SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9**:634–651
Published online 16 December 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1386

RESEARCH ARTICLE

A hybrid image encryption algorithm using chaos and Conway's game-of-life cellular automata

Brindha Murugan[1]*, Ammasai Gounden Nanjappa Gounder[2] and Sriram Manohar[1]

[1] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India
[2] Department of Electrical and Electronics and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

ABSTRACT

In this paper, a new image encryption algorithm employing the combination of chaos and cellular automata is proposed. The proposed algorithm consists of both permutation and diffusion stages. While the permutation process is carried out using logistic map and Conway's game-of-life cellular automata, the diffusion process is carried out using Chebyshev map and Lorenz equation. Further, a complex matrix generated from the plain image is used as an additional component in the diffusion process, which enables the encrypted image to exhibit a strong sensitivity to the input image. The proposed algorithm has been tested with various input images, and the performance is compared with other existing algorithms. The performance metrics obtained on the developed algorithm such as high key space, ideal number of pixels change rate and unified average changing intensity values, and very less correlation among the adjacent pixels demonstrate the high effectiveness and security features of the proposed algorithm. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS

chaos; cellular automata; Chebyshev map; image encryption; Lorenz equation; row scrambling; column scrambling; diffusion

***Correspondence**

Murugan Brindha, Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India.

E-mail: brindham@nitt.edu

1. INTRODUCTION

Nowadays, because of proliferation of the Internet, communication of data becomes very much popular, and ensuring authorized access to such crucial data is essential [1]. The crucial data include text, digital images, videos, and audios, and among them, the digital images play an important role in communication because of the rapid development in multimedia technology [2]. Also, many applications such as military, confidential business, and medical imaging need the images to be stored and transmitted in a secured way. To fulfill the security needs of such images, which contain such confidential information, good security tools have to be devised.

Cryptography is one of the widely used security tools by which the images can be encrypted and produced in a non-readable format. Many traditional number theory-based encryption algorithms such as Rivest Shamir Adleman, Advanced Encryption Standard (AES), Data Encryption Standard, and Triple Data Encryption Standard were proposed for the encryption of data. But these traditional ciphers are found to be not suitable for image encryption because of the need for higher computational power and time [3].

In particular, AES, one of the traditional ciphers, is vulnerable to square attacks, side channel attacks, extended sparse linearization, and differential attack. The dynamic nature of the key used in the AES algorithm is also under research at present. Since past decades, many researchers found that there is a strong relationship between chaos and cryptography. Image encryption based on chaos theory was initially designed by Mathews in 1989 [4]. The special characteristics of chaos such as sensitivity to initial conditions and system parameters, pseudo-randomness, and ergodicity have diverted the attention of researchers towards chaos theory for image encryption [5–9].

Chaotic maps, which are the building blocks of chaotic ciphers, can be widely classified into discrete and continuous-time systems. Discrete systems use a set of difference equations, whereas the continuous systems are managed by derivatives of differential equations. Many

researchers concentrate on Arnold cat map, Baker map, Standard map, and Skew tent map, which are discrete-time systems [10] because of their easy implementation, whereas the advantage of continuous-time systems is that it adds step size, which is the additional key parameter to the existing key space. In general, any encryption scheme is based on two steps. One is permutation, and the other is diffusion. The location of the pixels is changed in permutation [6], whereas in diffusion, the gray-level value itself is changed either by point by point [11] or block by block [12,13]. The combination [14,15] of the aforementioned two steps yields better results than their individual implementation.

2. RELATED WORK

After the introduction of cellular automata (CA) by John von Neumann [16,17] and subsequently the game of life by John Conway, the development of encryption algorithms based on the combination of chaos and CA was increasingly becoming popular. Lin Teng and Xingyuan Wang [18] discuss a bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive technique, and it has a low number of pixels change rate (NPCR) of 93.67% and a high correlation of 0.0243. Xingyuan Wang and Canqi Jin's [19] game-of-life permutation and piecewise linear chaotic map chaotic system encryption scheme has a high correlation of 0.038 and key space of only 10^{66}. The chaos-based bit-level permutation scheme proposed by Chong Fu, Bin-bin Lin, Yu-sheng Miao, Xiao Liu, and Junjie Chen [20] results in low entropy of 7.9880. In Di Xiao and Frank Y. Shih's [21] synchronizing method to improve security of the multichaotic systems, the unified average changing intensity (UACI) value is 26.27%, which is lesser compared with the ideal value.

The scheme proposed by Xingyuan Wang, Lintao Liu, and Yingqian Zhang [22] based on hybrid chaotic maps and dynamic random growth technique gives an ideal NPCR value, but the UACI value is only 26.5869% for certain images, which is very less compared with the ideal value. Further, the key space is only 10^{96}, and it has an entropy value of 7.997. The scheme by Jun-xin Chen, Zhi-liang Zhu, Chong Fu, Hai Yu, and Li-bo Zhang [23], based on dynamic state variables selection mechanism, has the key space as low as 10^{60}. Radu Boriga, Ana Cristina Dascalescu, and Iustin Priescu [24] have proposed an image encryption algorithm using a new hyperchaotic map. The NPCR and UACI values for this scheme are 99.24% and 33.13%, respectively, which are very less compared with the ideal values.

Pareek, N. K., Patidar, V., and Sud, K. K. [25] have proposed an encryption algorithm for gray images using a secret key of 128-bit size. In this scheme, the key space is only 10^{38}, the entropy is 7.9952, and the UACI values fall down to 32.11 (on average). Sun, F., and Liu, Z. L. S. [26], have proposed a new cryptosystem based on spatial chaotic system. Although this scheme exhibits a good key space, the entropy is 7.9965. Jianhua, L., and Hui, L. [27], have proposed a color image encryption scheme based on AES with two-dimensional (2D) chaotic map. It yields less correlation of adjacent pixels and good NPCR and UACI values. But this scheme becomes practical only after certain rounds of processing. In Abdul, H. A., Rasul, E., and Malrey L. [28] scheme, a hybrid genetic algorithm and chaotic function model for image encryption is proposed. This scheme yields a good entropy value of 7.9978 and less correlation along diagonal direction but takes more iterations and has a very small key space of 10^{12}.

In the present paper, an efficient permutation-based and diffusion-based image encryption algorithm using the combination of chaos along with CA is proposed. This new algorithm uses the logistic map for pseudo-random number generation, and the permutation process is carried out using Conway's game-of-life CA (CGLCA). It also uses Lorenz equation and Chebyshev map for diffusion process. The highlights of this cipher are as follows: (i) It yields ideal NPCR and UACI values, which indicates the resistance to differential cryptanalysis. (ii) It has extremely good key space of 10^{150}. (iii) It provides high key sensitivity and plain text sensitivity. (iv) Because full encryption is performed, no information about the plain image is leaked out. (v) It is simple to implement in hardware and very fast in nature.

Section 3 gives a brief review of logistic map, Chebyshev map, and Lorenz equation, which are used for the formulation of the algorithm. The development of the algorithm with details of permutation and diffusion processes is presented in Section 4. The details of security analysis carried out for assessing the quality of the proposed algorithm are furnished in Section 5, and Section 6 summarizes the conclusions of the paper.

3. BRIEF REVIEW OF BACKGROUND MATERIAL

In this section, the basic concepts of the logistic map (used for pseudorandom bit generation), Chebyshev map, and Lorenz equation (used for the diffusion process) are discussed.

3.1. Logistic map for pseudo-random bit generation

In the proposed algorithm, two one-dimensional logistic maps are used to generate the pseudo-random number bits with desirable properties [29]. These pseudo-random bits are used to generate the initial state of the CA. The way by which the pseudo-random bits are generated is illustrated in Figure 1.

The logistic map 1 and logistic map 2 are defined by the mathematical equations (1) and (2)

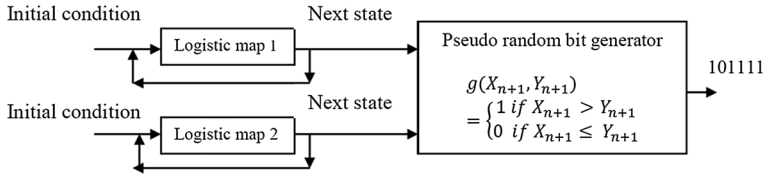

Figure 1. Pseudo-random bit generation using logistic map.

$$X_{n+1} = \mu^{(1)} X_n (1 - X_n) \quad (1)$$

$$Y_{n+1} = \mu^{(2)} Y_n (1 - Y_n) \quad (2)$$

where X_n and Y_n are called as state variables and they have values between 0 and 1; X_0 and Y_0 represent the initial conditions, and $\mu^{(1)} \& \mu^{(2)}$ are the system parameters that can have any value between 0 and 4. The logistic map shows chaotic character in the interval [0, 1] only when both $\mu^{(1)} \& \mu^{(2)}$ are between 3.569945 and 4. This property of the logistic map is used for generating the random bits for the initial state of the 2D CGLCA.

3.2. Chebyshev map

The equation for this map is given by (3);

$$C_{k+1} = \cos\left(\mu^{(3)} \times \cos^{-1}(C_k)\right) \quad (3)$$

C_0 is a value between -1 and +1, and $\mu^{(3)} \geq 3.569946$. The value of $\mu^{(3)}$ is set as approximately between 3.98 and 4.00; as for these values, this map generates highly chaotic sequences, which is necessary for generating random numbers, making the key hard to break. C_k is then converted to whole numbers between -256 and +256, by multiplying them with 256 and applying the greatest-integer function. The negative numbers and even numbers are discarded.

3.3. Lorenz equation

Lorenz equation [30,31] is a set of ordinary differential equations that exposes the chaotic behavior for certain initial values and key parameters. It is a continuous-time system that adds an additional key parameter known as step size. The equations are as follows:

$$\frac{dx}{dt} = s(x - y) \quad (4)$$

$$\frac{dy}{dt} = x(l - z) - y \quad (5)$$

$$\frac{dz}{dt} = xy - bz \quad (6)$$

where x, y, z make the system states, t is the time, and s, l, and b are known as system parameters. For $s = 10$, $l = 28$, and $b = 8/3$, the system exhibits chaotic behavior.

4. THE DEVELOPMENT OF THE PROPOSED ALGORITHM

Let $I(x, y)$ denote the input image (plain image) of size $M \times N$ and $E'(x, y)$ denote the encrypted image. Initially, two logistic maps act as a pseudo-random bit generators to form the initial state for the CA. The successive states are formed by using the rules of CGLCA. The final state of the CA is used for the row and column scrambling processes of the input image. Diffusion is carried out by Chebyshev map, Lorenz equation, and another complex matrix generated from the plain image. Because of this matrix, there exists a strong sensitivity to the input image. The flow chart depicting the permutation and diffusion process steps of the proposed algorithm is shown in Figure 2.

4.1. Permutation process

The following steps are involved in the permut ation process:

Step 1 Set the initial values of $\mu^{(1)}, \mu^{(2)}, X_0$, and Y_0 for the chaotic system, that is, the one-dimensional logistic map where X_0 and Y_0 lie between 0 and 1, and $\mu^{(1)} \& \mu^{(2)}$ lie between 3.5 and 4. The consecutive values of X and Y are determined using the following equations:

$$X_{k+1} = \mu^{(1)} X_k (1 - X_k) \quad (7)$$

$$Y_{K+1} = \mu^{(2)} Y_k (1 - Y_k) \quad (8)$$

Step 2 Evolve the successive states of the system X_1, $X_2, X_3 \ldots \ldots X_{M \times N}$ and $Y_1, Y_2, Y_3, \ldots \ldots Y_{M \times N}$.

Step 3 Generate the initial state of the 2D CA Z_0, which is a binary $M \times N$ matrix generated by the outputs of the logistic maps in the following way:

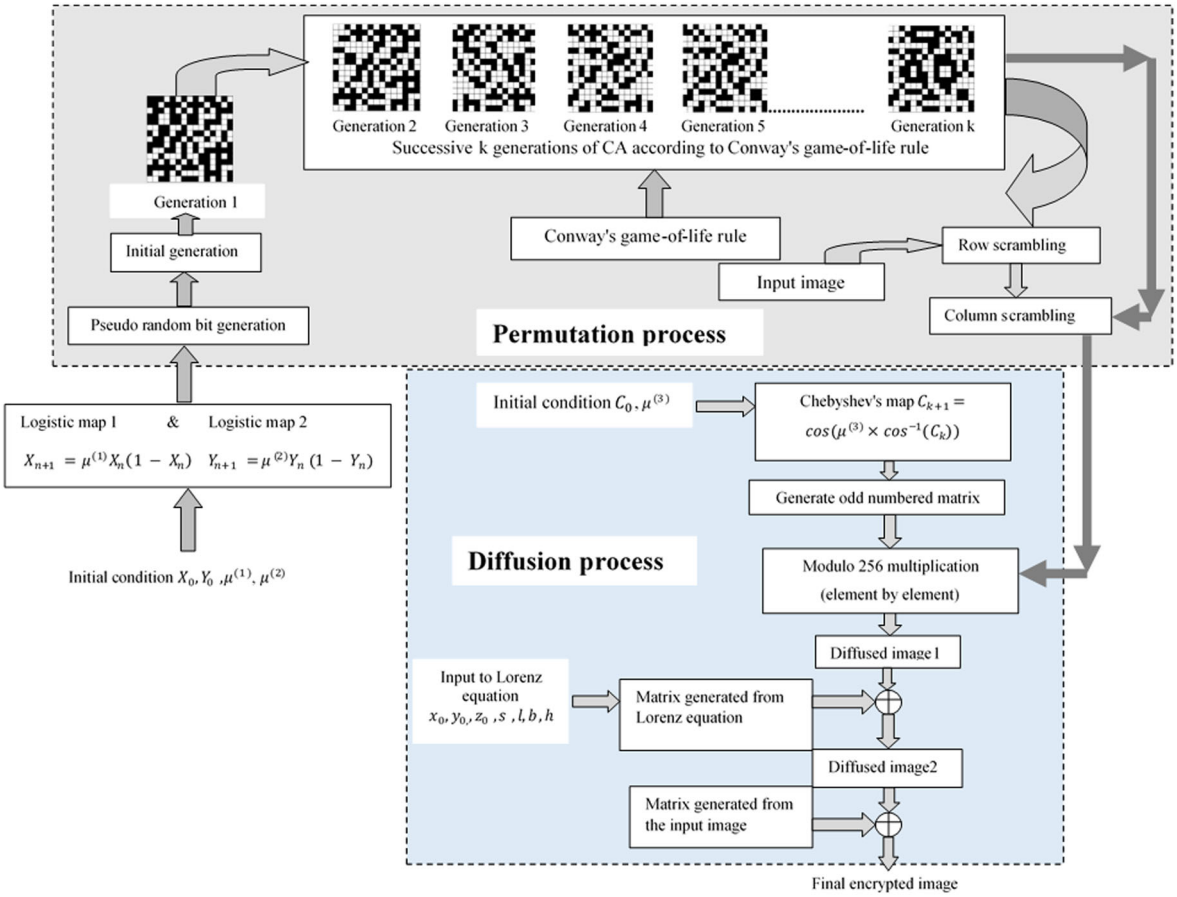

Figure 2. Flow chart of the proposed algorithm. CA, cellular automata.

$$Z_0[X_{n+1},,Y_{n+1}] = \begin{cases} 1 \ if \ X_{n+1} > Y_{n+1} \\ 0 \ if \ X_{n+1} \leq Y_{n+1} \end{cases} \quad (9)$$

The generation of this matrix is performed in row-first order.

Step 4 Set up the $M \times N$ Conway's game-of-life 2D CA with initial configuration Z_0. This is set to run up to k generations to obtain $\{Z_0, Z_1, \ldots \ldots, Z_k\}$ where the cells of next state are determined by the rules of Conway's game of life [32]. Periodic Moore neighborhood is used in the proposed algorithm (i.e., all eight neighbors of a cell are considered for the generation of the next state).

Step 5 Complete the row scrambling process [32], by doing the following three steps, after setting $row = 1$ and $col = 1$:

(i) For all i, j such that $Z_0(i, j) = 1$, place $I(row, col)$ in a new matrix $R(i, j)$ and increment (row, col) so that it points to the next pixel in the input image in row-first order.

(ii) For $q = 1, 2, \ldots \ldots k$ and for all i, j such that $Z_q(i, j) = 1$ and $Z_n(i, j) = 0$ $(for\ n = 1 \ldots q - 1)$, obtain the gray value of $I(row, col)$ and place it in $R(i, j)$ and increment (row, col) to point to the next pixel.

(iii) Obtain the gray value of the remaining pixels in I and place them in row-first order in those $R(i, j)$ for all $q = 1, 2, 3 \ldots \ldots k$ and $Z_q(i, j) = 0$.

Step 6 Complete the column scrambling process, by doing the following three steps, after setting $row = 1$ *and* $col = 1$:

(i) For all p such that $Z_0(i, j) = 1$, place $R(row, col)$ in another new matrix $O(i, j)$ and increment (row, col) so that it points to the next pixel in the input image in column-first order.

(ii) For $q = 1, 2, \ldots \ldots k$ and for all i, j such that $Z_q(i, j) = 1$ and $Z_n(i, j) = 0$ $(for\ n = 1 \ldots q - 1)$, obtain the gray value of $R(row, col)$ and place it in $O(i, j)$ and increment (row, col) to point to the next pixel.

(iii) Obtain the gray value of the remaining pixels in R and put them in column-first order in those $O(i, j)$ for all $q = 1, 2, 3 \ldots \ldots k$ and $Z_q(i, j) = 0$.

Figure 3 shows the generation of initial binary matrix and the successive eight generations for a sample 8×8

matrix. The two logistic maps act as pseudo-random number generators of the initial binary matrix Z_0. The successive generations Z_1, Z_2, …, Z_8 are obtained by applying CGLCA rules to the previous binary matrices. Figure 4 shows the row scrambling process described in step 5 for a sample 8×8 input matrix. The input matrix is first scrambled using the binary matrix Z_0. For better illustration, the input matrix of Figure 4(a) is shaded with the total number of alive cells (which correspond to 1) in the binary matrix Z_0 of Figure 4(b).

In the permuted matrix shown in Figure 4(c), the corresponding positions of the alive cells in the binary matrix are shaded and filled with the gray-level values of the input matrix in row-first order. Then, the first generation Z_1 is taken into account, and the corresponding positions of the alive cells in Z_1 are filled with the remaining gray-level values of the input matrix in row-first order. This process is repeated until all the generations are utilized and filled with the gray-level values of the input matrix. These stages of permutation are shown in Figure 4 (d)–4(j). After k generations, the left-out values of the input matrix are filled in the positions of the dead cells (which correspond to 0) in row-first order, as shown in Figure 4(k). Figure 5 shows the resultant images after the row scrambling and column scrambling processes for various test images.

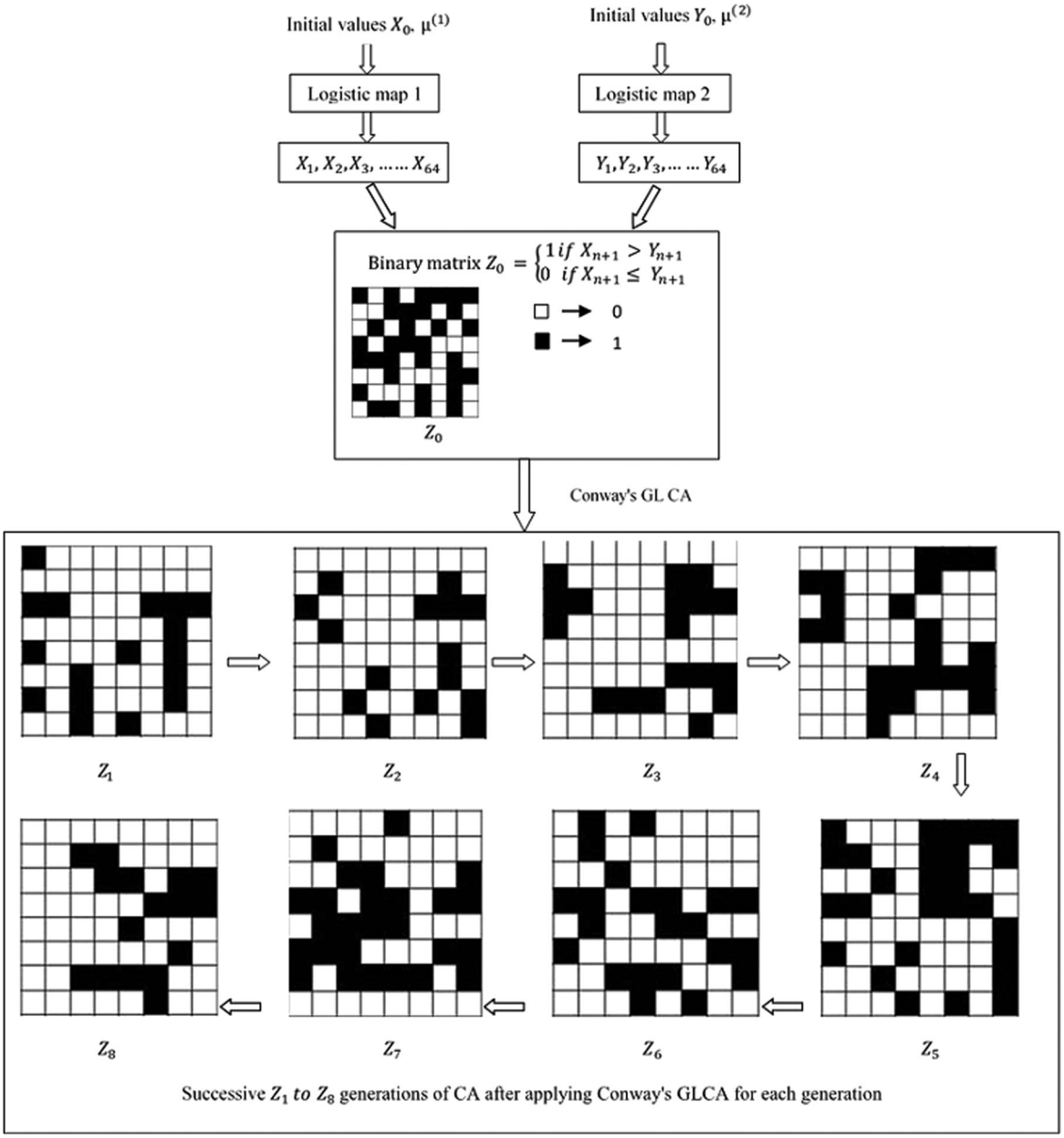

Figure 3. Generation of initial binary matrix and the successive eight generations using game-of-life cellular automata (GLCA).

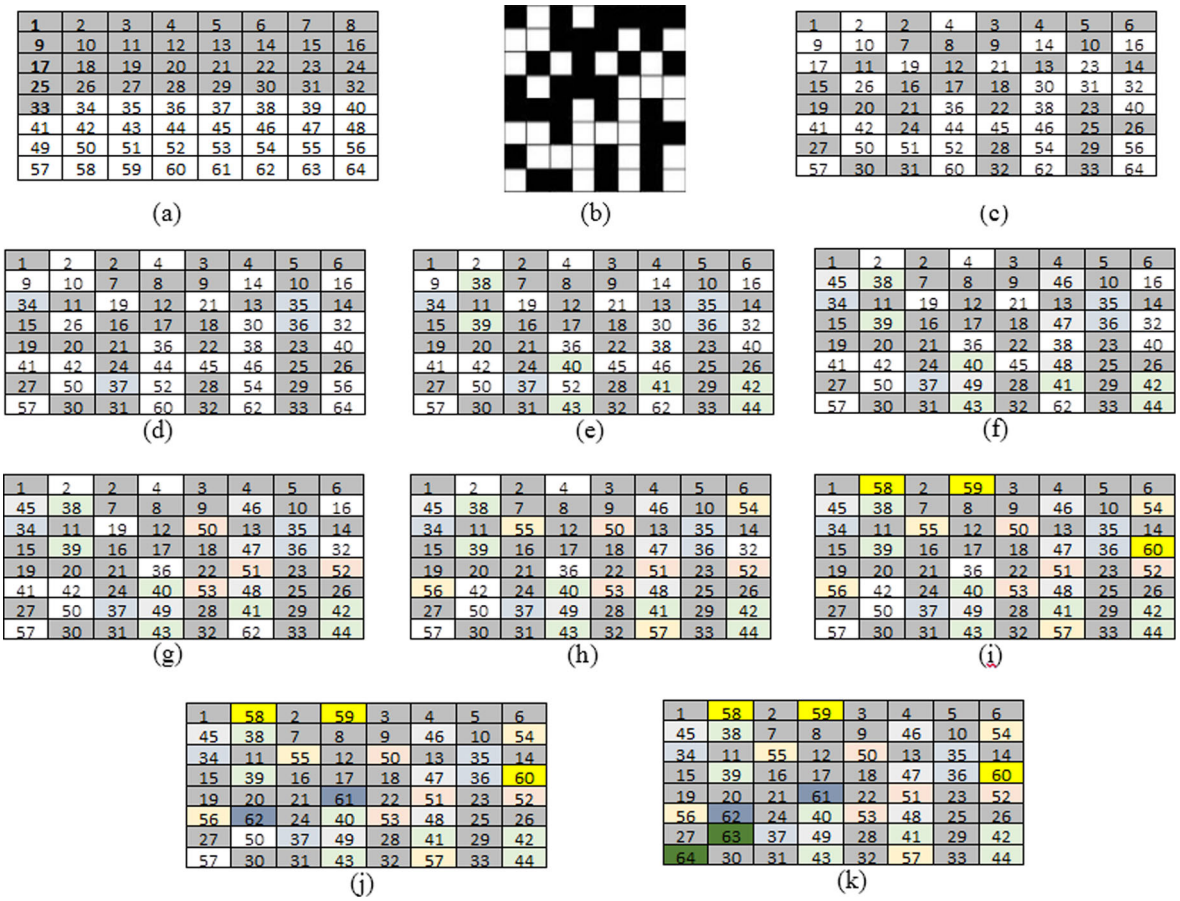

Figure 4. Illustration of row scrambling with an example: (a). 8 × 8 Input matrix (b). Binary matrix Z_0 (c). Permuted matrix using Z_0 (d)-(f). Row scrambling using 8 generations for the sample input matrix.

4.2. Diffusion process

The following steps are involved in the diffusion process:

Step 1 Generate the odd-numbered vector using the following procedure:

Set $i = 1$ and k = 0, iterate until $i = M \times N + 1$

$$k = k + 1$$

if (mod (fix $(256 \times C_{k+1})$, 2) $\neq 0$ and $(256 \times C_{k+1}) > 0$ // where C_{k+1} is computed using Chebyshev map.

$$zz\,(i) = fix\,(256 \times C_{k+1})$$

where $mod(a, b)$ gives the remainder obtained when "a is divided by b" and$fix(a)$rounds up "a" towards 0.It is to be noted that even numbers are discarded because inverse of multiplicative modulo 256 exists only for odd numbers as all odd numbers are relatively prime to 256. Negative numbers are discarded for obvious reasons.

Step 2 Convert the odd-numbered vector to an $M \times N$ matrix J in row-first order.

Step 3 Apply modulo 256 multiplication to multiply O and J (element by element)

$K = (O \times J) mod$ 256 and O is the diffused image1.

Step 4 With the initial conditions x_0, y_0, z_0, constants s, l, b, and the step size h, the Lorenz equation is solved by using Euler's method.

Step 5 At every iteration, the values of x, y, z are magnified to an appropriate natural number, $k \epsilon \mathbb{N}$, by performing $k_i = floor\left(m_i \times 10^{15}\right)$, $0 \leq i < MN$, $where$, $i \epsilon \mathbb{W}$, m where x, m is an array whose indices range from 0 to $MN - 1$, and the elements in m are calculated as $m_{3j} = x_j$, $m_{3j+1} = y_j$, $m_{3j+2} = z_j$, $3j + 2 < MN$, $j \geq 0$, $j \epsilon \mathbb{W}$ and F is the highest possible gray-level value in the image format.

Step 6 $p_i = k_i \, mod(F + 1)$, $0 \leq i < MN$, $i \epsilon \mathbb{W}$ is calculated. By using the $p$values found here, a $1 \times MN$ matrix P with MN elements is generated.

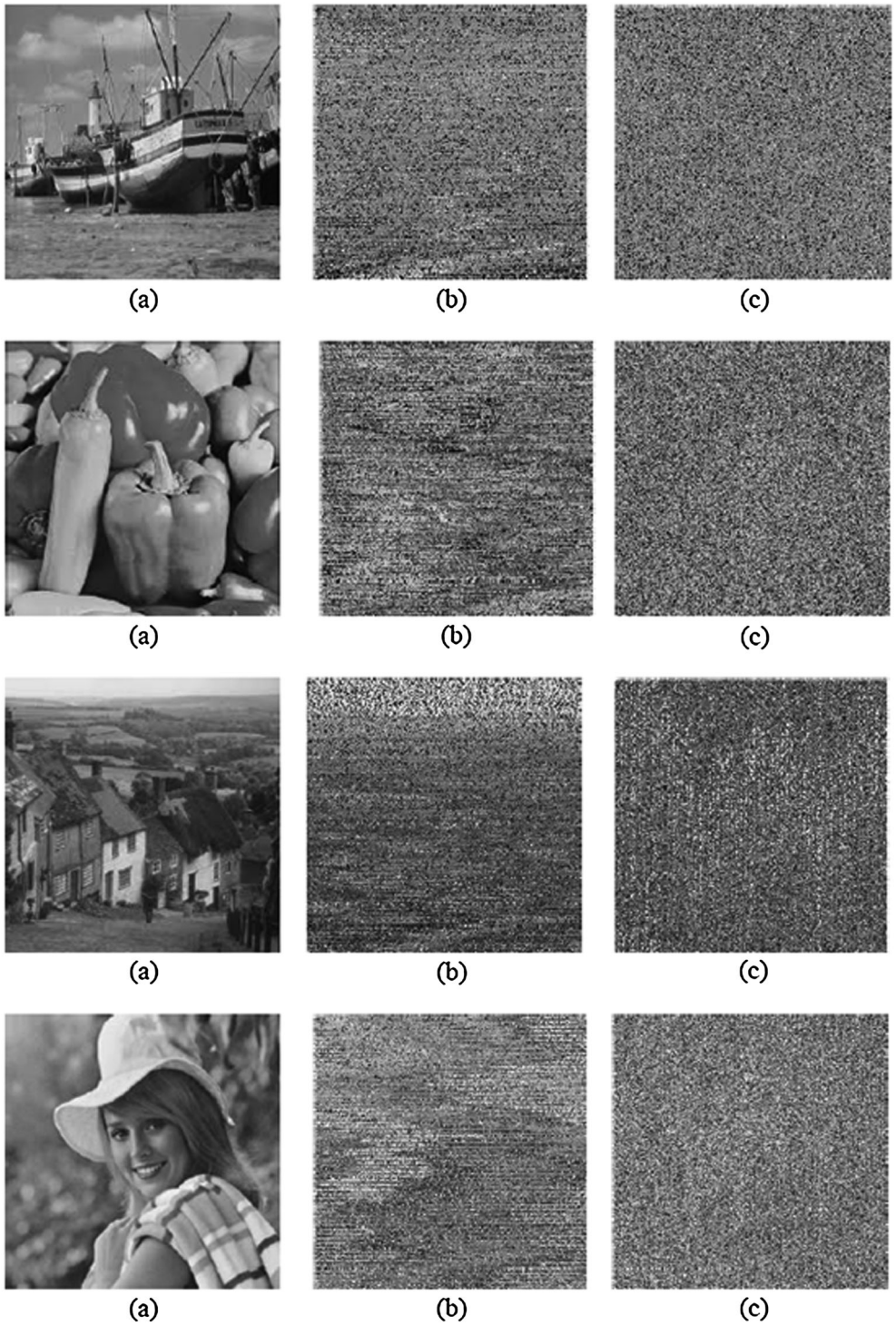

Figure 5. Original image and the resultant images after scrambling: (a) original image, (b) resultant image after row scrambling, and (c) resultant image after column scrambling.

Step 7 By taking N elements at a time from P, the matrix P is then used to generate an $M \times N$ matrix G.

Step 8 The original image I is converted into a $1 \times MN$ matrix T_1 by reversing the process used to generate the matrix G in the last step.

Then from T_1, another $1 \times MN$ matrix T_2, is constructed as

$$\begin{aligned} T2(n) &= T_1(1) \times 1 + T_1(2) \times 2 \\ &\quad + T_1(3) \times 3 \\ &\quad + \ldots\ldots\ldots\ldots \\ &\quad + T_1(n-1) \times (n-1) \\ &\quad + T_1(n) \times n, \\ &\text{where, } n = \{x | x \epsilon \mathbb{N},\ x < MN\} \end{aligned} \tag{10}$$

Step 9 Using $T_3 = T_2 \ mod \ (F+1)$, a matrix T_3 is obtained.

Step 10 Another $1 \times MN$ matrix T_4 is generated from T_3 using (11).

$$\begin{aligned} T_4(n) &= T_3(MN) \times 1 \\ &\quad + T_3(MN-1) \times 2 \\ &\quad + T_3(MN-2) \times 3 \\ &\quad + \ldots\ldots\ldots\ldots + T_3(MN \\ &\quad - (n-1)) \times n, \\ &\quad where,\ n = \{x | x \epsilon \mathbb{N},\ x < MN\} \end{aligned} \tag{11}$$

Step 11 Using T_4, matrix T_5 is calculated by $T_5 = T_4 \ mod \ (F+1)$. T_5 is then used to generate an $M \times N$ matrix $M \times N$ by following the process in step 5 of this diffusion process.

Step 12 Perform modulo 256 multiplication between O and J to obtain the $M \times N$ matrix K.

Step 13 An $(M \times N)$ matrix V also known as diffused image2 is generated by performing XOR operations on the elements of K and G.

Step 14 An $(M \times N)$ matrix E', which is the final encrypted image, is generated by performing XOR operations on the elements V and E.

Figure 6 shows the diffused image at various stages of the algorithm for different test images. The proposed algorithm is implemented in MATLAB that runs on a personal computer with an Intel i7 processor with 1 GB memory. Various analyses are performed using the observed data to prove the effectiveness of the proposed algorithm, and they are discussed in Section 5.

5. SECURITY ANALYSIS OF THE PROPOSED ALGORITHM

The efficiency and security of any image encryption scheme can be evaluated by a number of parameters. The proposed algorithm is analyzed using the metrics such as correlation of adjacent pixels and correlation between the plain and encrypted images (correlation coefficient). Further, key space analysis, key sensitivity analysis, and entropy analysis have also been performed on the proposed algorithm.

5.1. Correlation coefficient

Correlation is a measure of determining the degree of correspondence between two variables. It is very useful in determining the encryption quality of any cipher. A good image cipher produces highly uncorrelated cipher image by hiding the details of the plain image. If the plain and cipher images are totally different, the correlation coefficient between these two images is very low, and it is very close to zero. If the correlation coefficient is closer to 1, the plain and cipher images are said to be highly correlated. If the correlation coefficient is calculated between the plain image and with itself, then it results in 1, whereas if it is calculated between the plain image and its negative, then it results in -1.

5.1.1. Correlation of adjacent pixels

The correlation of adjacent pixels is tested using 12–14, by selecting 6000 samples of adjacent pixels randomly from the plain and encrypted image separately.

$$C_r = \frac{cov(x,y)}{\sqrt{D(x) \times D(y)}} \tag{12}$$

where

$$cov(x,y) = \frac{1}{T} \sum_{i=1}^{T} (x_i - E(x))(y_i - E(y))$$

$$E(t) = \frac{1}{T} \sum_{i=1}^{T} t_i \tag{13}$$

$$D(t) = \frac{1}{T} \sum_{i=1}^{T} (t - E(t))^2,\ t = x, y \tag{14}$$

The pair (x_i, y_i) denotes the two adjacent pixel values, $E(t)$ and $D(t)$ denote the mean and variance, respectively, and T is the total number adjacent pixel samples chosen. Mean and variance are calculated for all the values of x and y separately.

Table I gives the correlation values of the proposed scheme for some of the test images along all the directions. Correlation of original plain image is close to 1 because of higher redundancy. A cipher image is termed as a good one if it has a low correlation value ideally to zero along vertical, horizontal, and diagonal directions. It can be seen from this table that the cipher image has the correlation value very close to zero. A huge number of images with various sizes having different contents that belong to the library of standard images are taken as the test images. Table II gives the comparison of correlation coefficients of adjacent pixels of the proposed scheme with various existing schemes, which proves that the proposed algorithm scores over other algorithms.

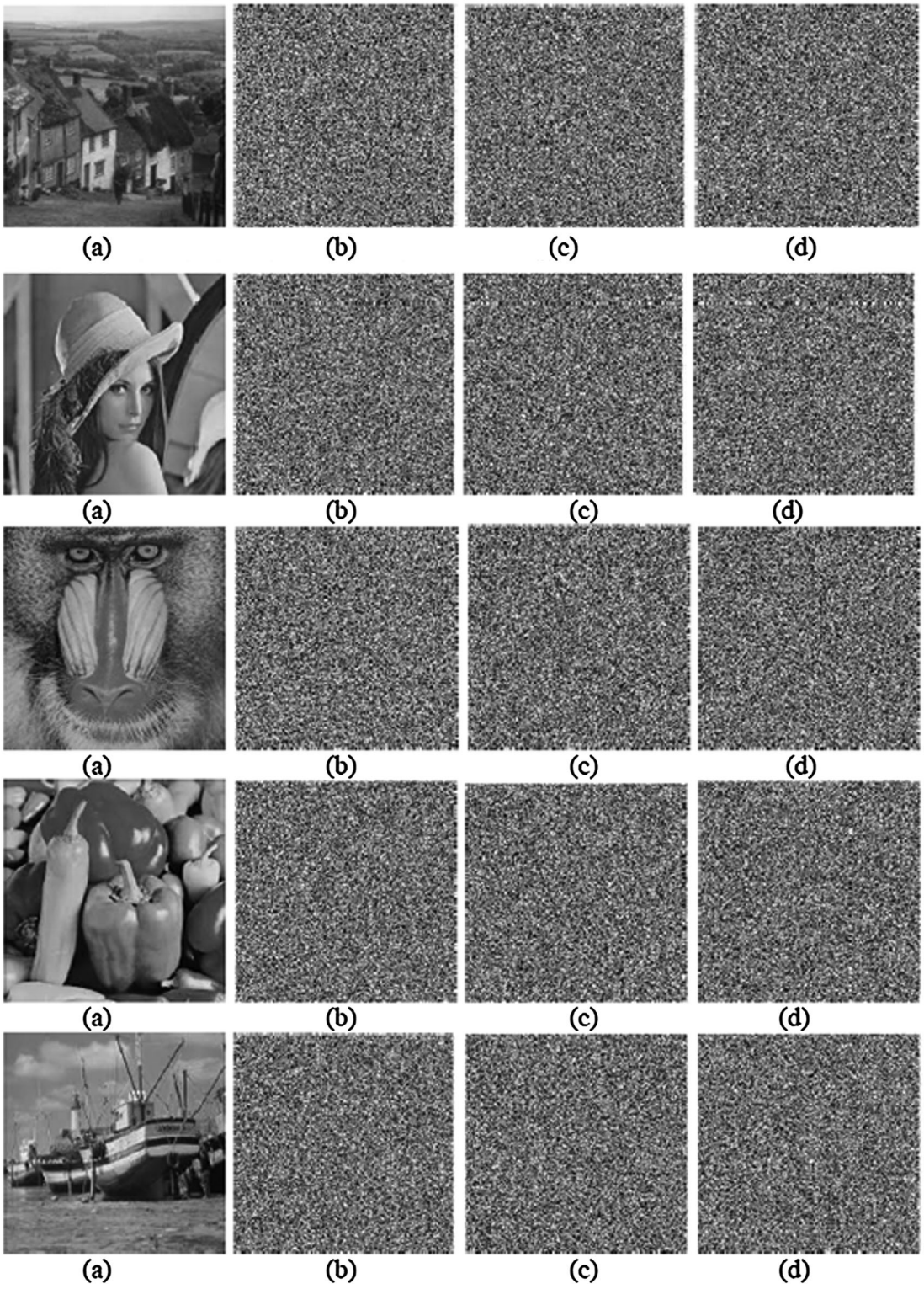

Figure 6. Original image and the diffused images: (a) original image, (b) diffused image after stage 1, (c) diffused image after stage 2, and (d) final encrypted image.

Security Comm. Networks 2016; **9**:634–651 © 2015 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

Table I. Correlation between adjacent pixels of various images.

Image name	Correlation values of plain image			Correlation values of cipher image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Aerial	0.9032	0.8443	0.8094	−0.0037	0.0025	0.0099
Airplane	0.9622	0.9381	0.9011	0.0069	−0.0006	0.0097
Baboon	0.8023	0.7661	0.7094	0.0068	−0.0070	0.0048
Cameraman	0.9232	0.9646	0.9093	0.0030	0.0014	−0.0042
Couple	0.9033	0.9082	0.8422	−0.0015	−0.0071	−0.0092
Elaine	0.9605	0.9682	0.9402	−0.0013	−0.0084	−0.0096
Lena	0.9231	0.9595	0.8890	−0.0019	−0.00023	−0.00013
Man	0.9153	0.9401	0.8799	−0.0086	−0.0040	0.0031
Moon	0.9345	0.9623	0.9299	−0.0042	0.0002	−0.0019
Zelda	0.9607	0.9794	0.9454	−0.00863	−0.0007	−0.0099

Table II. Correlation coefficients of adjacent pixels for Lena image by the proposed and various existing schemes.

Direction	Proposed algorithm	Algorithm [25]	Algorithm [26]	Algorithm [27]	Algorithm [28]
Horizontal	−0.0019	0.0031	0.00128	0.000508	−0.0054
Vertical	−0.00023	−0.0016	−0.00261	0.000050	0.0093
Diagonal	−0.00013	0.0067	0.00014	0.001699	−0.0009

5.1.2. Correlation between plain and encrypted images

An extensive study of the correlation among the original and its encrypted image produced by the proposed algorithm has been performed by calculating the correlation coefficients. This is performed using the formulae 15–17.

$$C_r = \frac{cov(x,y)}{\sqrt{D(x)\times D(y)}} \tag{15}$$

where

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x)(y_i - E(y))$$

$$E(t) = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{16}$$

$$D(t) = \frac{1}{N}\sum_{i=1}^{N}(t - E(t))^2 \ , \ t = x, y \tag{17}$$

where the pair (x_i, y_i) denotes the values of pixels in the image and $E(t)$ and $D(t)$ denote the mean and variance, respectively. Mean and variance are calculated for all x and y values separately, and N is the total number of pixels. Table III gives the correlation coefficient of the proposed scheme for various images. From this table, it is seen that the correlation coefficient is close to zero.

5.2. Key space analysis

The total number of keys used in the algorithm that are completely different from each other is called key space. Any encryption algorithm is said to be good if it has a larger key space. Larger key space makes certain attacks such as brute force attacks infeasible. The keys used in this algorithm are the initial conditions X_0 and Y_0, the parameters $\mu^{(1)}$ and $\mu^{(2)}$ of the logistic maps used for pseudo-random bit generation, the initial conditions x_0, y_0, z_0 and the step size h of the Lorenz equation, and finally the initial condition C_0 and the parameter $\mu^{(3)}$ of the Chebyshev map used for the diffusion process.

Based on the Institute of Electrical and Electronics Engineers floating point standard [33], 10^{-15} is the computational precision for a 64-bit double precision number. If all the initial conditions and the parameters used in the proposed algorithm have a precision of 10^{-15}, the key space of the proposed algorithm reaches 10^{150}, which is equivalent to 2^{498}. Considering that the fastest computer till date performs 1000 Million Instructions Per Second (MIPS), the computational load can be calculated as

Table III. Correlation coefficient of various images.

Images	Aerial	Airplane	Baboon	Cameraman	Couple	Elaine	Lena	Man	Moon	Zelda
Correlation coefficient	0.00267	−0.00209	0.00268	−0.00502	0.00112	−0.00452	−0.00798	0.00039	−0.00581	0.00650

$$\frac{2^{498}}{2^{80}\times 365\times 24\times 60\times 60}\cong 2.59\times 10^{130} \quad \text{years}$$

This key space is very large enough to resist various kind of brute force attacks.

5.3. Key sensitivity analysis

This is a method to determine the sensitivity of the encryption algorithm for any change in the key values. A good cipher image should be very much sensitive to the key used in the algorithm. Even modifying a single bit in the key should give a completely different encrypted image. The procedure for assessing key sensitivity is as follows:

(i) A plain Lena image of size 225×225 is encrypted using the secret key values listed in Table IV and is shown in Figure 7(a).
(ii) Then, a single bit difference in the same initial value $\mu^{(1)}=3.88031221556814$ is made, and it is used for the decryption, which is shown in Figure 7(b). The difference in image between Figure 7(a) and 7(b) is shown in Figure 7(c).
(iii) Among the entire length of secret keys, only the initial value of the logistic map1 is changed to $X_0=0.7895798803000001$ from the original value of $X_0=0.788998803$, and the other values remain the same. The secret keys along with the changed initial value of the logistic map1 are used to encrypt the same plain Lena image, and the resultant image is shown in Figure 7(d), and the difference in image between 7(a) and 7(d) is shown in Figure 7(e).
(iv) The same tests are repeated for the secret keys Y_0 and C_0 to obtain the decrypted images of Figure 7 (f) and 7(h) as well as the difference in images of Figure 7(g) and 7(i).
(v) The image is also decrypted with the correct key parameters and is given in Figure 7(j).

Figure 7(b), 7(d), 7(f), and 7(h) shows the decrypted images for a very small change in the original key values. From these figures, it is seen that the original image cannot be recovered unless the correct key value is substituted at the receiver side. This shows that the proposed algorithm is very much sensitive to the keys used in the algorithm so that even a small change in the keys used would not generate the correct plain image. The difference in images in Figure 7(c), 7(e), 7(g), and 7(i) shows that most of the pixels are different between the encrypted image with correct key values and the decrypted image with a slight change.

The procedure for calculating the sensitivity of the keys used in the algorithm is given as follows:(i) A set of initial conditions and constants are chosen for the various maps used in the proposed algorithm. (ii) By using the set of values, the $M\times N$ plain image I is converted to the cipher image S using the proposed algorithm. (iii) Then a small increment such as 10^{-15} is given to any one of the key values, and the encryption is performed on I again to obtain the new cipher image S_+ (increment). (iv) Similarly, a small decrement with the same magnitude is given to the chosen key value, and the encryption is again performed on the plain image I to obtain another cipher image S_- (decrement). (v) Then the sensitivity of the chosen key $P(k)$ is determined using the expression (18):

$$P(k)=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}N(S(i,j),\ S_-(i,j))+N(S(i,j),\ S_+(i,j))}{2\times M\times N}\times 100\% \tag{18}$$

where M and N are height and width of the plain image, respectively.

$$N(x,y)=\begin{cases}1\ , & if\ x\neq y\\ 0\ , & if\ x=y\end{cases}$$

The values of $P(k)$ for various key values are listed in Table IV. It is shown from Table IV that there is strong sensitivity to the change in initial key parameters and constants used in the proposed algorithm. A minor change in the plain image results in a drastic change in the cipher image.

Table IV. Sensitivity of various key values.

S.No.	Map	Key (k)	Value	Inc./dec.	$P(k)$, in %
1	Lorenz equation	x_0	7.78899e−10	10^{-15}	99.6017
2		y_0	1.23654e−10	10^{-15}	99.6122
3		z_0	1.23654e−10	10^{-15}	99.6115
4		h	0.01	10^{-15}	99.6119
5	Logistic map	$\mu^{(1)}$	3.98031221556814	10^{-15}	99.6207
6		X_0	0.7895798803	10^{-15}	99.6089
7		Y_0	0.69136	10^{-15}	99.6119
8		$\mu^{(2)}$	3.97345926782729	10^{-15}	99.6237
9	Chebyshev map	$\mu^{(3)}$	3.96535426376638	10^{-15}	99.6128
10		C_0	0.67854321	10^{-15}	99.6217

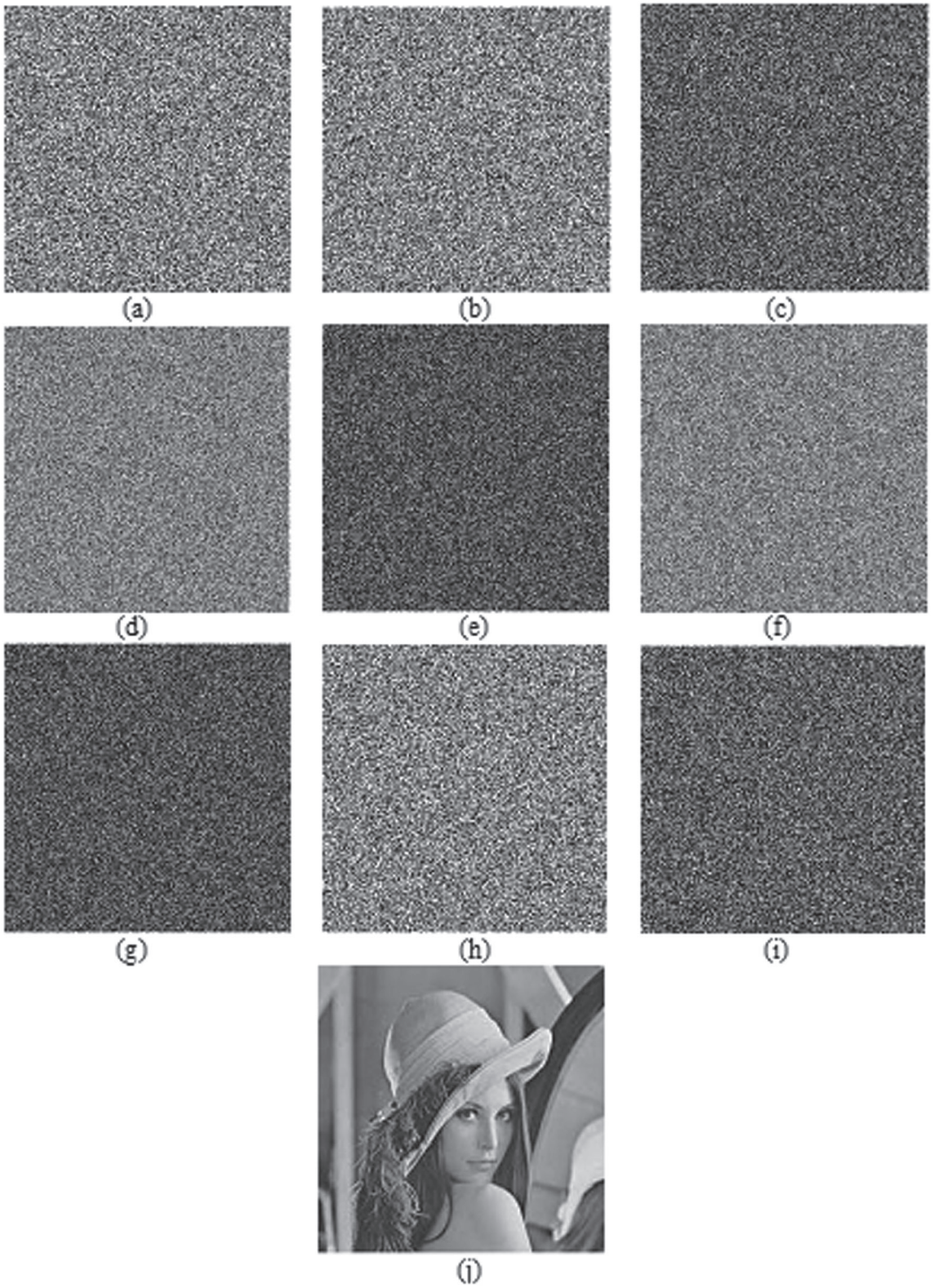

Figure 7. Key sensitivity test: (a). Encrypted image with correct key values (b). Decrypted image with the wrong value $\mu^{(1)}$ =3.88031221556814 (c). Difference image (a)-(b) (d). Decrypted image with wrong value X_0 = 0.7895798803000001 (e). Difference image (a)-(d) (f). Decrypted image with wrong value Y_0 = 0.69536 (g). Difference image (a)-(f) (h). Decrypted image with the wrong value C_0 = 0.078543 (i) Difference image (a)-(g) (j). Decrypted image with correct key values.

5.4. Information entropy analysis

The information entropy is a good measure of randomness. The entropy is determined using the following equation:

$$H(m) = -\sum_{i=0}^{F-1} p(m_i \times log_2(p(m_i)) \quad (19)$$

where F is the number of possible gray-level values for the image used and $p(m_i)$ gives the probability of the bit m_i. For 256 possible gray-level values, the value of F is 256. The information entropy is close to 8 if the proposed algorithm is a good one. The information entropy of various images for the proposed algorithm is given in Table V, and the comparison with various schemes is given in Table VI.

5.5. Diffusion characteristics of an image cipher

An attacker may make a minor change in the plain image and view the corresponding changes in the encrypted image. By doing so, the attacker is able to find out some important relationship between the plain and cipher images. A good image cipher should exhibit excellent diffusion characteristics such that a single bit change in the plain text should change the entire cipher image. Good diffusion is achieved by making the dependence between the plain input pixel values and the cipher output pixel values in a complex way. There are three methods, namely avalanche effect, number of pixels change rate (*NPCR*), and unified average changing intensity (*UACI*) to determine the influence of modifying a small amount of pixels in the original image.

5.5.1. Avalanche effect

A vey small change in the key or the original image should be able to produce a major change in the cipher image. This property is known as avalanche effect for any image cipher. Strict avalanche effect states that a very small change in the pixels or bits of the original image causes more than 50% of the pixel or bit changes in the cipher image. Mean squared error (*MSE*) is an important measurement for checking the avalanche effect by which the squared distance between the two images is determined. Let I and I' be two encrypted images in which the key or pixel is changed by a single bit. Then *MSE* can be calculated by using the following formula:

$$MSE = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[I(m,n) - I'(m,n)\right]^2}{M \times N} \quad (20)$$

where (m, n) denote the coordinates of the pixel values and $M \times N$ is the size of the image under consideration. The value of *MSE* is calculated to be 7.7759e+003, which is equivalent to 38.9075 dB. The results show that the proposed algorithm satisfies strict avalanche effect because $MSE > 30\ dB$ [34].

5.5.2. NPCR

NPCR determines the percentage of number of pixels changed between two cipher images. The *NPCR* values are calculated using (21) for the proposed algorithm for various images, and they are given in Table VII.

$$NPCR = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} D(i,j)}{M \times N} \times 100\% \quad (21)$$

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

C_1 and C_2 are the two cipher images, respectively, and M & N are the height and the width of the images, respectively.

5.5.3. UACI

UACI determines the difference of average change in pixel intensities between the two ciphered images. The

Table V. Entropy of various images.

Images	Aerial	Airplane	Baboon	Cameraman	Couple	Elaine	Lena	Man	Moon	Zelda
Entropy	7.9974	7.9975	7.9977	7.9975	7.9972	7.9988	7.9972	7.9956	7.9951	7.9952

Table VI. Information entropy for Lena image by the proposed and various schemes.

Algorithm	Proposed algorithm	Algorithm [25]	Algorithm [26]	Algorithm [28]
Entropy	7.9972	7.9952	7.9965	7.9978

Table VII. *NPCR* and *UACI* values for various images.

Images	Aerial	Airplane	Baboon	Cameraman	Couple	Elaine	Lena	Man	Moon	Zelda
NPCR	99.6025	99.6069	99.6010	99.5975	99.6094	99.6050	99.6124	99.5812	99.5910	99.6010
UACI	33.4169	33.4868	33.4491	33.4470	33.4027	33.4921	33.4468	33.4369	33.4860	33.4491

UACI values are calculated using (22) for the proposed algorithm for various images, and they are given together with *NPCR* values in Table VII.

$$UACI = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}|c_1\ (i,j) - c_2(i,j)|}{255\times M\times N}\times 100\% \qquad (22)$$

where C_1 and $O = GI$ are the two cipher images, respectively, and M & $N$are the height and the width of the images, respectively. From Table VII, it is seen that the proposed algorithm has good *NPCR* and *UACI* values. Table VIII shows the *NPCR* and *UACI* values for the Lena image by the proposed scheme in comparison with other existing schemes.

5.6. Encryption quality measures

The quality of the image encryption algorithm is an important measure and needs evaluation. Various measures such as maximum deviation, irregular deviation, and deviation from uniform histogram are proposed. The quality of the encryption algorithm is said to be good if the deviation of pixels between the plain and the cipher images is maximum and also irregular.

5.6.1. Maximum deviation

Maximum deviation [34] is calculated between the plain and the corresponding cipher image. The steps are as follows: (i) Obtain the histogram of both the plain and the cipher image. (ii) Calculate the absolute difference d between the aforementioned histograms. (iii) The sum of deviation is given in the following equation:

$$D = \frac{d_0 + d_{255}}{2} + \sum_{i=1}^{254} d_i \qquad (23)$$

where d_i is the value of the absolute difference at the index i. For the test Lena image of size 225×225, the value of maximum deviation is found to be 47 706, which confirms the high quality of the image encryption algorithm.

5.6.2. Irregular deviation

Irregular deviation [34] is calculated in order to know how uniform the input pixel values are distributed. From this measure, the deviation of the input pixel values to either larger value or a smaller value from their original initial values can be measured. The statistical distribution of the histogram deviation is compared with uniform distribution to measure the closeness between them. If it is found closer, the encryption algorithm is said to be good. The steps ((i)–(v)) listed in the following and the equations 24–28 are used to calculate the irregular deviation.

(i) Calculate the absolute difference D between the plain text and the cipher text as

$$D = abs(P - C) \qquad (24)$$

(ii) Calculate the histogram of D as

$$h = hist(D) \qquad (25)$$

(iii) The average of h is calculated as

$$ave_h = \frac{1}{256}\sum_{i=0}^{255} h_i \qquad (26)$$

where h_i is the amplitude at index i.

(iv) Calculate the absolute histogram deviation from the average value as

$$h_{D_i} = abs(h_i - ave_h) \qquad (27)$$

(v) Calculate the irregular deviation as

$$I_D = \sum_{i=0}^{255} h_{D_i} \qquad (28)$$

For the test Lena image of size 225×225, the value of irregular deviation is found to be 9209, which indicates the effectiveness of the encryption algorithm.

5.6.3. Peak signal-to-noise ratio (*PSNR*)

PSNR is used to measure the changes in the pixel values between the plain and the encrypted images. Let I and E' be the plain and encrypted image, respectively. Then,

$$PSNR = \frac{M\times N\times 255^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}[I(m,n) - E'(m,n)]^2} \qquad (29)$$

where (m,n) gives the coordinates of the pixel value and M & N are the height and width of the image, respectively. The value of *PSNR* is calculated as 9.223 dB, which is a lower value and indicates the better encryption quality.

Table VIII. Comparison of *NPCR* and *UACI* values for Lena image of the proposed scheme with other existing schemes.

Algorithm	Proposed algorithm	Algorithm [18]	Algorithm [22]	Algorithm [25]	Algorithm [27]
NPCR	99.6124	93.6768	99.5864	96.0000	99.6836
UACI	33.4468	33.3364	33.2533	32.5900	33.4647

5.6.4. Entropy quality

The quality of encryption can be determined by the total number of changes in the pixel values between the plain and its corresponding encrypted images. Let I and E' represent the plain and the corresponding encrypted images, respectively, with a size of $M \times N$ and F gray-level values. $H_F(I)$ and $H_F(E')$ represent the total number of occurrences of each possible gray-level value F in the plain and encrypted images, respectively. The entropy quality is expressed using the formula

$$\text{Entropy quality} = \frac{\sum_F^{255} \left| H_F(I) - H_F\left(E^{'}\right) \right|}{256} \tag{30}$$

The entropy quality for the proposed encryption scheme is calculated as 127 for the Lena image of size 225×225.

5.7. Image quality criterion

The quality of the image after decryption can be determined by the mean square error (*MSE*). Smaller value of *MSE* indicates that the quality of the image is not compromised during decryption process. Let I and I' be the plain and decrypted image, respectively. The *MSE* values can be determined as

$$MSE = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} \left[I(m,n) - I^{'}(m,n) \right]^2}{M \times N} \tag{31}$$

where (m, n) denotes the coordinates of the pixel values and $M \times N$ is the size of the image under consideration. $I(m, n)$ and $I'(m, n)$ are the plain and decrypted images, respectively. For more than 30 images, *MSE* was calculated, and in all the cases, the value is 0. Thus, it shows that the proposed image encryption produces lossless image cipher.

5.8. Statistical attack

Shannon in his paper [1] noted that there is the possibility to break any type of cryptosystem by means of statistical analysis. Therefore, it is important that any cryptosystem should be able to pass the statistical analysis test. The following statistical tests are performed in order to prove that the proposed scheme is able to withstand the statistical attack.

The histograms of the original Lena image (Figure 8(a)) and the encrypted image (Figure 8(c)) are shown in Figure 8(b) and 8(d), respectively. From these figures, it is clear that the histogram of the encrypted image is nearly flat and it shows that it is significantly different from that

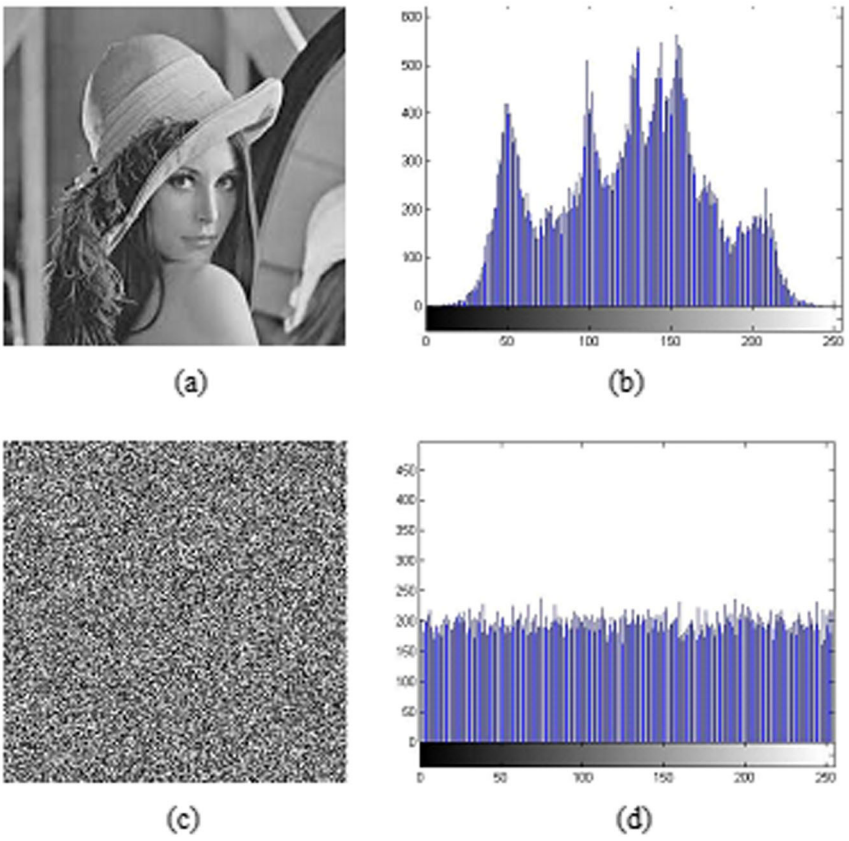

Figure 8. Histogram analysis for Lena image: (a). The original image (b). Histogram of the original image (c). The encrypted image (d). Histogram of the encrypted image.

of the plain image. So, there is no useful information that can be retrieved from the histogram. The chi-square value of the cipher image can also be computed in order to show that the histogram of the cipher image is uniform. The value can be determined using the following formula:

$$\chi^2 = \sum (V_k - 1024)^2 / 1024 \tag{32}$$

where k is the total number of possible gray-level values (0–255) and V_k is the occurrence frequency of each possible gray-level value that is observed, the expected occurrence frequency being 1024. For a significant level of 0.05, $\chi^2(255,\ 0.05) = 293$. For the proposed algorithm, the chi-square value of the final encrypted image is 257. This shows that the histogram distribution of the proposed algorithm is uniform, $\chi^2_{test} < \chi^2(255, 0.05)$.

5.9. Encryption speed

The proposed algorithm has been run on a personal computer with an Intel i7 processor with 1 GB memory. The encryption speed is measured for various-sized images and compared with various algorithms in Table IX. From the table, it is seen that the encryption time of the proposed scheme is less compared with existing schemes. It is because CA is used in the permutation process of the scheme.

5.10. Chosen plaintext attack/known plaintext attack

By means of cryptanalysis, an attacker uses various levels of attacks such as Ciphertext only, Known plaintext, Chosen plaintext, and Chosen ciphertext to break any kind of cryptosystem. Kerckhoffs principle [35] has an assumption that the attacker knows everything about the encryption architecture except the secret key. Further, in Chosen plaintext attack, the attacker obtains access to the encryption architecture and obtains the plaintext and its corresponding ciphertext. Mostly, all of these chaos-based ciphers [36–38] are attacked by exposing the intermediate parameters using the obtained pairs rather than disclosing the actual key parameters.

5.10.1. Extracting parameters of permutation process

The proposed cipher comes under chaotic linear cipher. In the permutation process of the proposed cipher, the initial generation of the CA is obtained from the output of the two logistic maps. The next generation is obtained by applying CGLCA rules to the initial generation. Similarly, successive k generations are obtained. Using these k generations, the plain image undergoes k stages of row scrambling process and k stages of column scrambling process to obtain the final permuted image.

The permutation process of the proposed cipher is represented as

$$0 = GI \tag{33}$$

where G is the transformation matrix obtained from CA and needs to be extracted by an attacker. This equation may look simple for an attacker to obtain G. But actually in real situation, the process of obtaining G is very much challenging because it actually has k stages of permutation for row scrambling and k stages of permutation for column scrambling. For instance, the k stages of row scrambling process are represented as

$$R_1 = t_1 I;\ \ R_2 = t_2 R_1;\ \ R_3 = t_3 R_2; \ldots\ldots\ R = R_k = t_k R_{k-1}; \tag{34}$$

where $t_1, t_2, t_3, \ldots \ldots, t_k$ are the transformation matrices obtained from CGLCA.

Similarly, for column scrambling process,

$$O_1 = t_1 R;\ \ O_2 = t_2 O_1;\ \ O_3 = t_3 O_2; \ldots\ O = O_k = t_k O_{k-1}; \tag{35}$$

Hence, to obtain G, the attacker needs to find all the transformation matrices $t_1, t_2, t_3, \ldots \ldots, t_k$ of the k stages, which in turn shows how difficult it is to obtain G.

5.10.2. Extracting parameters of diffusion process

The proposed cipher has three levels of diffusion. In the first level, the matrix J generated from the Chebyshev map is used to obtain the first level of diffused image K.

$$K = J \oplus O \tag{36}$$

Table IX. Encryption time (in seconds) for Lena image by the proposed and various schemes.

Size of the image	Time in sec.		
	Proposed algorithm	AES	Algorithm [25]
256 × 256	0.12	0.46	0.16
512 × 512	0.86	1.7	0.92
1024 × 1024	4.98	6.55	5.65

In the next level, the matrix G generated from the Lorenz equation is used to obtain the diffused image.

$$V = G \oplus K \tag{37}$$

In the third level, the matrix E generated from the plain image is used to obtain the final encrypted image.

$$E^{'} = E \oplus V \tag{38}$$

From the previous equation, it can be seen that the encrypted value at every point is not only dependent on the matrix J and matrix G (keys) but also on the plain image matrix E. Hence, the attacker cannot acquire any valuable information from the cryptosystem because the resultant image highly depends on the image chosen that makes the chosen plain text attack infeasible.

6. CONCLUSION

A new image encryption scheme using the combination of chaos and CA has been proposed in this paper. The proposed algorithm uses the logistic map as a pseudo-random bit generator and CGLCA for permutation of the image. Chebyshev map and Lorenz equation are used for diffusion. The introduction of an additional matrix, which is generated by the weighted sequence of the plain image pixel values, has increased the sensitivity of the encrypted image to the input image. It is proved that the proposed algorithm is a very good resistant to statistical attack as well as differential attack; it also has very less correlation among adjacent pixels, high entropy, good *NPCR*, and *UACI* values. The high sensitivity of the algorithm to the key space is an added advantage in view of the large key space of the proposed algorithm (10^{150}). However, the speed of the algorithm is not compromised, as CA has been employed for permutation. This shows that the proposed algorithm is extremely secure.

REFERENCES

1. Shannon CE. Communication theory of secrecy system. *Bell System Technical Journal* 1949; **28**:656–715.
2. Uhl A, Pommer A. *Image and Video Encryption: From Digital Rights Management to Secured Personal Communication*. Springer: New York, 2005.
3. Li S, Chen G, Cheung A, Bhargava B, Lo KT. On the design of perceptual MPEG video encryption algorithms. *IEEE Transactions on Circuits Systems for Video Technology* 2007; **17**:214–223.
4. Mathews R. On the derivation of a "chaotic" encryption algorithm. *Crpytologia* 1989; **13**:29–42.
5. Baptista MS. Cryptography with chaos. *Physics Letters* 1998; **240**:50–54.
6. Fridrich J. Image encryption based on chaotic maps. *IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulations* 1997; **2**:1105–1110.
7. Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. *International Journal Bifurcation and Chaos* 1998; **8**:1259–1284.
8. Kocarev L, Jaimovsvski G. *Chaos and Cryptography: From Chaotic Maps to Encryption Algorithms*. Springer: New York, 2007.
9. Gao H, Zhang Y, Liang S, Li D. A new chaotic algorithm for image encryption. *Chaos, Solitons and Fractals* 2006; **29**:393–399.
10. Kocarev L, Lian S. *Chaos-Based Cryptography Theory, Algorithms and Applications*. Springer: New York, 2011.
11. Zhang XH, Liu F, Jiao LC. An encryption arithmetic based on chaotic sequence. *Image and Graphics* 2003; **8**:374–378.
12. Neto LG, Sheng YL. Optical implementation of image encryption using random phase encoding. *Optical Engineering* 1996; **35**:2459–2463.
13. Jakimovski G, Kocarev L. Chaos and cryptography: block encryption ciphers based on chaotic maps. *IEEE Trans. on Circuits and Systems* 2001; **48**:163–169.
14. Chuang TJ, Lin JC. A new multi-resolution approach to still image encryption. *Pattern Recognition and Image Analysis* 1999; **9**:431–436.
15. Li CG, Han ZZ, Zhang HR. An image encryption algorithm based on random key and quasi-standard map. *Chinese Journal of Computers* 2003; **26**:465–470.
16. Von Neumann J. *Theory of Self-Reproducing Automata*. University of Illinois Press: Champaign, IL, 1966.
17. Burks AW. *Von Neumann's Self-Reproducing Automata*. University of Illinois Press: Champaign, IL, 1970.
18. Teng L, Wang X. A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive. *Optics Communications* 2012; **285**: 4048–4054.
19. Wang X, Jin C. Image encryption using game of life permutation and PWLCM chaotic system. *Optics Communications* 2012; **285**:412–417.
20. Fu C, bin Lin B, sheng Miao Y, Liu X, jie Chen V. A novel chaos-based bit-level permutation scheme for digital image encryption. *Optics Communications* 2011; **284**:5415–5423.
21. Xiao D, Shih FY. Using the self-synchronizing method to improve security of the multi chaotic systems-based image encryption. *Optics Communications* 2010; **283**:3030–3036.
22. Wang X, Liu L, Zhang X. A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Optics and Lasers in Engineering* 2015; **66**:10–18.

23. Chen J, Zhu Z, Fu C, Yu H, Zhang L. A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. *Communications in Nonlinear Science and Numerical Simulation* 2015; **20**:846–860.
24. Boriga R, Dăscălescu AC, Priescu I. A new hyper chaotic map and its application in an image encryption scheme. *Signal Processing: Image Communication* 2014; **29**:887–901.
25. Pareek NK, Patidar V, Sud K. Diffusion-substitution based gray image encryption scheme. *Digital Signal Processing* 2013; **23**:894–901.
26. Sun F, Liu ZLS. A new cryptosystem based on spatial chaotic system. *Optics Communications* 2010; **283**: 2066–2073.
27. Jianhua L, Hui L. Colour image encryption based on advanced encryption standard algorithm with two-dimensional chaotic map. *IET Information Security* 2013; **7**:265–270.
28. Abdul HA, Rasul E, Malrey L. A hybrid genetic algorithm and chaotic function model for image encryption. *International Journal of Electronics and Communications (AEU)* 2012; **66**:806–816.
29. Patidar V, Sud KK. A novel pseudo random bit generator based on chaotic standard map and its testing. *Electronic Journal of Theoretical Physics EJTP* 2009; **6**:327–344.
30. Stogartz SH. *Non Linear Dynamics and Chaos*. Addison-Wesley, 1994.
31. Lorenz EN. *The Essence of Chaos*. University of Washington Press: Seattle, WA, 1993.
32. Dalhoum ALA, Mahafzah BA, Awwad AA. Digital image scrambling using 2D cellular automata. *IEEE Multimedia* 2012; **19**:28–36.
33. IEEE Computer Society. *IEEE standard for binary Floating Point Arithmetic*ANSI/IEEE std 1985.
34. Ahmad J, Ahmed F. Efficiency analysis and security evaluation of image encryption schemes. *International Journal of Video & Image Processing and Network security* 2012; **12**:18–31.
35. Menezes AJ, van Oorschot PC. *Vanstone SA*. CRC Press: Handbook of Applied Cryptography, 1997.
36. Rhouma R, Belghith S. Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Physics Letters A* 2008; **372**:5973–5978.
37. Çokal C, Solak E. Cryptanalysis of a chaos-based image encryption algorithm. *Physics Letters A* 2009; **373**:1357–1360.
38. Arroyo D, Li C, Li S, Alvarez G, Halang W. Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. *Chaos, Solitons and Fractals* 2009; **41**:2613–2616.