



# A modified method for image encryption based on chaotic map and genetic algorithm

Mahdieh Ghazvini<sup>1</sup>  · Mojdeh Mirzadi<sup>1</sup> · Negin Parvar<sup>2</sup>

Received: 23 May 2019 / Revised: 1 May 2020 / Accepted: 8 May 2020 /

Published online: 20 June 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

The security of digital data has been attending more than past, spatially image data security. In this study, a hybrid image encryption method has been proposed based on genetic algorithm and chaos. Encryption process consists of three main steps: confusion phase, diffusion phase, and improvement phase using a genetic algorithm. At first, Chen's chaotic map is used in the confusion phase to generate a scrambled image by shuffling plain-image pixels, and in the diffusion step, Logistic-Sine map alters those pixels gray-level values. It produces some of encrypted images which were considered as the initial population for the genetic algorithm. Then, by using the genetic algorithm, the encrypted images are optimized as much as possible. Finally, the best encrypted image is the final cipher image. The experimental results and several security analyses show that the proposed modified method provides an efficient scheme for image encryption and good robustness against frequent statistical and security attacks.

**Keywords** Image encryption · Chaotic map · Security · Genetic algorithm

## 1 Introduction

Nowadays, with the rapid development in digital technology and communication networks, digital data and information are store and transmit more and more, over the networks. Thus,

---

✉ Mahdieh Ghazvini  
mghazvini@uk.ac.ir

Mojdeh Mirzadi  
m.mirzadi@gmail.com

Negin Parvar  
n\_parvar@yahoo.com

<sup>1</sup> Department of Computer Engineering, Faculty of Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

<sup>2</sup> Department of Computer Engineering, Islamic Azad University, Kerman Branch, Kerman, Iran

developing techniques for maintaining the security of storing and transmitting digital data has been more and more interested. Digital images and videos carrying a large amount of essential information for many applications, thus image data protection against unauthorized users has become an important issue for searches [19, 20]. One of the methods of protecting information is to use encryption algorithms. By image encryption techniques, the original image converted to an encrypted image that is not easy to understand.

On one hand, different kinds of image encryption methods based on different technologies have been presenting that, among them, chaos-based techniques are more interested in recent years. The chaotic encryption was characterized by high sensitivity to initial situations and parameters, pseudo-random behavior, non-periodicity, the simplicity of implementation in hardware & software, and mixing property. Therefore, it is widely attractive to image encryption [8, 18, 21, 37]. On the other hand, one of the most frequently used optimization algorithms is a genetic algorithm (GA), and it also has some applications in cryptography fields. To traverse the solution search space, GA uses selection, crossover, and mutation operators inspired by evolutionary biology. In this algorithm, each solution encoded as a string called a *chromosome*. The algorithm starts by initializing a population with random or default values. For the first time, Abdullah et al. [1] proposed an encryption method based on a hybrid genetic algorithm and chaotic function. They used the chaotic logistic map function to encrypt images and produce several encrypted images as the initial population for genetic algorithms. Then, by using the genetic algorithms, these encrypted images are optimized to choose the final encrypted image as the best cipher image with the low correlation coefficient between pixels and high entropy. In this study, a hybrid GA and chaos-based image encryption method proposed. In the proposed scheme, the encryption process includes three main confusion phases, diffusion phase, and applying the GA algorithm for improvement.

A list of abbreviations and acronyms used throughout the paper was given in Table 1. The rest of this paper was organized as follows. The preliminary and related work are described in sections 2 and 3, respectively. Section 4 describes the proposed method, and section 5 gives the simulation results and analysis. Finally, the paper concluded in section 6.

## 2 Preliminary

Here, we briefly describe chaotic systems like Chen and Logistic-Sine chaotic map and genetic algorithms, which are the basic components of the most chaotic based encryption methods as well as our proposed method.

### 2.1 Chaotic maps

Chaotic maps are subtypes of non-linear dynamical systems that are similar to the noise signal but very certain. The chaotic map employed to produce disordered sequence [15]. High sensitivity to initial value is one of the important characters of these systems. It means if a little change was made in the primary value, a sharp change can happen in the result, so this main characteristic becomes this system's robustness, effectiveness, and non-predictable that makes these suitable for encryption applications [8, 17, 31]. The Chaotic system was explained as follows:

**Table 1** List of acronyms and abbreviations

Acronyms/Abbreviations	Definition
CCP	Column Circular Permutation
CI	Cipher Image
D	Diagonal
DNA	deoxyribonucleic acid
GA	Genetic Algorithm
H	Horizontal
HD	High-Dimensional
IEA	Image Encryption Algorithm
ILS	Improved Logistic System
LS	Logistic-Sine
LSMCL	Logistic-Modulated-Sine-Coupling-Logistic chaotic map
NCPR	Number of Pixels Change Rate
PI	Plain Image
PSNR	Peak signal-to-noise ratio
RCP	Row Circular Permutation
SHA	Secure Hash Algorithms
UACI	Unified Average Changing Intensity
V	Vertical

$$X_{K+1} = f(X_K), : \rightarrow I = [0, 1], X \in I \quad (1)$$

Where  $X_0$  is the initial value, and the output is a sequence of  $n$  numbers between 0 and 1. The  $f$  system called the chaotic system when it is non-linear, deterministic, susceptible to initial situations, irregularity, and non-predictable.

## 2.2 Chen chaotic system

Chen system is a hyper chaotic system, and because of sophisticated mathematical equations and sequences generated by this system are randomized, and their prediction is difficult. The complexity and the large key space of hyper chaotic systems is the reason why these systems have higher resistance against all types of attacks [26]. The hyper chaotic Chen system is defined as follows:

$$\dot{x} = \begin{cases} a(y-z) \\ y \\ z \end{cases} = (c-a)x - xz + cyz = xy - bz \quad (2)$$

Where  $a$ ,  $b$ , and  $c$  are control parameters of the system. In the case of  $a = 35$ ,  $b = 3$ , and  $c = 28$ , the system behaves as a chaotic system [13, 23], and gives three random sequences by repeating the Chen system, as shown in Fig. 1.

## 2.3 Logistic-sine chaotic system

Logistic-Sine (LS) system is a nonlinear mixture of two different 1D chaotic Logistic and Sine systems [37]. The 1D chaotic systems like Logistic Sine, and Tent map have a lot of applications due to their natural structures. Still, they have some difficulties consist of the restricted range of chaotic behaviors or discontinuous range of these, the vulnerability to low-computation-cost analysis, and the non-uniform distribution of data [2, 34, 37]. The combined

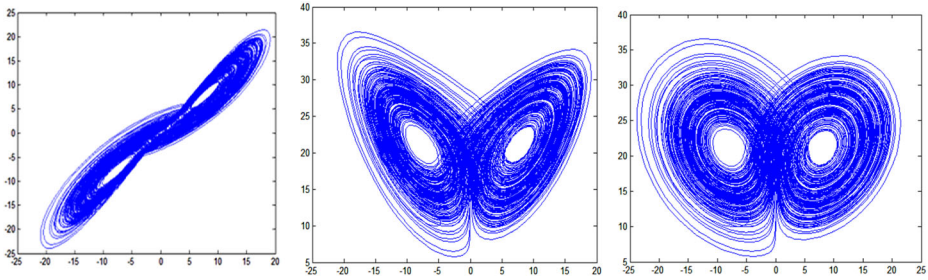


Fig. 1 Three generated sequences X, Y, and Z by hyper chaotic Chen system

LS system solves the above problems and has excellent chaotic properties like a wide range of parameters and uniform distribution of data [37]. Figure 2 illustrated the combination of Logistic and Sine chaotic systems.

Where  $L(r, X_n)$ , and  $S((4-r), X_n)$  are Logistic and Sine chaotic systems with parameter  $r \in (0, 4]$ , respectively, and  $X_n$  is the chaotic output sequence. Also,  $n$  is the iteration number, and  $mod$  is modulo operation that assures its output in  $[0,1]$ . LS system was described by the following equation [37]:

$$X_{n+1} = A_{LS}(r, X_n) = (L(r, X_n) + S((4-r), X_n)) \bmod 1 = (rX_n(1-X_n) + (4-r)\sin(\pi X_n)/4) \bmod 1 \quad (3)$$

Compared with its corresponding maps, Logistic, and Sine, LS has more complex, chaotic properties. When one of its maps is out of the chaotic range, this chaotic system can still have chaotic behaviors. The bifurcation diagram and Lyapunov Exponent of Logistic and LS systems are illustrated in Figs. 3 and 4, respectively. In the Logistic system, if  $r \in (0, 3.57]$  the system has non-chaotic behavior and its chaotic behavior is restricted only within  $[3.57, 4]$  where the chaotic behavior of the LS system exists within the whole range of  $(0, 4]$  [37]. The Lyapunov exponent acts as an indicator to measure the chaotic behavior of systems [14]. As illustrated in Fig. 4, the LS Lyapunov exponent closes to 0.7 in range  $r \in (0, 4]$ .

### 2.4 Genetic algorithm

One of the best algorithms used to solve a wide variety of optimization problems is the Genetic Algorithm (GA). The GA is a heuristic search algorithm inspired by Darwin’s theory of evolution. This algorithm shows the natural selection procedure in which the right individuals are selected for reproduction to produce offspring. GA acts on a population of some solutions where the population size is the number of solutions. Each solution is called an individual solution. Every single one of these individual solutions has a chromosome. The chromosome has been presented as a set of parameters that characterize the individual. Each chromosome has a set of genes. Each gene is as a string of 1 s and 0 s as illustrated in Fig. 5.

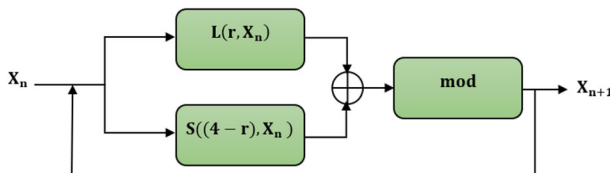
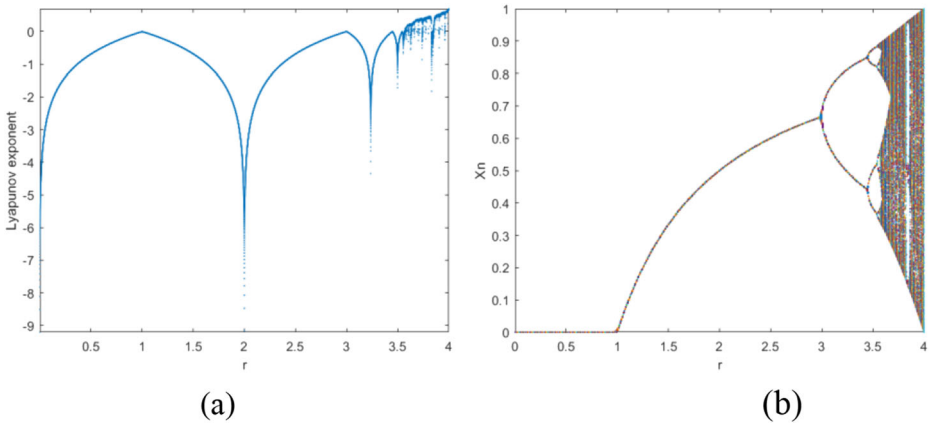
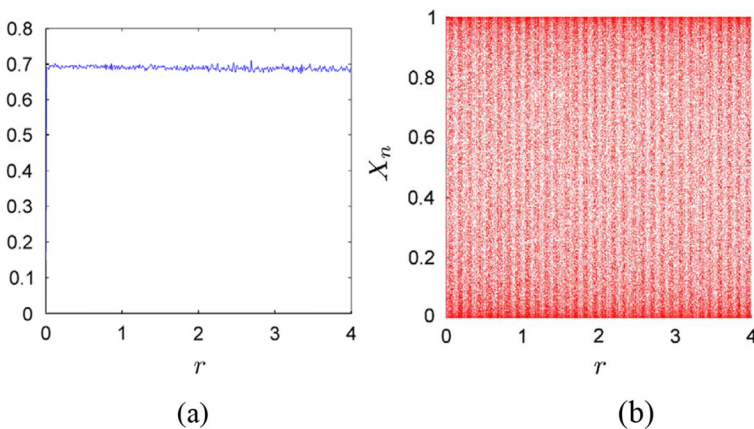


Fig. 2 The LS chaotic system with parameter r



**Fig. 3** (a) The Lyapunov Exponent and (b) the bifurcation diagram of the Logistic system

Besides, every individual has a fitness value. A fitness function used to select the best individuals. The result of the fitness function is the fitness value that represents the solution quality. The higher the amount of fitness, the higher the solution quality. Choosing the best individuals based on their quality is used to produce what is called the mating pool, where the higher quality individual is more likely to be chosen in the mating pool. The individuals in the mating pool are called parents. Every two parents selected from the mating pool will generate two offspring. By just mating high-quality individuals, it was expected to get a better-quality offspring than its parents. So, this will kill the corrupt individuals from making more corrupt individuals. By keeping selecting and mating high-quality individuals, there will be higher chances just to keep good properties of the individuals and leave out bad ones. Finally, this will end up with the desired optimal or acceptable solution. Individuals in the mating pool are called parents. Both parents from the mating pool produce two children. Only by pairing high-quality individuals are they expected to have better quality offspring than their parents. So, this causes corrupt individuals eliminated by producing more bad individuals. By keeping the selection and mating of high-quality individuals, you are more likely just to maintain the good



**Fig. 4** (a) The Lyapunov Exponent and (b) the bifurcation diagram of the LS system

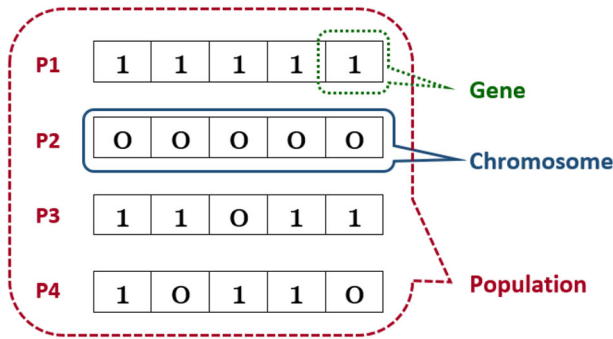


Fig. 5 Population, Chromosome, and gene

qualities of the individuals and avoid the bad ones. Finally, it ends with the desired or desirable solution.

Offspring that were currently produced by using selected parents only have the characteristics of their parents, and there is no change in them anymore. Nothing new added to it, and so there will be the same parenting bugs in the new offspring. Changes will be made for each offspring to create new individuals in order to fix this problem. All the newly created individuals will be a new population, replacing the old population used previously. Any population created is called generation. The process of replacing the old population with the new population is called replacement. Figure 6 summarizes the GA steps.

### 3 Related work

Many data encryption methods use chaotic maps [1–3, 7, 12, 22, 27, 28, 31, 33] because it is widely applicable and easy to understand. Chaotic based image encryption occurs in two steps, confusion, and diffusion. In the confusion step, an image’s pixels scrambled using a secret key

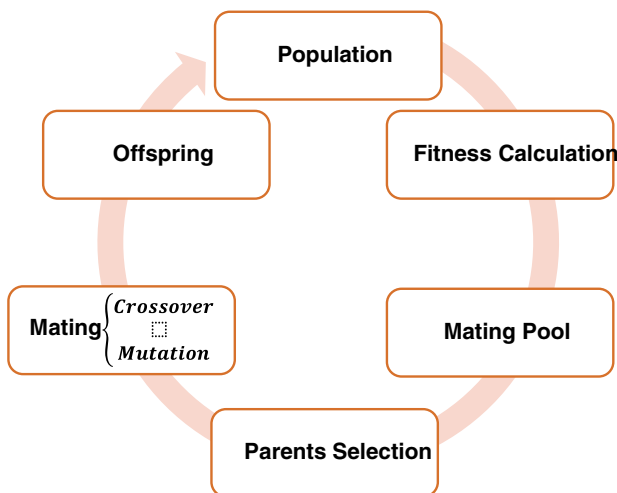


Fig. 6 GA steps. Here we used the Genetic Algorithm to optimization and selecting the best encrypted images that explained in section 4.3

based upon control parameters. While in the diffusion step. While in the diffusion stage, the pixel values changed using the sequence generated by the chaos. Both methods make chaos encryption very secure [9]. The core of a chaos-based cryptosystem is the chaotic map, one-dimensional (1D), and high-dimensional (HD) are two types of chaotic map that 1D chaotic map has fewer parameters and variables, and simpler phase space trajectory rather than HD chaotic map [21].

In 1998, Fredrick proposes the first chaos-based permutation-diffusion algorithm for image encryption [11]. He used a 2D chaotic map to shuffle pixels of plain-image, and next by exploiting a 1D chaotic function, these pixels gray-level are altered sequentially [18]. After that, many chaos-based encryption methods introduced according to his scheme [32]. However, different researchers use different chaos systems. The analysis shows that each algorithm has its own strengths and weaknesses in terms of security and efficiency. The essential problems of these algorithms relate to three aspects: chaos system selection, cryptographic architecture selection, and cryptographic implementation [12].

Chaos systems are used as the backbone of cryptographic systems so that their complex behavior depends on control parameters and initial conditions. The chaotic system selection must be such that it can guarantee chaotic behavior for all values of control parameters. If this is not the case, the codes associated with the dynamic system will no longer be secure. Confusion and diffusion features that reflect security are directly related to the probability distribution function of the underlying chaos system. Hence a chaotic map with a uniform probability distribution function is the best choice to prevent information leakage from an encryption system.

For example, Pareek et al. [28], proposed an image encryption approach based on the logistic map, they employed an 80-bit secret key and two chaotic logistic maps in their method. This external secret key provided the initial situations for logistic maps by applying variant weightage to all bits. Another plaintext-dependent image encryption algorithm introduced in [29]. They employed the logistic map for encryption gray-scale images. The original image was partitioned into blocks and then, by XOR operation and chaotic windows, encrypted. They used 1D logistic chaotic map in their algorithm. However, that is efficient and simple but has a small key space, so they used the chaotic logistic map several times in their algorithm to overcome this drawback. To do parallel processing, they used two  $16 \times 16$  chaotic blocks. Hua and Zhou [14] proposed a 2D Logistic-adjusted-Sine map (2D-LASM), in which the logistic map used to tune the sine map input, and; afterwards its phase plane extended to 2D [14]. Hua et al. [15] proposed a 2D chaotic map called Logistic-modulated-sine-coupling-logistic chaotic map (LSMCL). they exploit logistic map to modulate sine map and couple logistic and sine map together. In addition, an image encryption method based on the chaotic improved logistic system (ILS) proposed in [3]. In [16], a method for data stream encryption presented using a genetic algorithm and pseudo-random sequence. Zhang et al. employ a hyper chaotic Chen system for image encryption [35]. Chai et al. [6] proposed an image encryption technique using a 2D logistic chaotic map and DNA sequence operations. In [36], to improve the security and robustness of existing image encryption algorithms and decrease the security risks of encryption algorithms against most of the attacks, authors presented an image encryption algorithm based on image augmentation, chaos, and DNA coding.

Nepomuceno et al. [25] use natural interval extensions and the lower bound error to generate a random sequence based on the pseudo-orbits of the 1D logistic map. Their proposed method produces a large key space, which can be easily increase. Therefore, it is more important to choose a proper chaos system than to design an encryption algorithm. No matter

how good and how strong the cryptographic system designed, the cipher can easily break if the poor chaos system had selected [12].

In another view, one of the major limitations when running chaotic systems on digital devices is their numerical precision. In fact, after a certain number of iterations, the generator may indicate a certain degree of periodicity or digital chaos destruction. So, these limits the digital chaos encryption effect to large amounts of information, such as high-resolution images. Although most chaos-based encryption algorithms report good statistical properties, they are slow to execute due to their inherent dependence on the data of the proposed programs. Some of these methods designed to use sophisticated chaos systems that require significant computational resources. Technology is constantly evolving with significant advances, including multi-core processors, which is why new threats and vulnerabilities devised that endanger security information in communications systems. Therefore, some researchers used parallel computing to increase security and speedup of the encryption process [5, 10, 30]. For example, Song et al. exploited the multi-threading technique in which every thread competes with others to encrypt images. They used the logistic map and Lorenz system for generating keystream for permutation and diffusion, respectively.

## 4 The proposed encryption scheme

The proposed method includes three main steps: confusion, diffusion, and optimization step by the GA. In the first phase, a scrambled image will be generated by using Chen map function and moving all of the pixels of the original image. In the next phase, intensity values of pixels will be changed by LS map function, and the initial population for GA will be generated. In the last phase, correlation coefficient between pixels will be calculated by using the genetic algorithm, the best cipher will be selected.

### 4.1 Confusion phase

A suitable encryption system has to be susceptible to encryption keys, and good key space is large enough to have the best strength against any kind of attack [37]. In hyper-chaotic functions, space keys are very large, so they have most security against attacks. The proposed algorithm, we used hyper-chaotic Chen function that is one of the hyper-chaotic systems, which is explained by Eq. 2.

We set system parameters as  $a = 35$ ,  $b = 3$ , and  $c = 28$ . The primary values are  $X_0$ ,  $Y_0$  and  $Z_0$  that consider as systems-keys. We set these initial values as  $X_0 = -9$ ,  $Y_0 = -5$ ,  $Z_0 = 14$ .

The confusion phase consists of the following steps:

- Step 1. Take the plain image as input.
- Step 2. Change the 2D original image  $I$  ( $M \times N$ ) to 1D array  $A$  of length  $L = M \times N$ .
- Step 3. Generating three random sequences  $X_n$ ,  $Y_n$ ,  $Z_n$  by using Chen-chaotic function.
- Step 4. One of three above random sequences is selected and arranged low to high as 1D array  $W$ .
- Step 5. Indexes of array  $W$  saved in new array  $D$ .
- Step 6. All Pixels of 1D array  $A$  moved according to elements of  $D$ .
- Step 7. Reshape 2D-image  $I$  of size  $M \times N$  from 1D array  $A$  and generate a scramble image as presented in Fig. 7.



## 4.2 Diffusion phase

The LS chaotic map is a nonlinear mixture of two different Logistic map, and Sine map [37]. The Logistic map has an analogous chaotic behavior with the Sine map. So, both of them have problems that by combining them, a system with more complex chaotic properties generated. One of the Logistic chaotic map problems is its limited chaotic range within  $[3.57, 4]$  which, in LS map solved. The LS map has a wider chaotic range within  $(0,4]$ , as shown in Fig. 4(b). Also, the chaotic sequences range in the Logistic map is non-uniform distribution in  $[0,1]$ , but the LS map sequences have a uniform-distribution within  $[0,1]$ , as shown in Fig. 4(a).

In the proposed method, we used the LS map that explained in Eq. (3) to change intensity values of scramble Image's pixels from confusion phase by following these steps:

Step 1. Split the scrambled 2Dimage with size  $512 \times 512$  to four  $256 \times 256$  blocks, as shown in Fig. 8.

Step 2. Employ the LS function to encrypt all pixels in each block as follow:

- A. In each block, the first five pixels are selected as the encryption key for forming the initial value.
- B. Each of these initial values is an eight-bit block that is converted into decimal numbers and reserved in array B with size 40 by using the following equation:

$$B = [p1, 1, p1, 2, \dots, p2, 1, p2, 2, \dots, p5, 7, p5, 8] \quad (4)$$

- C. apply the *LS* function (Eq.3) on above initial values as follows:

$$\text{Encryption function} = LS(r_1, B) \oplus LS(r_2, B) \quad (5)$$

Where  $r_1$  and  $r_2$  are random numbers in  $(0, 4]$ .

Step 3. Step2 must repeated 256 times, and then the initial population of the genetic algorithm with 256 members ( $512 \times 512$ ) will be obtained.

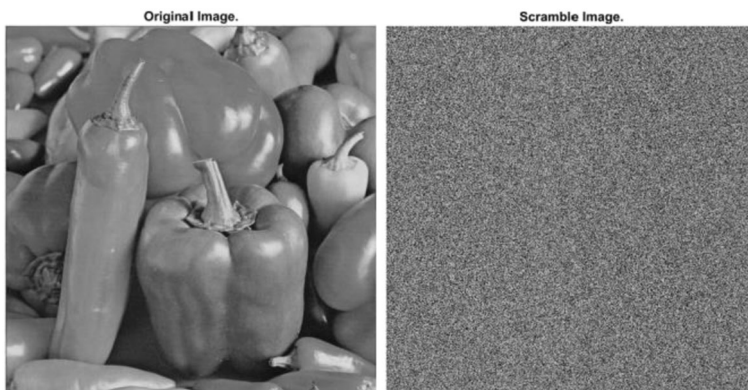
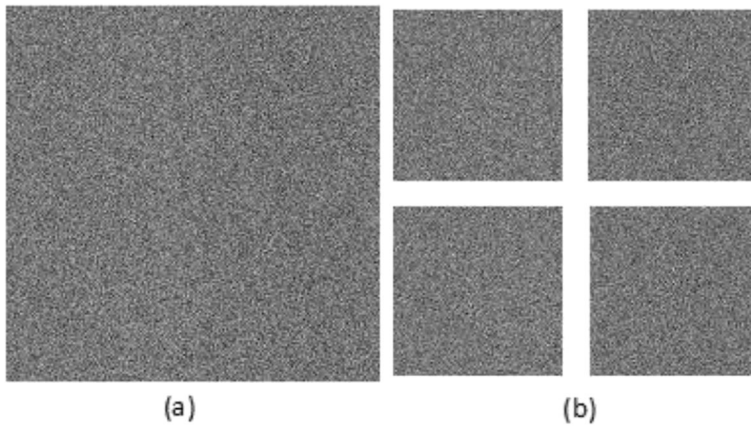


Fig. 7 The plain image and the scrambled image of peppers



**Fig. 8** (a) Scramble image of peppers (b) Scramble image partitioned into four equal blocks

### 4.3 Optimization phase using genetic algorithm

In the proposed method, we used GA to optimize and selecting the best-encrypted images by the below steps. The initial population for GA formed in previous phases after employing *LS* function.

- Step 1. The correlation coefficient between the next pixels of the encrypted image considered as the fitness function that will be calculate in each stage. The horizontal, vertical, and diagonal correlation coefficients between two next pixels can be obtain by Eq.6 [7, 8, 37].
- Step 2. The master population made by choosing 10 % of the population with the minimum correlation.
- Step 3. Use the rest population and the crossover operation for generating new generations and select the best from by evaluated fitness function. The best generated with the minimum correlation coefficient was chosen.
- Step 4. Steps 1 to 3 will be repeated 100 times to generated new generations and select the best-encrypted images. Finally, the best-encrypted image with the minimum correlation coefficient will be select as the cipher image. The process of this approach illustrated in Fig. 9.

## 5 Simulation results and analysis

The results of validating the proposed method's efficiency described in this section. Input images are gray scale with dimension  $512 \times 512$ . Some experiments performed to Image evaluation, and their results shown. Finally, we compared the proposed encryption method with some recent methods [1, 8, 21, 27, 29] in term of some security measures defined in Table 2.

## 5.1 Statistical analysis

### 5.1.1 Histogram analysis

The histogram is one of the most statistical characteristics that shows the distribution of gray pixels of an image [22, 26]. The histogram of the plain image is regular, and the attackers can extract some information. The histogram of the cipher image of a suitable encryption approach should be uniform [6]. In fact, a beneficial image encryption method must generate a uniformly distributed histogram for the cipher images [22]. In Fig. 10, the outputs of our method for some images and their histogram illustrated. As shown in Fig. 8, all encrypted images have a uniform histogram. Thus, the attacker is not able to extract any information about gray values from the encrypted image, so the proposed method is strength against statistical assaults.

### 5.1.2 Correlation analysis

The correlation coefficient is also an important characteristic in the image encryption area, which is defined by the correlation between two adjacent pixels in an image [8]. In an ordinary image, the values of adjacent pixels are close, so its correlation coefficients are always high [4, 7]. A strength image encryption algorithm should generate cipher images with low correlation among adjacent pixels in three directions as far as possible [1, 8, 37].

The correlation coefficient in horizontal, vertical, and diagonal directions can be obtain by the following Eq.6 [7, 8, 37].

$$r_{xy} = \frac{|cov(x,y)|}{\sqrt{D(x)} \times \sqrt{D(y)}} \tag{6}$$

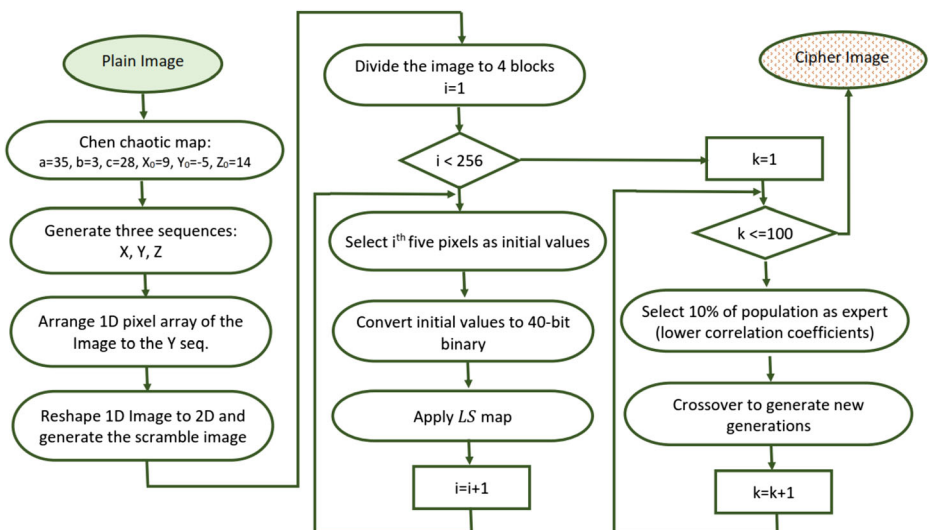
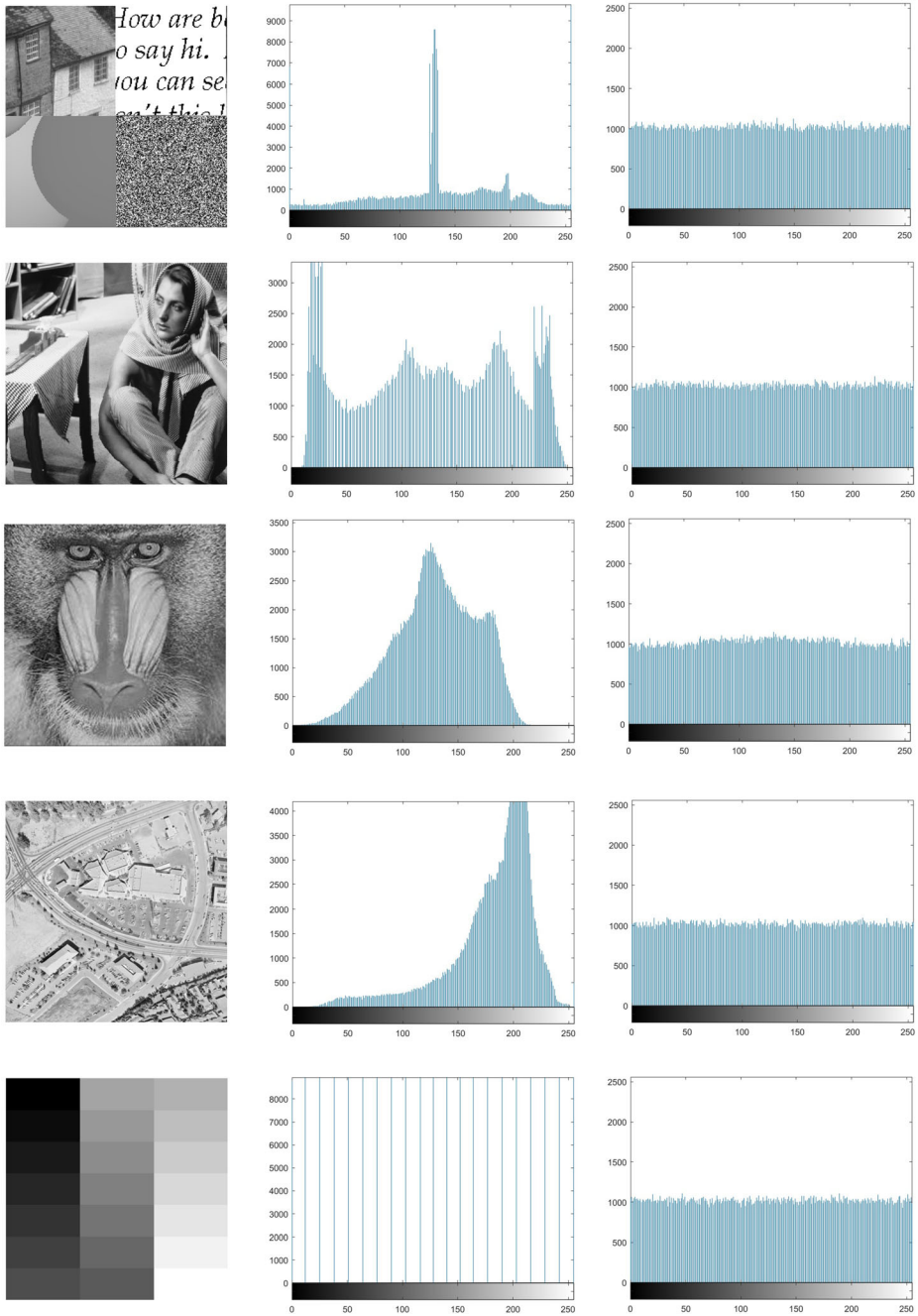


Fig. 9 Flowchart of the proposed method

**Table 2** Security measures

Measure	Calculating Method	Description
Key space and key sensitivity	If a key were eight bits (one byte) long, the key space would consist of 28 or 256 possible keys.	The size of the key space should not be smaller than $2^{100}$ to prevent brute force attacks. In addition, an encryption algorithm must be sensitive to any change of its keys. Higher the key space and key sensitivity, the system is less detectable.
Entropy	$H(s) = \sum_{i=0}^{i=2^M-1} P(s_i) \log_2 \left( \frac{1}{P(s_i)} \right)$	The entropy used for defining the level of randomness in an image.
Histogram	The histogram represents the frequency distribution of each gray level in an image.	The histogram of the plain image is regular, and the attackers can extract some information. The histogram of the cipher image of a suitable encryption approach should be uniform
Correlation coefficient	$E(x) = \frac{1}{N} \sum_{i=1}^N x_i, \quad D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2$ $r_{xy} = \frac{ cov(x,y) }{\sqrt{D(x)} \times \sqrt{D(y)}}$ $cov(x,y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y))$	Zero correlation between the pixels of the ciphered image is an ideal condition in a good cipher
NCPR & UACI	$D(i,j) = \begin{cases} 0 & C_1(i,j) = C_2(i,j) \\ 1 & C_1(i,j) \neq C_2(i,j) \end{cases}$ $NPCR(C_1, C_2) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N D(i,j)$ $UACI(C_1, C_2) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{ C_1(i,j) - C_2(i,j) }{255}$	The sensitivity of cipher image to the slight changes (even one bit), in the clear image, is an important feature in image encryption. In other words, by changing even one bit in plain image, the cipher image must be changed. Ideally higher the values of NPCR and UACI, better is the key sensitivity and plain sensitivity.
PSNR	$PSNR(p_m, p_n) = 20 \log \left( \frac{255}{\sqrt{MSE}} \right)$	The Peak Signal to noise ratio (PSNR) is utilized to measure the discrepancy between plain image P, and decrypted image D. For a good decrypted image, the discrepancy between plain image, and decrypted image must be low, and the value of PSNR must be high.
MSE	$MSE(p_m, p_n) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (p_m(i,j) - p_n(i,j))^2$	MSE is the difference between the plain image and the decrypted image. For good performance, this difference must be meager.



**Fig. 10** The histogram of some sample image and their encrypted output. (1st column: Plain images; 2nd column: Histogram of Plain images; 3rd column: Histogram of encrypted images)

$$cov(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y)) \tag{7}$$

$$E(x) = \frac{1}{N} \sum_{i=1}^N x_i \tag{8}$$

$$D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2 \tag{9}$$

Here,  $x$  and  $y$  are gray values of two next pixels, and  $N$  is the total number of pairs of vertically, horizontally, or diagonally neighbor pixels of the image [4, 7]. The proposed approach and Table 3 have encrypted some sample plain images in  $512 \times 512$  presents the correlation coefficients of these images and their corresponding encrypted images.

As shown in Table 3, two neighbor pixels of plain images are highly correlated (close to 1), where the neighbor pixels in ciphers have low correlation (close to 0). Also, Fig. 11 illustrates the correlation coefficient of neighbor pixels in the Lena image and its cipher, respectively. As a result, the correlations of adjacent pixels of encrypted images generated by the proposed encryption approach in all directions are approximately equal to zero. Thus, this proposed method has good efficiency in image encryption and is secure against statistical attacks.












### 5.1.3 Information entropy analysis

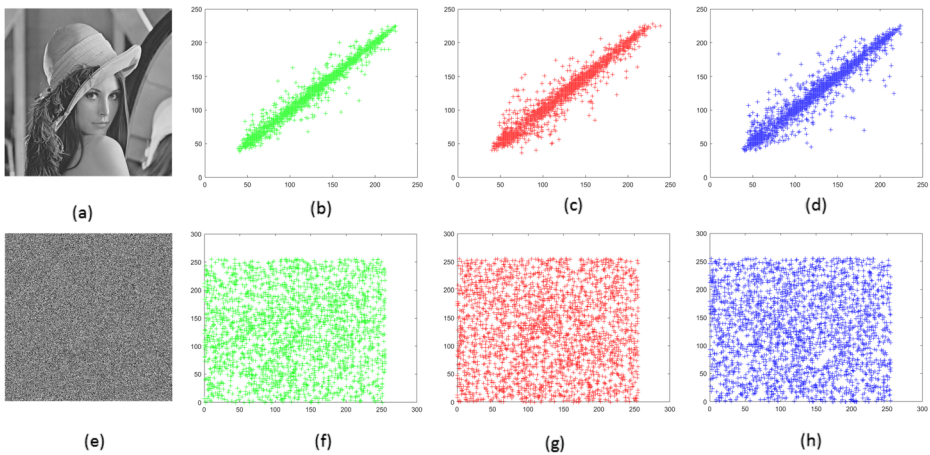
The entropy has been used for defining the level of randomness in an image. [1, 21]. Calculating the entropy of a message with 2  $M$  symbols was introduced by the following equation.

$$H(s) = \sum_{i=0}^{i=2^M-1} P(s_i) \log_2 \left( \frac{1}{P(s_i)} \right) \tag{10}$$

Where  $M$  is the number of bits which is need to indicate the symbol  $S_i$  ( $M$  for gray level images is 8), and  $P(S_i)$  represents the probability of the existence of symbol  $S_i$  in the image [1, 33]. In 8-bit gray-scale images, all of the pixels are distributed randomly and has 256 symbols. The probability of each symbol is  $1/256$ , and the entropy is equal with eight. So in an ideal encryption system with the distributed histogram of output, the entropy of cipher images

**Table 3** The correlation coefficients in horizontal, vertical and diagonal directions for plain image (PI) and cipher image (CI)

												
<b>HORIZONTAL</b>	PI	0.9741	0.9908	0.9647	0.9900	0.7891	0.9008	0.8927	0.4542	0.9965	0.9381	
	CI	0.0033	0.0068	-0.0023	0.0026	0.0006	-0.0030	0.0017	-0.0002	-0.0017	0.0019	
<b>VERTICAL</b>	PI	0.9862	0.9927	0.9544	0.9829	0.8079	0.8602	0.9542	0.4648	0.9998	0.9713	
	CI	-0.0040	-0.0022	-0.0002	0.0002	-0.0012	0.0006	0.0022	0.0013	0.0032	0.0003	
<b>DIAGONAL</b>	PI	0.9619	0.9828	0.9275	0.9732	0.6784	0.8031	0.8839	-0.0290	0.9964	0.9222	
	CI	-0.0002	0.0005	-0.0008	-0.0015	-0.0002	-0.0004	0.0011	-0.0022	0.0007	0.0006	



**Fig. 11** (a) The plain Lena image- (b) distributions of two horizontally neighboring pixels- (c) distributions of two diagonally neighbor pixels- (d) distributions of two diagonally neighboring pixels- (e) the cipher image of Lena- (f) distributions of two horizontally neighboring pixels- (g) distributions of two vertically neighbor pixels- (h) distributions of two diagonally neighboring pixels

should be equal to eight [37]. Table 4 shows the entropy of specific gray encrypted pictures generated by our proposed scheme.

### 5.2 Differential attacks

The sensitivity of cipher image to the slight changes (even one bit), in the plain image, is an important feature in image encryption. In other words, by changing even one bit in plain image, the cipher image must be changed. By comparing the encrypted images of the original image and the modified original image, the sensation of changing a single pixel of the original image can be measured. NPCR and UACI are two criteria for measuring this property. The Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) calculated by Eqs. 11–13.

$$D(i, j) = \begin{cases} 0 & C_1(i, j) = C_2(i, j) \\ 1 & C_1(i, j) \neq C_2(i, j) \end{cases} \tag{11}$$

$$NPCR(C_1, C_2) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N D(i, j) \tag{12}$$

$$UACI(C_1, C_2) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{|C_1(i, j) - C_2(i, j)|}{255} \tag{13}$$

**Table 4** The entropy of original some image before and after encryption

Plain image	7.3479	7.6040	7.2230	7.1030	6.4391	6.9940	7.3438	0.5000	4.3923	7.1914
Cipher image	7.9990	7.9992	7.9986	7.9991	7.9993	7.9992	7.9993	7.9979	7.9993	7.9993

Here,  $M$  and  $N$  are sizes of the image,  $C_1$ , and  $C_2$  are corresponding cipher images of the original images that have one different pixel. First, the clear image is encrypted. Afterward, one pixel of the original image is modified, and then it is encrypted. The calculated NPCR and UACI between two obtained cipher images were shown in Table 5. Figure 12. shows the cipher images of the original image Ruler.512 and its modified image that have only one different pixel with it. The optimal values of these parameters are NPCR and UACI are 0.9961, and 0.3346, respectively.

### 5.3 Key analysis

Key space and key sensitivity analysis, play main roles to strength against brute-force attack.











#### 5.3.1 Key space analysis

The encryption algorithm has to have a large key space in order to resist brute-force attacks. In fact, a brute-force attack has made by seeking all possible keys to obtain the real key. We used three initial values  $X_0, Y_0, Z_0$ , and two parameters  $r_1$  and  $r_2$  in our algorithm. The precision of the initial values and parameters was  $10^{-14}$  in our analysis. Therefore, the key space of the proposed algorithm could be calculated as:  $(10^{14})^3 \times (10^{14})^2 = (10)^{70} \cong 2^{224}$ , so the key space is large enough.

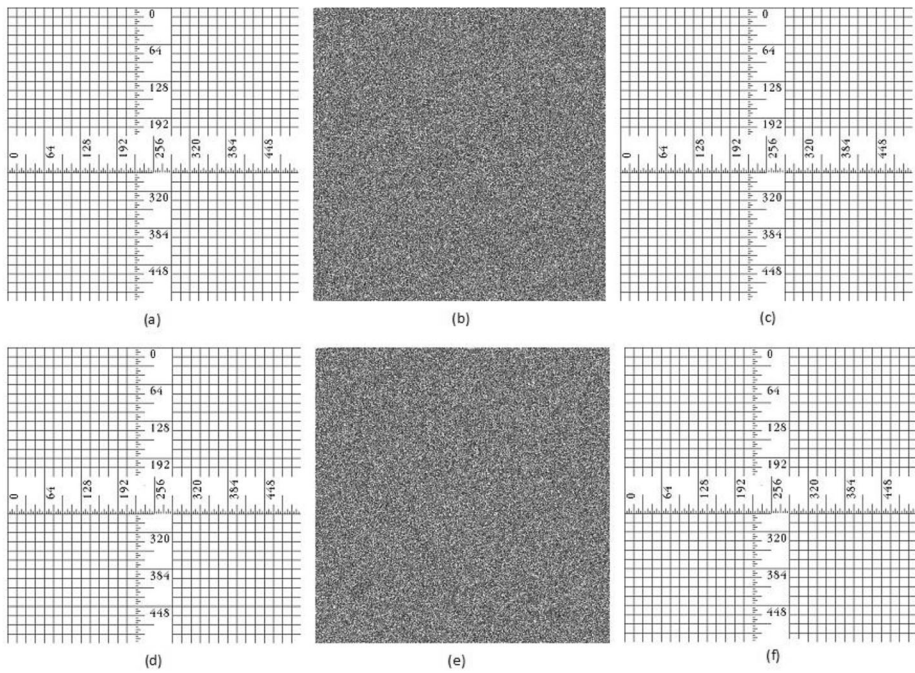
#### 5.3.2 Key sensitivity analysis

Another important feature in the encryption algorithm is the sensitivity to small changes in keys. In other word, the encryption algorithm must have a large key space to resist brute-force attacks and must be sensitive to any change of its keys. As it was mentioned in section 4, keys of our algorithm consist of two parameters ( $r_1$  and  $r_2$ ) where  $r_1$  and  $r_2$  are in the range of  $(0, 4]$ . For key sensitivity, a primary key set ( $k_1$ ) is  $r_1 = 3.99$  and  $r_2 = 3.79$ . We use  $k_1$  to encrypt plain image of Barbara to obtain the encrypted image defined as E1. Then we generated another key set ( $K_2$ ) by changed  $r_1$  to 3.990000000000001 while keeping others unchanged. Using  $k_2$  to encrypt the Barbara image again and generated another encrypted image (E2). Table 6 shows the result of evaluating the difference between these two encrypted images (E1, E2). The proposed algorithm is highly sensitive to changing key value, even a small change ( $10^{-14}$ ) in the key value. In the decryption process, we used  $k_1$  to encrypt the Barbara image and then used  $k_2$  to decrypt the encrypted image. Figure 13 shows the

**Table 5** Evaluating the difference between two cipher images of the original images, which have one-pixel difference in terms of NPCR and UACI

										
NPCR	0.9957	0.9955	0.9957	0.9956	0.9956	0.9955	0.9958	0.9954	0.9959	0.9958
UACI	0.3335	0.3335	0.3317	0.3340	0.3356	0.3359	0.3364	0.3428	0.3362	0.3331















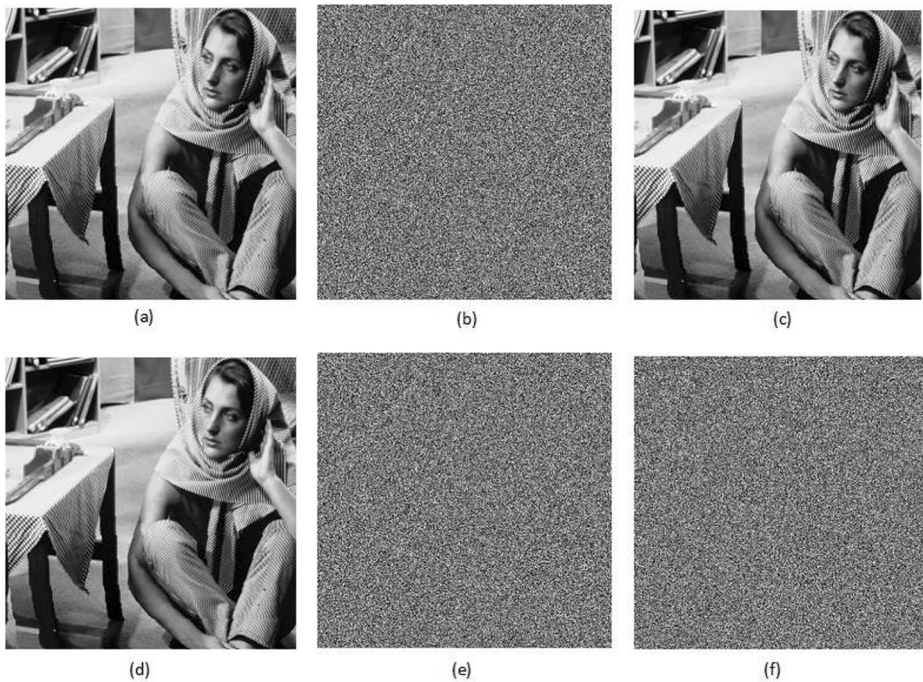
**Fig. 12** (a) The original image – (b) The cipher image – (c) The decrypted image – (d) The modified original image in one pixel – (e) The cipher of modified image – (f) The encrypted the modified cipher image

decrypted results. As we can see, by using the correct key ( $k_1$ ), the decryption process works correctly and reconstruct the original image completely. However, the decryption process will fail by a small change in key value. Table 7 shows the NPCR and UACI results by changing keys. Hence, the proposed algorithm is very sensitive to the key value in encryption and decryption processes.

The Peak Signal to Noise Ratio (PSNR) is utilized to measure the discrepancy between plain image P and decrypted image D [15]. For good decrypted image, the discrepancy between the plain image and the decrypted image must be low and the value of PSNR must be high. PSNR is defined by Eq.14:

**Table 6** Evaluating the difference between original and cipher images in terms of PSNR, MSE

										
PSNR	9.4369	8.9758	9.8383	8.4414	7.4554	8.2149	7.8660	4.7211	7.5747	9.3419
MSE	4.6527	4.1948	5.0364	3.6782	2.7094	3.4523	3.0965	0.0074	2.8125	4.5673



**Fig. 13** results of key sensitivity: (a), (d) The original image – (b), (e) encrypted image by  $k_1$  – (c) decrypted image by correct key ( $k_1$ ) - (f) decrypted image with a small change in key ( $k_2$ ).

$$PSNR(p_m, p_n) = 20 \log \left( \frac{255}{MSE} \right) \tag{14}$$

The Mean Square Error (MSE) is the difference between the plain image and the decrypted image. For a good performance, this difference must be very low. MSE is evaluated by Eq.15:

$$MSE(p_m, p_n) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (p_m(i, j) - p_d(i, j))^2 \tag{15}$$

Where  $p_m(i, j)$  and  $p_d(i, j)$  is the grey-scale values of the pixels of plain and decrypted images with  $M \times N$  size, respectively. For a good encrypting method, the discrepancy between the plain image and the decrypted image is very low, so MSE is low, and PSNR will be high.

Table 8 shows the comparison of the proposed approach with several recently related approaches, which states our algorithm efficiency in terms of statistical analysis. According to this table, all methods are acceptable, and among these comparable methods, there is no one

**Table 7** Evaluating the difference between these two encrypted images by  $k_1$  and  $k_2$

	Key value $k_1$ $r_1 = 3.99, r_2 = 3.75$	Key value $k_2$ $r_1 = 3.9900000000000001, r_2 = 3.75$
NPCR	0.995956420898438	
UACI	0.331040565640319	

**Table 8** Comparison of the proposed method with other methods

Measure	Img. Name	Methods											Proposed		
		[3]	[27]	[1]	[8]	[15]	[30]	[5]	[25]	[24]	[24]				
Entropy	Lena	7.9980	7.9965	7.9923	<b>7.9994</b>	7.9572	7.9993	7.9968	7.9991						
	Pep.		7.9950	7.9929	7.9983			7.9961	7.9992	7.9961	7.9968				
	Bab.		7.9962	7.9926	7.9981			0.9958	7.9971	7.9971	7.9971				
	C.Man				7.9972			0.9959	7.9971	7.9971	7.9904				
	Aerial		7.9935		7.9988	7.9018									
	Boat				<b>0.9975</b>										
	Lena		0.9964		0.9950										
	Pep.		0.9961		0.9789										
	Bab.		0.9962												
	C.Man		<b>0.9960</b>												
Aerial															
Boat					<b>0.9959</b>										
NPCR	Lena		<b>0.3414</b>		0.3261					0.3161	0.3346	0.3308	0.3335	0.3335	
	Pep.		0.3317		0.3339				<b>0.3374</b>			0.3219	0.3335	0.3317	
	Bab.		<b>0.3356</b>		0.3261							0.3156	0.3317	0.3340	
	C.Man								<b>0.3370</b>			0.3315	0.3340	<b>0.3359</b>	
Aerial		0.3303										0.3331	0.3331		
Boat													0.0033		
Correlation Coefficients	Lena		0.0242		0.0093					0.0059	-0.0021	<b>-0.0006</b>	0.0033	0.0033	
	Pep.		0.0261	0.0033	-0.0054					0.0059	<b>-0.0029</b>	0.0048	-0.0040	-0.0040	
	Bab.		0.0245	-0.0067	-0.0009					0.0059	-0.0016	-0.0243	<b>-0.0002</b>	<b>-0.0002</b>	
	C.Man												<b>-0.0008</b>	<b>-0.0008</b>	
Aerial													<b>0.0019</b>		
Boat													<b>0.0003</b>		
Key space	Lena													<b>0.0006</b>	
	Pep.													224	
	Bab.													224	
	C.Man													224	
Key space	Lena		2.189	2.40	2.240	2.256	2.212	2.179	2.283	2.179	2.179	2.179	2.179	2.224	
	Pep.														
	Bab.														
	C.Man														

**Table 9** PSNR results after salt-peppers noise

	d = 0.001	d = 0.01	d = 0.05	d = 0.1
PSNR	38.6066	15.3057	14.7927	11.9384
MSE	32.9589	9.9062	9.4172	6.8015

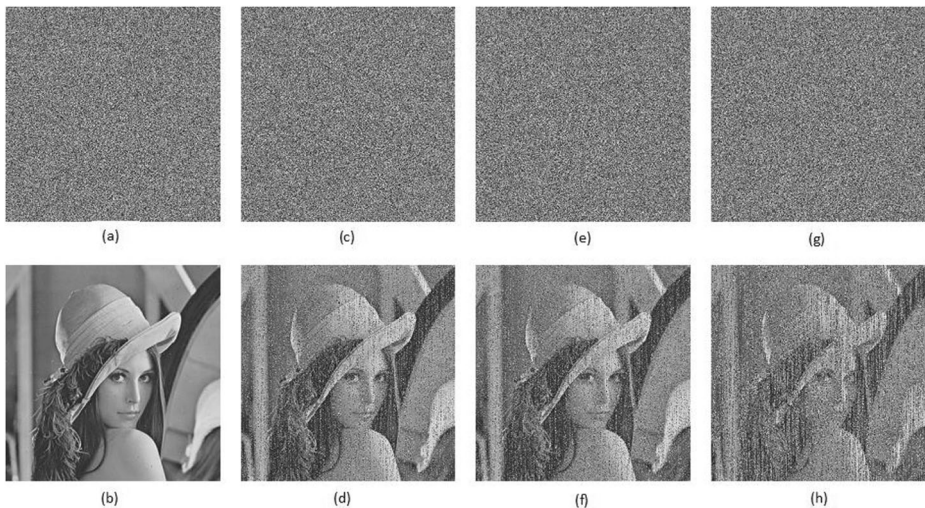
works best on all criteria for all the images tested. The proposed method performs better than others in terms of entropy and correlation in most cases. However, the fitness function definition is an essential step in the proposed method, and the other parameters can also be improved by more precisely defining a combined fitness function for the optimization algorithm.

#### 5.4 Noise attack and data loss

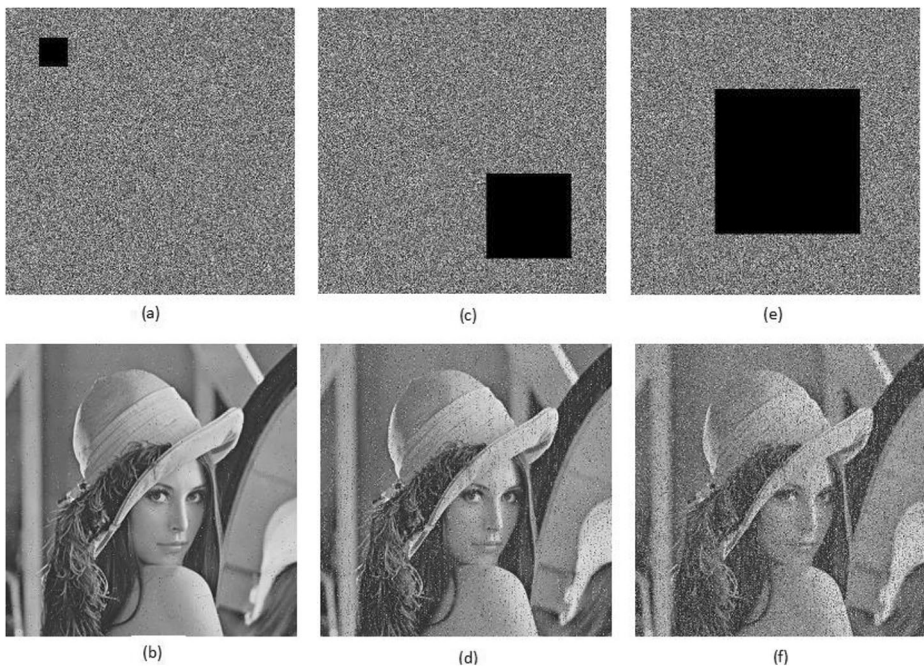
When an image transmitted in the real physical world, the data might have noised or lost. Therefore, a great design encrypted image algorithm should be robust against noise and loss data and recover the original image without missing too much critical information [15, 22, 27].

We tested our algorithm against salt-pepper noise by appending this noise to the encrypted image and then decrypting it again. PSNR results for Lena plain image with salt-pepper noise illustrated in Table 9. Figure 14 shows our algorithm has good robustness against noise as well as other methods.

To test the effect of data loss in the proposed algorithm, first, the plain image was encrypted, then we replaced a  $50 \times 50$ -pixel partition of the encrypted image with zero value. So, the modified encrypted image was decrypted by our algorithm. The results of the data loss attack illustrated in Fig. 15. As can be seen, even by losing 50% of the pixels, we are still able to get a sharp decrypted differentiable image. Therefore, our algorithm is tight against the data loss attack.



**Fig. 14** The results of the salt-peppers attack. (a), (b) adding noise with  $d=0.001$  and the resulting decrypted image, (c), (d) adding noise with  $d=0.01$  and the resulting decrypted image, (e), (f) adding noise with  $d=0.05$  and the resulting decrypted image, (g), (h) adding noise with  $d=0.1$  and the resulting decrypted image



**Fig. 15** The results of data loss attack. The (a), (b) Data loss of a  $50 \times 50$  pixels area and the resulting decrypted image. (c), (d) Data loss of a  $150 \times 150$  pixels area and the resulting decrypted image. (e), (f) Data loss of a  $256 \times 256$  pixels area and the resulting decrypted image

## 6 Conclusions

In this paper, an improved gray scale image encryption algorithm based on chaotic systems presented.

The chaotic systems like Logistic, Sine, and Tent map have a lot of applications due to their natural structures, but they have some difficulties. The combined LS system solves above problems and has good chaotic properties like a wide range of parameters and uniform distribution of data. By using this combination, different chaotic systems can be found.

In this schema, three disordered functions in confusion, and diffusion phases were employed for encryption and reducing the coefficient of correlation between pixels, and also to increase the security level of encryption system as well as possible. Then by using genetic algorithm, the encryption process is improved, and the best cipher images selected while the correlation coefficient among pixels in three vertical, horizontal and diagonal directions was closed to zero. As we are using two chaotic systems, the proposed algorithm has a large and secure key space. Also, this algorithm makes uniformly distributed histograms for cipher images. Thus, the attacker is not able to get any information about gray level values distribution from the encrypted image, and hence the proposed method is resistant against statistical attacks. The decryption system is susceptible to key, and if there is a small change in key, the output of the decryption system has a distributed histogram and will be like a noise.

In the future, we can exploit chaotic quantum systems and hyper-chaos as well as the significant effect of coupling and modulation to design more complex encryption algorithms with different simple quantum chaotic systems to enhance the encryption process. Also, other heuristic optimization algorithms such as particle swarm optimization (PSO), Black Widow

Optimization (BWO), Barnacles Mating Optimizer (BMO) algorithms instead of GA. It is obvious that a particular chaos function is not suitable for all types of images. So, one can use some machine learning techniques to choose a proper chaotic function corresponding to the input image to improve the encryption process.

## Compliance with ethical standards

**Conflict of interest** The authors declare no conflict of interest.

## References

1. Abdullah AH, Enayatifar R, Lee M (2012) A hybrid genetic algorithm and chaotic function model for image encryption. *AEU-Int J Electron Commun* 66(10):806–816
2. Arroyo D, Díaz J, Rodríguez FB (2013) Cryptanalysis of a one round chaos-based substitution permutation network. *Signal Process* 93(5):1358–1364
3. Attaullah ST, Jamal SS (2020) An improved chaotic cryptosystem for image encryption and digital watermarking. *Wirel Pers Commun* 110(3):1429–1442
4. Belazi A, El-Latif AAA, Diaconu A-V, Rhouma R, Belghith S (2017) Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. *Opt Lasers Eng* 88:37–50
5. Çavuşoğlu Ü, Kaçar S (2019) A novel parallel image encryption algorithm based on chaos. *Clust Comput* 22(4):1211–1223
6. Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. *Opt Lasers Eng* 88:197–213
7. Enayatifar R, Abdullah AH, Isnin IF (2014) Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. *Opt Lasers Eng* 56:83–93
8. Enayatifar R, Abdullah AH, Isnin IF, Altameem A, Lee M (2017) Image encryption using a synchronous permutation-diffusion technique. *Opt Lasers Eng* 90:146–154
9. Fadhel S, Shafry M, Farook O (2017) Chaos image encryption methods: a survey study. *Bullet Electric Eng Inform* 6(1):99–104
10. Flores-Vergara A, Inzunza-González E, García-Guerrero EE, López-Bonilla OR, Rodríguez-Orozco E, Hernández-Ontiveros JM, Cárdenas-Valdez J, Tlelo-Cuautle E (2019) Implementing a chaotic cryptosystem by performing parallel computing on embedded systems with multiprocessors. *Entropy* 21(3):268
11. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. *Int J Bifurcat Chaos* 8(06):1259–1284
12. Gayathri J, Subashini S (2016) A survey on security and efficiency issues in chaotic image encryption. *Int J Inf Comput Secur* 8(4):347–381
13. Hu H, Liu L, Ding N (2013) Pseudorandom sequence generator based on the Chen chaotic system. *Comput Phys Commun* 184(3):765–768
14. Hua Z, Zhou Y (2016) Image encryption using 2D logistic-adjusted-sine map. *Inf Sci* 339:237–253
15. Hua Z, Jin F, Xu B, Huang H (2018) 2D logistic-sine-coupling map for image encryption. *Signal Process* 149:148–161
16. Kumari A, Goyal S (2016) Encryption and Code Breaking of Image Using Genetic Algorithm in MATLAB. *Int J* 4(7)
17. Li J, Liu H (2013) Colour image encryption based on advanced encryption standard algorithm with two-dimensional chaotic map. *IET Inf Secur* 7(4):265–270
18. Li C, Luo G, Qin K, Li C (2017) An image encryption scheme based on chaotic tent map. *Nonlinear Dynamics* 87(1):127–133
19. Liao X, Qin Z, Ding L (2017) Data embedding in digital images using critical functions. *Signal Process Image Commun* 58:146–156
20. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. *IEEE Trans Circ Syst Vid Technol*
21. Liu W, Sun K, Zhu C (2016) A fast image encryption algorithm based on chaotic map. *Opt Lasers Eng* 84:26–36
22. Liu Y, Wang J, Fan J, Gong L (2016) Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences. *Multimed Tools Appl* 75(8):4363–4382
23. Lü J, Chen G (2002) A new chaotic attractor coined. *Int J Bifurc Chaos* 12(03):659–661

24. Mozaffari S (2018) Parallel image encryption with bitplane decomposition and genetic algorithm. *Multimed Tools Appl* 77(19):25799–25819
25. Nepomuceno EG, Nardo LG, Arias-Garcia J, Butusov DN, Tutueva A (2019) Image encryption based on the pseudo-orbits from 1D chaotic map. *Chaos: An Interdiscip J Nonlin Sci* 29(6):061101
26. Niyat AY, Moattar MH, Torshiz MN (2017) Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Opt Lasers Eng* 90:225–237
27. Noshadian S, Ebrahimzade A, Kazemitabar SJ (2018) Optimizing chaos based image encryption. *Multimed Tools Appl*:1–22
28. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. *Image Vis Comput* 24(9):926–934
29. Rostami MJ, Shahba A, Saryazdi S, Nezamabadi-Pour H (2017) A novel parallel image encryption with chaotic windows based on logistic map. *Comput Elect Eng*
30. Song W, Zheng Y, Fu C, Shan P (2020) A Novel Batch Image Encryption Algorithm Using Parallel Computing. *Inform Sci*
31. Takkar P, Singh V (2017) Image encryption approach using chaotic map for gray scale images
32. Tewani R, Garg Y, Bagga JS, Singh A, Bhalsodia R (2020) Image Encryption Using Permutation–Diffusion Approach. *Advances in Data Sciences, Security and Applications: Springer* 363–73
33. Xu L, Gou X, Li Z, Li J (2017) A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. *Opt Lasers Eng* 91:41–52
34. Ye G (2010) Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recogn Lett* 31(5):347–354
35. Zhang M, Peng B, Chen Y editors (2019) An Efficient Image Encryption Scheme for Industrial Internet-of-Things Devices. *Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things*
36. Zhang Q, Han J, Ye Y (2019) Image encryption algorithm based on image hashing, improved chaotic mapping and DNA coding. *IET Image Process* 13(14):2905–2915
37. Zhou Y, Bao L, Chen CP (2014) A new 1D chaotic system for image encryption. *Signal Process* 97:172–182

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Mahdieh Ghazvini** received her B.Sc. from Shahid Bahonar University of Kerman, Iran in 2000, and her M.Sc. and Ph.D. from the University of Isfahan, Isfahan, Iran in 2004 and 2013, in Computer Architecture Engineering, respectively. Currently she is assistant professor of Computer Engineering Department at Shahid Bahonar University of Kerman. She is the author of several technical papers in signal processing and telecommunications journals and conferences. Her research interests are wireless networks, game theory, signal processing and neural networks



**Mojdeh Mirzadi** received the B.S. from Shahid Bahonar University of Kerman, Kerman, Iran in 2010 in computer engineering and the M.Sc. in computer engineering from University of Isfahan, Isfahan, 2012. Her research interests are in the area of image processing and security.



**Negin Parvar** received the B.S. from Azad University of Kerman, Kerman, Iran in 2012 in computer engineering and the M.Sc. in artificial intelligence from Azad University of Kerman, Kerman, 2017. Her research interests are in the area of image processing and digital systems.