256-bit.

So, the initial states of (x_0^1, y_0^1, a_1^1), (x_0^2, y_0^2, a_1^2), (x_0^3, y_0^3, a_1^3) , and (x_0^4, y_0^4, a_1^4) are $(0.9217, 0.1395, 6.7605)$, $(0.0442, 0.2286, 0.3407)$, $(0.2528,\ 0.1684, 1.9496)$, and $(0.3764, 0.9881, 1.2005)$, respectively. From this collection of initial states, we can create S_1, S_2, S_3 and S_4 matrices by way of the new 2D hyperchaotic map. Then, we used these matrices to apply the confusion and diffusion operations. We can apply this algorithm to any digital image of any formula.

3.4 Bit Manipulation Confusion

The output distribution is affected depending on the secret key of the property of confusion [34]. The random confusion of bit manipulation shuffles the pixel locations within the image, depending on the chaotic matrix generated by the new 2D hyperchaotic map. We suppose that P_i, where $i = 1,2,3,4$, is some part divided from the plain image, while S_i, where $i = 1,2,3,4$, is the generated chaotic matrix. All elements of the matrix are represented by p bits. The definition of bit manipulation is expressed as follows:

$$T = B(P, S).$$

We illustrate and describe the detailed process of bit manipulation confusion in Algorithm 2. Figure 9 shows an example of bit manipulation confusion. The streams of the binary from S are placed in the locations of the most significant bits. Thus, S controls the change of the location of the pixels. In the confusion operation, the order or arrangement of pixels in any location or part of the image can be changed.

Algorithm 2 Bit manipulation confusion $T = B\ (P, S)$.

Input: Image P and chaotic matrix S. They are of size $Q \times W$ and their elements are represented by p bits.

Output: Bit manipulation confusion result T.

Initial a matrix R of size $Q \times W$;

$q = \lceil log2(QW) \rceil$;

for $i = 1$ **to** Q **do**

for $j = 1$ **to** W **do**

$t = (i - 1)W + j$;

$tb = Bin(t, q)$; {$Bin\ (x, n)$ transforms the integer number x into n bits.}

$R_{i,j} = Joint\left(S_{i,j}, tb, Pi_{i,j}\right)$; {$Joint\ (x1, x2, x3)$ joints the 3 binary sequences $x1, x2, x3$ into one binary sequence by order.}

end for

end for

$R\ = Sort\ (R)$; {Sort $R(X)$ sorts the matrix X along horizontal direction.}

$R\ = Sort\ (R)$; {Sort $R(X)$ sorts the matrix X along vertical direction.}

$T = FetEnd(R1{:}\,Q, 1{:}\,W, p)$; {$FetEnd$ (x, n) fetches the last n bits from the binary sequence x.

3.5 Bit Manipulation Diffusion

The diffusion operation exerts a significant effect on ciphertext change, such that a one-bit change of a plain image causes each part of the ciphertext to change by 50% [34]. We used the chaotic matrix S to change pixels. This operation was repeated twice. The change can be posted in single pixels throughout the entire image. The definition of bit manipulation diffusion is expressed as follows:

$$O_{i,j} = \begin{cases} T_{i,j} \oplus T_{Q,W} \oplus S_{i,j} & for\ i = 1, j = 1 \\ T_{i,j \oplus} O_{i-1,W} \oplus S_{i,j} & for\ i \neq 1, j = 1 \\ T_{i,j} \oplus O_{i,j-1} \oplus S_{i,} for\ j \neq 1 & \end{cases} \quad (10)$$

Where O is the bit manipulation diffusion result and $\oplus$ is the bitwise XOR operation. This part is an inverse operation of the decryption process.

01111101 125	11010110 214	00100101 37	11111001 249	01010100 84
00000001 1	00111001 57	00001000 8	00011100 28	00010011 19
00110011 51	00010101 21	00011111 31	10110110 182	11001011 203
00100101 37	11010010 210	00011010 26	00001101 13	01101000 104
11101111 239	01111010 122	11011111 223	11000111 199	01010100 84

S

00001 1	00010 2	00011 3	00100 4	00101 5
00110 6	00111 7	01000 8	01001 9	01010 10
01011 11	01100 12	01101 13	01110 14	01111 15
10000 16	10001 17	10010 18	10011 19	10100 20
10101 21	10110 22	101111 23	11000 24	11001 25

I

01111000 120	01111001 121	01111010 122	01111011 123	01111100 124
01111101 125	01111110 126	01111111 127	10000000 128	10000001 129
10000010 130	10000011 131	10000100 132	10000101 133	10000110 134
10000111 135	10001000 136	10001001 137	10001010 138	10001011 139
10001100 140	10001101 141	10001110 142	10001111 143	10010000 144

P

011111010000101111000 1024376	110101100001001111001 1753721	001001010001101111010 303994	111110010010001111011 2040955	010101000010101111100 689532
000000010011001111101 9853	001110010011101111110 468862	000010000100001111111 67711	000111000100110000000 231808	000100110101010000001 158337
001100110101110000010 420738	000101010110010000011 175235	000111110110110000100 257412	101101100111010000101 1494661	110010110111110000110 1666950
001001011000010000111 307335	110100101000110001000 1724808	000110101001010001001 217737	000011011001110001010 111498	011010001010010001011 857227
111011111010110001100 1963404	011110101011010001101 1005197	110111111011110001110 1832846	110001111100010001111 1636495	010101001100110010000 694672

$R = \text{Joint}(S, I, P)$

Sort by Row

001001010001101111010 303994	010101000010101111100 689532	011111010000101111000 1024376	110101100001001111001 1753721	111110010010001111011 2040955
000000010011001111101 9853	000010000100001111111 67711	000100110101010000001 158337	000111000100110000000 231808	001110010011101111110 468862
000101010110010000011 175235	000111110110110000100 257412	001100110101110000010 420738	101101100111010000101 1494661	110010110111110000110 1666950
000011011001110001010 111498	000110101001010001001 217737	001001011000010000111 307335	011010001010010001011 857227	110100101000110001000 1724808
010101001100110010000 694672	011110101011010001101 1005197	110001111100010001111 1636495	110111111011110001110 1832846	111011111010110001100 1963404

Sort by Column

000000010011001111101 9853	000010000100001111111 67711	000100110101010000001 158337	000111000100110000000 231808	001110010011101111110 468862
000011011001110001010 111498	000110101001010001001 217737	001001011000010000111 307335	011010001010010001011 857227	110010110111110000110 1666950
000101010110010000011 175235	000111110110110000100 257412	001100110101110000010 420738	101101100111010000101 1494661	110100101000110001000 1724808
001001010001101111010 303994	010101000010101111100 689532	011111010000101111000 1024376	110101100001001111001 1753721	111011111010110001100 1963404
010101001100110010000 694672	011110101011010001101 1005197	110001111100010001111 1636495	110111111011110001110 1832846	111110010010001111011 2040955

Truncate the Pixel Part

01111101 125	01111111 127	10000001 129	10000000 128	01111110 126
10001010 138	10001001 137	10000111 135	10001011 139	10000110 134
10000011 131	10000100 132	10000010 130	10000101 133	10001000 136
01111010 122	01111100 124	01111000 120	01111001 121	10001100 140
10010000 144	10001101 141	10001111 143	10001110 142	01111011 123

Opertion result T

Figure 9: An example of bit manipulation confusion of some parts of a plain image.

The decryption process of this part is to do the inverse operation, which can be mathematically expressed as follows:

$$O_{i,j} = \begin{cases} T_{i,j} \oplus O_{i,j-1} \oplus S_{i,j} \quad for\ j \neq 1 \\ T_{i,j\oplus} O_{i-1,W} \oplus S_{i,j},\ for\ i \neq 1, j = 1 \\ T_{i,j} \oplus T_{Q,W} \oplus S_{i,j}\ for\ i = 1, j = 1 \end{cases} \tag{11}$$

By using two different chaotic matrices and, following two rounds of the bit manipulation confusion and diffusion operations, we merged the four parts of a plain image to obtain a cipher image that is unrecognizable.

4. Simulation Results and Reliability

Any image encryption system should have the strength to encrypt any image with different formulas into a random image without clear milestones. In this section, we introduce the simulation results of our image encryption for different kinds of images, and its reliability is also discussed.

4.1 Simulation Results

In this study, we used the MATLAB language (MathWorks, Natick, MA, USA) to implement our algorithm on different types of greyscale images and apply it to RGB images. Figure 10 illustrates the simulation results of greyscale images. In this simulation, we can observe that our system can encrypt images into random cipher images that do not have clear milestones. By using different keys to encrypt images, we can reconstruct the original image. Figure 10 also illustrates the histograms of the greyscale images.

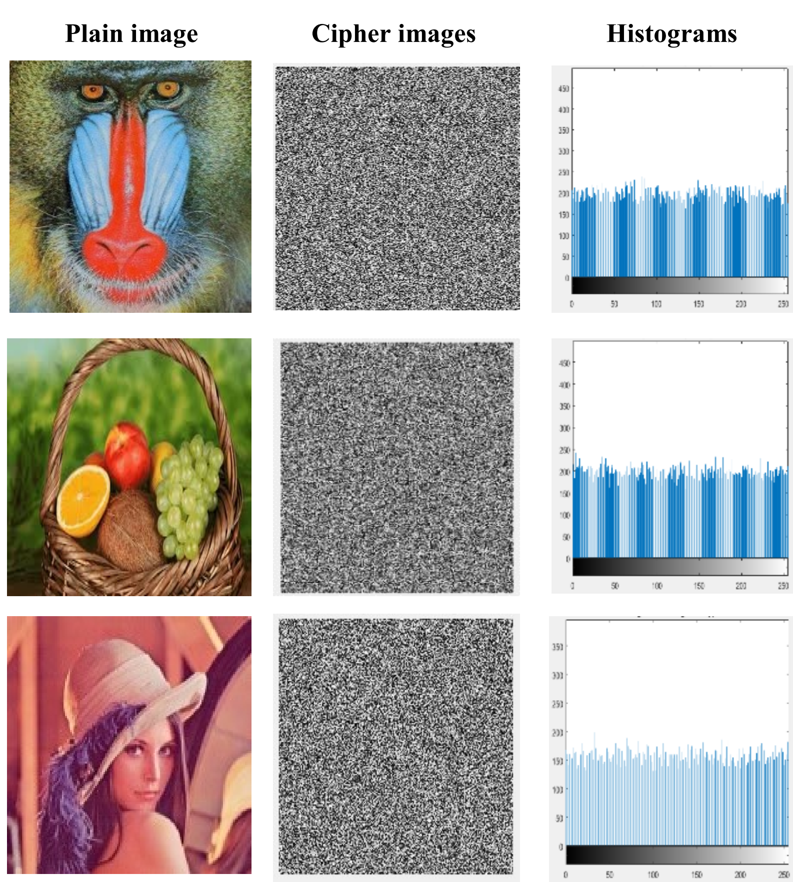

Figure 10: Simulation results of our system. (a) Plain images; (b) cipher images; and (c) histograms of the encryption image.

4.2 Correlation between Adjacent Pixels

Each algorithm of encryption is considered a good algorithm only if it can break the correlations between adjacent pixels. The correlation can be measured from the adjacent between pixels according to the following mathematical relationship:

$$\rho_{xy} = \frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x \sigma_y} \qquad (12)$$

Where x and y are the data sequences; μ_x and μ_y are the average values of x and y, respectively; and σ_x and σ_y are the standard deviations of x and y, respectively. Thus, $\rho_{xy} \in [0, 1]$ and a high correlation indicate large values, which are not favourable. Table 3 shows the correlation of several different images of our proposal, while Table 4 presents the comparison correlation of the obtained Lena Söderberg image with the results of the other methods. The distributions of pixels in the horizontal as well as vertical and diagonal directions are shown in Figure 11. In a plain image, the majority of points are close to the diagonal line of the axis, while the distribution is random and takes up more space of the cipher image.

Table 3: Correlation between some different images

Name	Original image			Encryption image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon	0.8584	0.7649	0.7321	0.0111	0.0133	0.0030
Boat	0.9397	0.8830	0.8385	0.0538	0.0125	0.000845
Fruits	0.9155	0.9011	0.8483	0.0386	0.0385	0.0043
House	0.9753	0.9478	0.9271	0.0436	0.0370	0.000053

Table 4: Comparison correlation of the Lena Söderberg image with other methods

Direction	Plain image	Wang [35]	Liu [13]	Hua [12]	Cao [1]	Our proposal
Horizontal	0.965352	0.0331	0.0030	0.0013	0.0019	0.0018
Vertical	0.932559	0.0169	0.0024	0.0 0 06	0.0012	0.0012
Diagonal	0.907119	0.0057	0.0034	0.0019	0.0 0 09	0.009

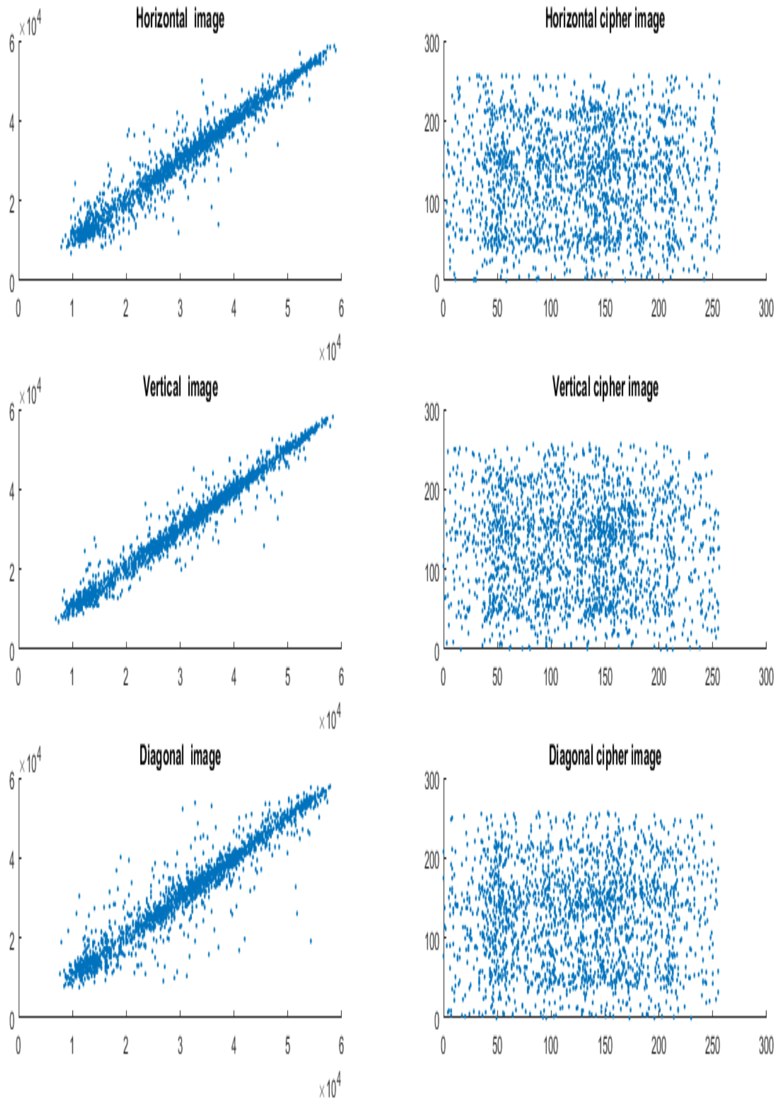

Figure 11: Pixel distribution. On left, the plain image is shown in the first column and the cipher image is shown in the second column. On right, data for the horizontal direction, vertical direction, and diagonal direction can be seen.

4.3 Information Entropy

Information entropy is a measure of greyscale randomness and can be expressed as follows:

$$H(M) = -\sum_{i=1}^{L} p(m_i) \log p(m_i) \quad (13)$$

Where L is the total number of symbol m_i and $p(m_i)$ is the probability of symbol m_i. The maximum entropy of information is approximately 8 and is applied to the grey-level images. Table 5 shows the information entropy of different images. This table also details the comparison between 2D-LICM [1], 2D-SHAM [16], and our proposed algorithm. The results reveal that the information entropy of some standard images encrypted by our proposed algorithm is higher than that obtained by 2D-LICM [1] and SHAM [16]. This finding indicates that the randomness of image encryption of our algorithm is good.

Table 5: Information entropy of different images for some different methods such as 2D-LICM, 2D-SHAM, and our proposal.

Name	Peppers	Lena	Flowers	Boats	Man	House	Baboon	Jump
Our proposal	7.9995	7.9993	7.9980	7.9965	7.9993	7.9973	7.9994	7.9990
2D-LICM [1]	7.9974	7.9976	7.9973	7.9972	7.9974	-	-	-
2D-SHAM [16]	7.9964	7.9965	-	-	7.9964	7.9961	-	-

4.4 Resisting Differential Attack Analysis

Resistance to differential attacks is determined using two measures: (1) the number of pixel change rates (NPCR) and the number of changed pixels in the encrypted image and (2) the unified average changing intensity (UACI), which is the average of the differences between two encrypted images. If we have two original images, O_1 and O_2 ,with a one-bit difference, then C_1 and C_2 are the encryption images corresponding to the original images, respectively. NPCR and UACI can be expressed as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \tag{14}$$

$$UACI = \frac{1}{M \times N}\left(\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right) \times 100\% \tag{15}$$

$$D(i,j) = \begin{cases} 1 & if \quad C_1(i,j) \neq C_2(i,j) \\ 0 & if \quad C_1(i,j) = C_2(i,j) \end{cases}$$

Where M and N are the width and height of the image, respectively. Additionally, NPCR and UACI are greater than 99% and 31%, respectively. Table 6 shows the results of the NPCR and UACI of several different images. The results show that our algorithm has a good capability to withstand differential attacks.

Table 6: Some results of NPCR and UACI of some different images

Name	Lena	Peppers	Jump
NPCR	0.996	0.992	0.996
UACI	0.300	0.313	0.334

5. Conclusion

This study proposes a new 2D hyperchaotic map derived from three standard maps, namely, those of the logistic, circle, and Henon kind, respectively. The properties of the dynamics of this system are investigated using Lyapunov exponents, trajectories, bifurcation diagrams, and a sensitivity dependence test. The results of all of these tests indicate that our system is hyperchaotic and highly sensitive to the initial values and control parameters. The algorithm of sample entropy is also used to investigate the complexity of our system. We additionally propose a new algorithm of image encryption on the basis of the new 2D hyperchaotic map. Notably, confusion and diffusion can be achieved with this algorithm, which are fundamental demands. Moreover, the suggested algorithm has high security in external attack resistance, as well as low time complexity, which enables faster processing and faster data transmission and prevents attacks like that which require more time to hack data. Therefore, the suggested algorithm offers a high ability to encrypt images and video.

References

[1] C. Cao, K. Sun and W. Liu, "A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map", *Signal Processing*, 2018, 143, 122-133.

[2] N. Zhou, S. Pan, S. Cheng and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing", *Optics & Laser Technology*, 2016, 82, 121-133.

[3] X. Chai, Z. Gan, Y. Chen and Y. Zhang, "A visually secure image encryption scheme based on compressive sensing", *Signal Processing*, 2017, 134, 35-51.

[4] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen and X. He, "A review of compressive sensing in information security field", *IEEE access*, 2016, 4: 2507-2519.

[5] Y. Luo, M. Du and J. Liu, "A symmetrical image encryption scheme in wavelet and time domain", *Communications in Nonlinear Science and Numerical Simulation*, 2015, 20.2: 447-460.

[6] X. Wu, D. Wang, J. Kurths and H. Kan, "A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system", *Information Sciences*, 2016,349, 137-153.

[7] J. Zhang, D. Fang and H. Ren, "Image encryption algorithm based on DNA encoding and chaotic maps", *Mathematical Problems in Engineering*, 2014.

[8] A. Nag, J. P. Singh, S. Khan, S. Ghosh, S. Biswas, D. Sarkar and P. P. Sarkar, "Image encryption using affine transform and XOR operation", *In: 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies. IEEE*, 2011, 309-312.

[9] S. V. Chalam and M. K. Singh, "Unified Approach with Neural Network for Authentication, Security and Compression of Image: UNICAP", *International Journal of Image Processing (IJIP),* 2012, 6(1), 13.

[10] M. Suresh and M. Neema, "Hardware implementation of blowfish algorithm for the secure data transmission in Internet of Things", *Procedia Technology*, 2016, 25, 248-255.

[11] A. P. U. Siahaan, "*RC4 Technique in Visual Cryptography RGB Image Encryption*", 2017.

[12] Z. Hua, and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map", *Information Sciences,* 2016, 339, 237-253.

[13] W. Liu, K. Sun and C. Zhu. "A fast image encryption algorithm based on chaotic map", *Optics and Lasers in Engineering*, 2016, 84, 26-36.

[14] Y. Zhou, L. Bao and C. P Chen, "A new 1D chaotic system for image encryption", *Signal processing*, 2014, 97, 172-182.

[15] Z. Hua, Y. Zhou, C. M. Pun and C. P. Chen, " 2D Sine Logistic modulation map for image encryption", *Information Sciences*, 2015, 297, 80-94.

[16] H. Natiq, N. M. G. Al-Saidi, M. R. M. Said, and A., Kilicman, "A new hyperchaotic map and its application for image encryption", *The European Physical Journal Plus*, 2018, 133 (1), 6.

[17] H. Liu, and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps", *Computers & Mathematics with Applications*, 2010, 59 (10), 3320-3327.

[18] R. Ye, "A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism", *Optics Communications*, 2011, 284 (22), 5290-5298.

[19] S. Sheng, and X. Wu, "A novel bit-level image encryption scheme using hyper-chaotic systems", *In Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on* (pp. 1015-1019). IEEE.

[20] X. G. Wang,, S. M. Lo, H. P. Zhang, and W. L. Wang, " A novel conceptual fire hazard ranking distribution system based on multisensory technology", *Procedia engineering*, 2014, 71, 567-576.

[21] X. Huang, and Y. Guodong, "An image encryption algorithm based on hyper-chaos and DNA sequence", *Multimedia tools and applications,* 2014, 72 (1), 57-70.

[22] R. Enayatifar, H. J. Sadaei, A. H. Abdullah, M. Lee, and I. F. Isnin, "A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata", *Optics and Lasers in Engineering*, 2015, 71, 33-41.

[23] X. Wang, L. Lintao, and Z. Yingqian, "A novel chaotic block image encryption algorithm based on dynamic random growth technique", *Optics and Lasers in Engineering*, 2015, 66, 10-18.

[24] X. Y. Wang, Y. Q. Zhang, and X. M. Bao, "A novel chaotic image encryption scheme using DNA sequence operations", *Optics and Lasers in Engineering*, 2015, 73, 53-61.

[25] S. Koppu, and V. M. Viswanatham, “A fast enhanced secure image chaotic cryptosystem based on hybrid chaotic magic transform”, *Modelling and Simulation in Engineering*, 2017.

[26] L. Liu, and M. Suoxia, “A new image encryption algorithm based on logistic chaotic map with varying parameter”, *SpringerPlus*, 2016, 5 (1), 289.

[27] Y. Li, W. Wenxian, Z. Jun, C. Hongsheng and Z. Peng, “10B areal density: A novel approach for design and fabrication of B4C/6061Al neutron absorbing materials”, *Journal of Nuclear Materials,* 2017,487, 238-246.

[28] M. Hénon, “A two-dimensional mapping with a strange attractor. In The Theory of Chaotic Attractors”, *Springer, New York, NY.* 1976, (pp. 94-102).

[29] R. M. May, “Simple mathematical models with very complicated dynamics”, *In the Theory of Chaotic Attractors springer*, New York, NY. 2004, 85-93.

[30] D. He, C. He, L. G. Jiang, H. W. Zhu, and G. R. Hu, “Chaotic characteristics of a one-dimensional iterative map with infinite collapses”, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2001, 48 (7), 900-906.

[31] J. LaSalle, “Some extensions of Liapunov's second method”, *IRE Transactions on circuit theory*, 1960, 7(4), 520-527.

[32] L. Arnold, H. Crauel, and J.P. Eckmann, “Lyapunov Exponents*”, proceedings of a conference held in Oberwolfach,* Springer*,* 1990, 388.

[33] H. Natiq, M. N. Al-Saidi and M. R. M. Said, “Complexity and dynamic characteristics of a new discrete-time hyperchaotic model”, *International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA),* 2017, 1–6.

[34] D. Xiao, X., Liao, and S. Deng, “One-way Hash function construction based on the chaotic map with changeable-parameter”, *Chaos, Solitons & Fractals*, 2005, 24 (1), 65-71.

[35] X. Wang, Q. Wang and Y. Zhang, “A fast image algorithm based on rows and columns switch”, *Nonlinear Dynamics*, 2015, 79 (2), 1141-1149.