Journal of Information Security and Applications 52 (2020) 102472

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Deterministic chaos game: A new fractal based pseudo-random number generator and its cryptographic application

Peyman Ayubi[a,*], Saeed Setayeshi[b], Amir Masoud Rahmani[c]

[a] Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
[b] Department of Medical Radiation Engineering, Amirkabir University of Technology, Tehran, Iran
[c] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

ARTICLE INFO

Keywords:
Digital image encryption
Region of interest
Deterministic chaos game
security

ABSTRACT

In this paper, a digital image encryption algorithm is proposed based on the generalized model of the chaos game. The chaos game is a well-known fractal, which acts as a pseudo-random number generator (PRNG) in the proposed encryption algorithm. The foundation of the chaos game is based on basic points and its distance ratio that determine the basis of how they distribute random values in 2D or 3D space. These basic points are entered by the user interface and are the result of an encrypted image with a fractal structure. The use of the bifurcation diagram and Lyapunov exponent analysis showed that the proposed chaos game has the dynamical behavior, and fully chaotic characteristic, and can be used as a secure PRNG in cryptography systems. In the proposed method, the region of interest is determined by a number of Bases, and the fractal mechanism of chaos game for the encryption process is performed on the image. This process is very sensitive to any changes in keys and refers to confusion. The evaluation results of security and performance analysis on standard images confirm the efficiency of the proposed method and demonstrate that the proposed method is robust against attacks.

1. Introduction

With the development of the Internet networks and free access to private information by unauthorized users, the use of an encryption system to protect multimedia systems such as image, video, and audio seems to be necessary. The image is widely used as one of the most important multimedia systems on personal computers and mobile phones. Several algorithms and methods have been used to encrypt the digital image in the last three decades, and the challenges are still ongoing [1].

Image encryption algorithms are divided into two categories: the spatial domain and the transform domain [2]. Spatial domain techniques directly encrypt the original image pixels, that have high speed and more security and easy to use. But the main problem in pixel-based methods is the high capacity of the encrypted image. Transform-based methods are used to encrypt compressed images, such as JPEG and JPEG2000, which has many uses and benefits in saving memory consumption. These algorithms are widely used in DCT and DWT transform domains [3,4].

On the other hand, cryptosystem algorithms are divided into two class of block and stream cipher [5]. The block cipher has high security and less speed than the stream cipher. Particular examples of block cipher techniques are algorithms such as DES [6] and AES [7], which are also used in image encryption [8]. In most block and cipher stream algorithms, the pseudo-random number generator is used to ensure data security. The PRNGs are deterministic sequences that are generated with an initial condition that these initial conditions play as the secret key in the cryptosystem [9]. Providing a secure PRNG requires the passing of standard statistical tests such as NIST [10], Diehard [11,12], ENT [13], TESTU01 [14] and etc.

The use of chaos theory as a secure cryptosystem in the last two decades has been at the forefront of dynamical systems. The sensitivity to the initial condition and the deterministic chaotic sequences in discrete chaotic maps are very important properties in cryptographic systems [15]. For this reason, chaotic iterated maps have been used as pseudo-random number generators along with encryption operators for image and video encryption [16–18]. A variety of applications such as image and video encryption [19–23], digital stenography [24–26], image and video

* Corresponding author.
E-mail addresses: p.ayubi@iaurmia.ac.ir (P. Ayubi), s.setayeh@iut.ac.ir (S. Setayeshi).

https://doi.org/10.1016/j.jisa.2020.102472
2214-2126/

watermarking [27–32] and information hiding [33–35] are applications that use chaotic discrete maps in the security element of applied methods.

Classic maps such as logistic map [36–39], Tent map [40], Arnold cat map [41,42], Henon Map [43,44], and Sine map [45] are used in many image encryption algorithms. Classical logistics or tent maps, due to their smaller keyspace, have low security which has been investigated in cryptanalysis of chaotic maps [46,47].

In the research literature, in order to increase the key space, there are various chaotic maps such as DNA sequences[48,49], hyper chaos [50], mixed map [51–55], chaotic neural network [56], Josephus Scrambling[57], and Quantum map [58]. Also, one of the most important uses of chaotic mapping in the last two decades has been the substitution boxes in block cipher methods [59,60]. Using the science of cryptanalysis, a great deal of study has been done on the security of chaotic maps and the efficiency of these PRNGs [61–65].

A fractal is a geometric structure consisting of components that are obtained by increasing each component to a certain scale, the same basic structure. In other words, the fractal is a structure that each component of a structure is similar to the whole structure [66]. Fractals are found in many natural structures such as snow, mountains, clouds, roots, trunks and leaves of trees, the growth of crystals in igneous rocks, the network of waterways and rivers, electrochemical deposition, bacterial mass growth and blood vessel systems, DNA and..., which can be described, interpreted and predicted by many natural phenomena [67,68]. Many classic fractals have been raised in real and complex domains, such as the Cantor set, the Koch curve, the Sierpinski gasket, the Julia set and the Mandelbrot set [69]. Fractal networks can be used to produce chaotic attractors that have sensitivity to the initial conditions and control parameters [70].

The chaos game as a self-similar fractal was presented in 1988 by Barnsley [71]. The Barnsley's chaos game is a deterministic fractal shape which uses random numbers to produce a geometric structure. In this paper, a modified model of the chaos game is presented which is sensitive to initial conditions and has a fully chaotic behavior. To illustrate the chaotic behaviors of the proposed dynamic system, the Lyapunov exponent and bifurcation diagram are used. The standard randomness tests are used to display the randomness of the chaos game as a safe PRNG. Finally, a database of standard images is used to evaluate the performance of the proposed encryption algorithm and the obtained results of the proposed algorithm are compared with other algorithms.

The rest of the paper is organized as follows. In Section 2, the proposed chaos game will be introduced as a new dynamical system. The encryption and decryption process will be described in Section 3. The experimental results of the proposed method and the evaluation of efficiency parameters in the proposed algorithm are discussed in Section 4. Section 5 will be the conclusion section of this paper.

2. Chaos game

2.1. Barensley's chaos game

Chaos game (CG) was first introduced by Michael F. Barnsley [71]. At first glance, there seems to be no connection between chaos and fractal. In fact, this is not just a game. In this game, there are some simple rules to select and all of them uses the same procedures. These rules are

1. *Set three points on the screen as bases (points are vertices of a triangle).*
2. *The bases (vertices) are tagged with numbers 1, 2 and 3.*
3. *Select a starting point on the sheet.*
4. *Select a base randomly.*
5. *Mark the midpoint between the current point and the selected base as the new game point.*
6. *After obtaining a new point, go to step 4.*

The 2D mathematical model of Barnsley's chaos game is

$$\begin{cases} x_{n+1} = x_n + (B[r][1] - x_n) \times \mu \\ y_{n+1} = y_n + (B[r][2] - y_n) \times \mu \end{cases} \tag{1}$$

Where $x_0, y_0 \in [0, 1]$, $r \in rand\ \{1, 2, 3, .., BaseNo\}$ and B is a two-dimensional array that contains the coordinates of the chaos game bases. The distance ratio in the chaos game is one of the important parameters that we show with μ. In Fig. 1, the steps and outcomes of the chaos game are shown for 3-Bases(BaseNo=3) with a ratio of $\mu = 0.5$ for different points (iterations). In Fig. 1(a), the bases are marked with red points (points represent the number of bases) and the generated sequence of the game points have been labeled by $p_1, p_2, p_3, ...$

Chaos game is a simple procedure for randomly generating a sequence of points. The random attribute in the chaos game is not efficient for encryption, and this property must be converted to a deterministic property to become a pseudo-random number generator. In this paper, a deterministic model of chaos game will be proposed.

Note that when game points are inside the triangle, this process will always remain within the triangle. It is obvious that if the game starts out of the triangle, sooner or later, the game points will be placed inside the triangle. Hence, due to random production, we should expect a random distribution of points that are somewhere between 1, 2 and 3 bases.

Fig. 2 show the results of different *BaseNo* for various μ values. The B array in this figure is

$$B_{(BaseNo=3)} = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, B_{(BaseNo=4)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$
$$B_{(BaseNo=5)} = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & \frac{2}{3} \\ 1 & \frac{2}{3} \\ \frac{1}{5} & 0 \\ \frac{4}{5} & 0 \end{bmatrix}, B_{(BaseNo=6)} = \begin{bmatrix} \frac{1}{5} & 1 \\ \frac{4}{5} & 1 \\ 0 & \frac{1}{2} \\ 1 & \frac{1}{2} \\ \frac{1}{5} & 0 \\ \frac{4}{5} & 0 \end{bmatrix} \tag{2}$$

In fact, with random sequences, we obtain a definite form of geometry, which is called fractal that has a regular and self-similar structure. The self-similarity feature is described in Section 2.2.

The variation in the number of bases and location of bases create beautiful fractals. Some of these generated fractals are shown in Fig. 3.

2.2. Self-similarity

Self-similarity is an essential property in all fractals. In mathematical science, the self-similarity is a complex geometric structure that has the same details on any scale of its structure. In other words, it is said to be an object that exactly or approximately similar to a part of itself. The Sierpinski triangle is one of the most famous fractal objects that is obtained from basic chaos game (See Fig. 1). This shape consists of a large triangle with an infinity of small triangles inside it, and at each stage, there is a form that is a part of the shape of the next step. In other words, this structural triangle contains an exact copy of itself in all magnifications, so the shape has self-similarity properties.

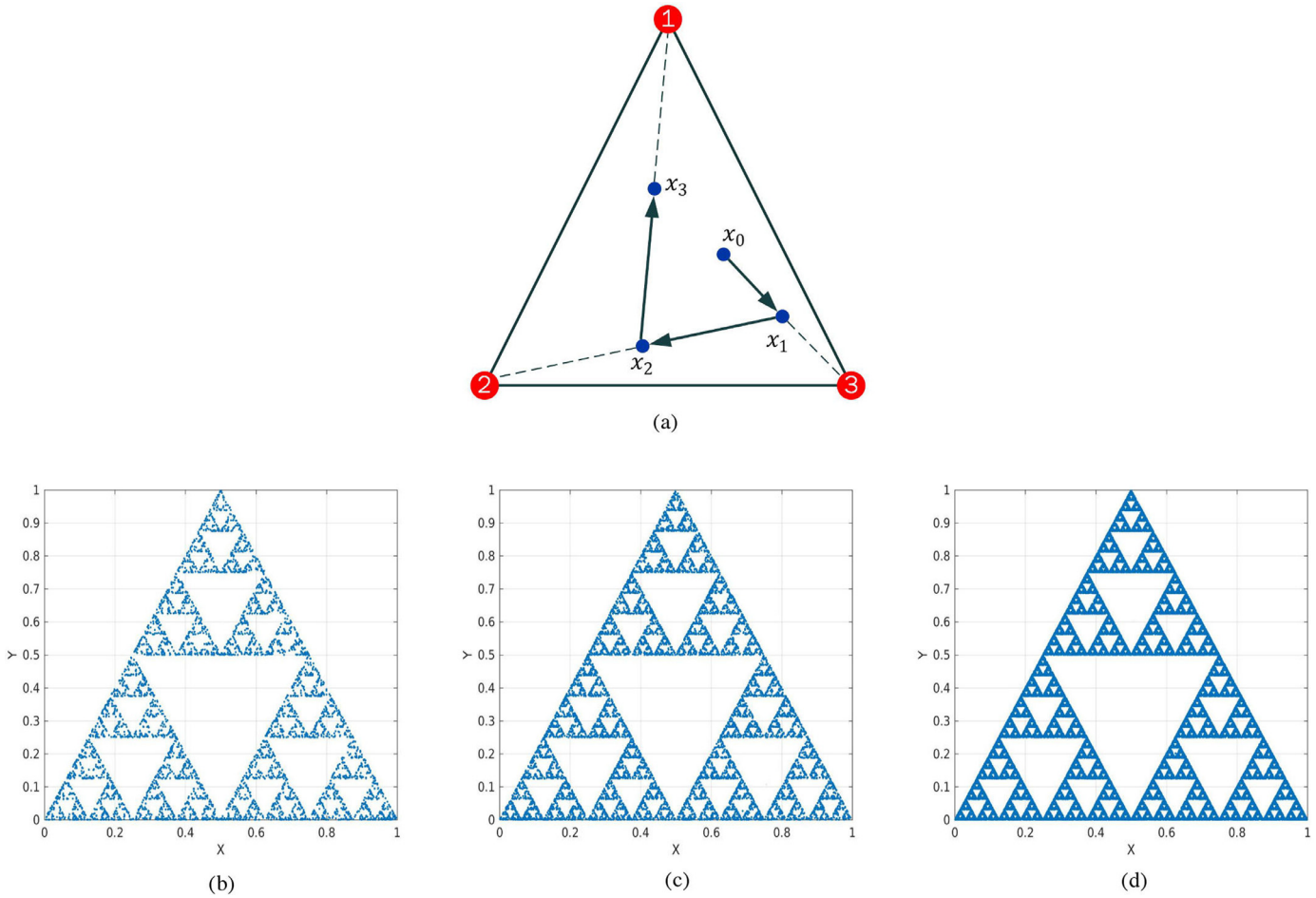

Fig. 1. (a) Chaos game reconstruction with an initial point (x_0) for 3-iterations, (b) 5000, (c) 10000 and (d) 50000 iterations.

2.3. Proposed deterministic 3D chaos game with cubic organization

The chaos game can be developed to different bases. However, the 4-base organization has better distribution in space and randomness features. Hence, this status has great importance. Due to the multi-dimensional space, the quadrilateral chaos game can be implemented with a number of 2^n bases with different dimensions for $n = 2, 3, \ldots$. The mathematical equation of the proposed 3D model is

$$\begin{cases} x_{n+1} = x_n + (B[k][1] - x_n) \times \mu \\ y_{n+1} = y_n + (B[k][2] - y_n) \times \mu \\ z_{n+1} = z_n + (B[k][3] - z_n) \times \mu \end{cases} \tag{3}$$

Where $x_0, y_0, z_0 \in [0, 1]$ and $\mu \in [0, 1]$ are real numbers and $k \in \{1, 2, ..., 8\}$ are bases address that are calculated

$$k = \lfloor x_n(1 - y_n)z_n \times 10^{14} \rfloor \; MOD \; 8$$

$$and \;\; B_{(BaseNo=8)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Fig. 4 illustrates the results of a cubic organization for proposed chaos game in a three-dimensional representation with various $\mu \in [0, 1]$ distances. In a three-dimensional view for $\mu = 0.70$ (Fig. 4(c)), the 3D space is divided into partitions with a specified interval, and this property is self-similarity, which is clearly visible.

2.4. Bifurcation diagram

The bifurcation is used to display of structural changes in the system orbit and was first introduced by *Henri Poincare'*. In fact, bifurcation investigates the structural changes of a system when changing parameters. In a dynamical system, fixed points can be removed or created, or its stability characteristics can be changed [72]. Representing of structural changes and examining the behavior of transition state in a dynamic system is considered to be the most important part of the evolution of that system [66].

The simulation results of the bifurcation diagram are shown in Fig. 5 based on control parameters(μ, α). As shown in Fig. 5(a,c,e), the proposed dynamical system is chaotic in $\mu \in [0, 0.55]$ and is periodic in $\mu \in [0.6, 1]$.Also, the proposed system is fully chaotic in $\mu \in [0.5, 0.55]$.As shown in figure Fig. 5(b,d,f), for $\alpha \in [0, 2]$, the dynamical behavior of the system is fully chaotic.

2.5. Lyapunov exponent

The Lyapunov exponent, taken from the name of the Russian mathematician *A.M.Lyapunov*, that shows the chaotic map sensitivity to the initial condition, which is represented by the λ parameter. Using Lyapunov exponent, we can show the convergence and divergence of two different circuits with different initial conditions. If λ is negative, then the paths close to each other will converge and the system won't have a chaotic state. However, if the λ is

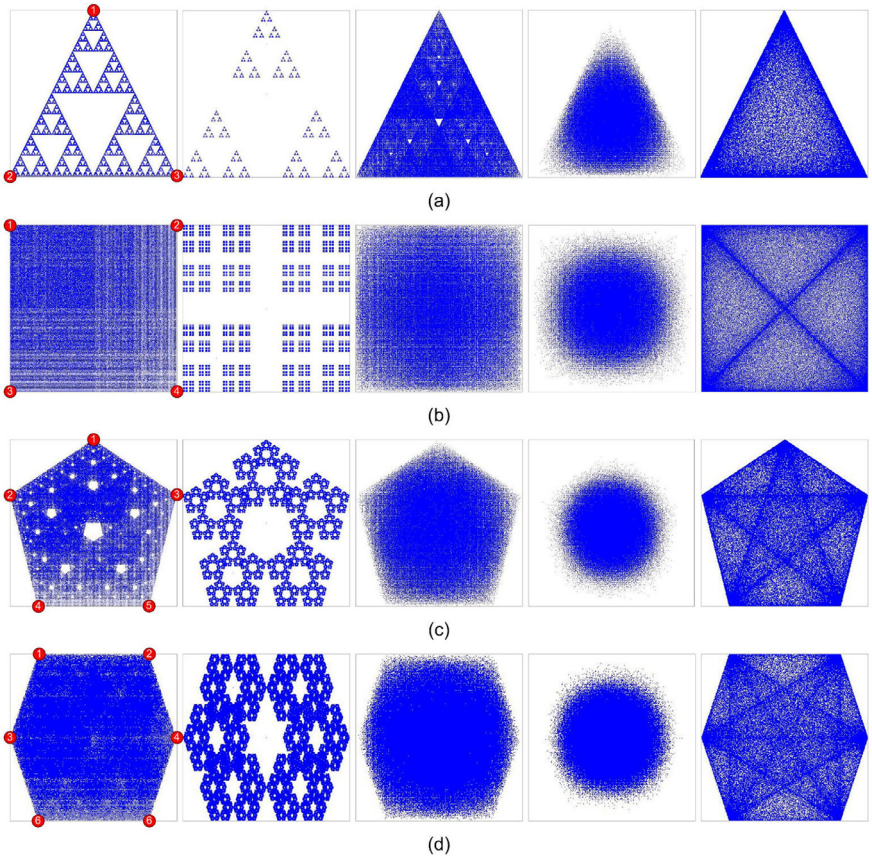

Fig. 2. Effect of Base Number and distance ratio (left to right $\mu = 0.5$, $\mu = 0.70$, $\mu = 0.35$, $\mu = 0.15$ *and* $\mu = rand \in [0, 1]$), (a) 3-Base (b) 4-Base (c) 5-Base (d) 6-Base.

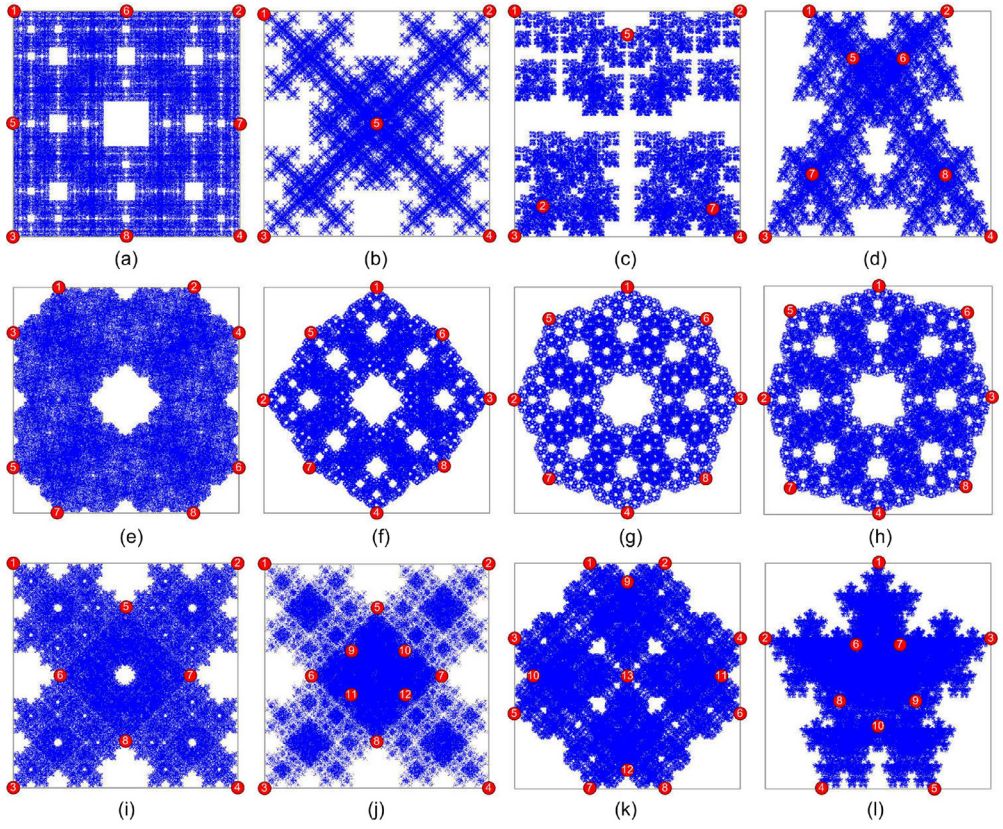

Fig. 3. Generation of fractal shapes by various base number in different locations ($\mu = 0.60$).

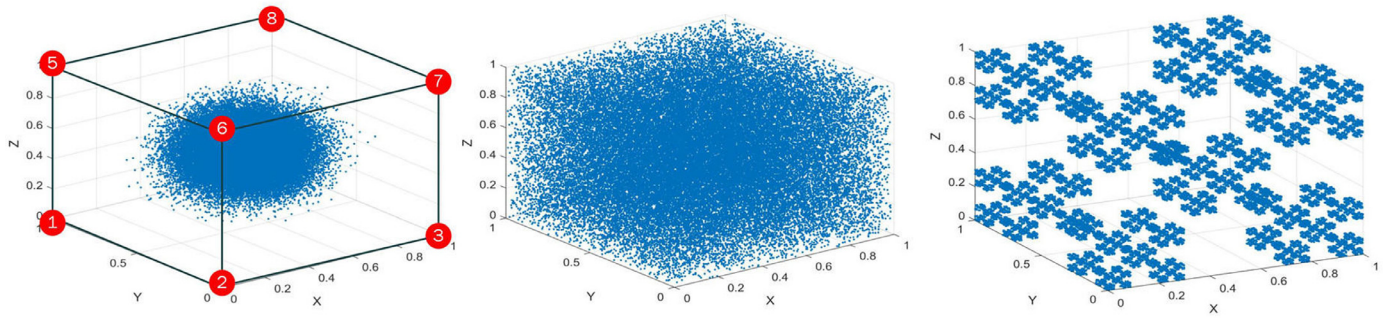

Fig. 4. The result of the chaos game on cubic organization (8-base), (a) $\mu = 0.10$, (b) $\mu = 0.50$, (c) $\mu = 0.70$.

Table 1
Configuration of Lyapunov exponent parameters and the obtained results from Fig. 6.

Lyapunov exponent	Initial conditions			Control parameters		Results		
	x_0	y_0	z_0	μ	α	Periodic	Chaotic	*fullychaotic*
$\lambda_1(x_0+\delta_0) = \frac{1}{10000}\sum_{n=1}^{10000}\log(\lvert\frac{f^n(x_0+\delta_0)-f^n(x_0)}{\delta_0}\rvert)$	0.6534+10^{-14}	0.3496	0.7135	$\mu \in [0, 1]$	$\alpha = 1.234$	$\mu \in [0.8, 1]$	$\mu \in [0, 0.8]$	$\mu \in [0.50, 0.55]$
$\lambda_2(y_0+\delta_0) = \frac{1}{10000}\sum_{n=1}^{10000}\log(\lvert\frac{f^n(y_0+\delta_0)-f^n(y_0)}{\delta_0}\rvert)$	0.6534	0.3496+10^{-14}	0.7135	$\mu \in [0, 1]$	$\alpha = 1.234$	$\mu \in [0.8, 1]$	$\mu \in [0, 0.8]$	$\mu \in [0.50, 0.55]$
$\lambda_3(z_0+\delta_0) = \frac{1}{10000}\sum_{n=1}^{10000}\log(\lvert\frac{f^n(z_0+\delta_0)-f^n(z_0)}{\delta_0}\rvert)$	0.6534	0.3496	0.7135+10^{-14}	$\mu \in [0, 1]$	$\alpha = 1.234$	$\mu \in [0.8, 1]$	$\mu \in [0, 0.8]$	$\mu \in [0.50, 0.55]$
$\lambda_4(x_0+\delta_0) = \frac{1}{10000}\sum_{n=1}^{10000}\log(\lvert\frac{f^n(x_0+\delta_0)-f^n(x_0)}{\delta_0}\rvert)$	0.6534+10^{-14}	0.3496	0.7135	$\mu = 0.51$	$\alpha \in [0, 2]$	-	$\alpha \in [0, 2]$	$\alpha \in [0, 2]$
$\lambda_5(y_0+\delta_0) = \frac{1}{10000}\sum_{n=1}^{10000}\log(\lvert\frac{f^n(y_0+\delta_0)-f^n(y_0)}{\delta_0}\rvert)$	0.6534	0.3496+10^{-14}	0.7135	$\mu = 0.51$	$\alpha \in [0, 2]$	-	$\alpha \in [0, 2]$	$\alpha \in [0, 2]$
$\lambda_6(z_0+\delta_0) = \frac{1}{10000}\sum_{n=1}^{10000}\log(\lvert\frac{f^n(z_0+\delta_0)-f^n(z_0)}{\delta_0}\rvert)$	0.6534	0.3496	0.7135+10^{-14}	$\mu = 0.51$	$\alpha \in [0, 2]$	-	$\alpha \in [0, 2]$	$\alpha \in [0, 2]$

positive, the paths close to each other will diverge and the system will have a chaotic state and it will be sensitive to initial conditions.

The several parameters for calculation of Lyapunov exponent is illustrated in Table 1. Lyapanov exponent measures (λ_1, ..., λ_6) are shown in Fig. 6 based on denoted parameters in Table 1. As shown in Fig. 6(a,c,e), the deterministic chaos game is chaotic in $\mu \in [0, 0.8]$ ($\lambda > 0$) and is periodic in $\mu \in [0.8, 1]$($\lambda < 0$). Also, the proposed system is fully chaotic in $\mu \in [0.5, 0.55]$.As shown in figure Fig. 6(b,d,f), for $\alpha \in [0, 2]$, the dynamical behavior of the proposed system is fully chaotic.

2.6. Sensitivity to initial condition and control parameter

In this section, we try to examine the sensitivity of the proposed system to very small changes in the amount of initial conditions and control parameters. For this reason, the correlation test between the two generated sequences from the original initial value and the modified initial value is used [73]. The mathematical equation for the correlation coefficients for the pairs of generated sequences (x, x'), (y, y') and (z, z') is expressed

$$C_{x,x'} = \frac{\sum_{i=1}^{N}(x_i-\bar{x}).(x'_i-\bar{x})}{[\sum_{i=1}^{N}(x_i-\bar{x})^2]^{\frac{1}{2}}.\left[\sum_{i=1}^{N}(x'_i-\bar{x'})^2\right]^{\frac{1}{2}}}$$

$$C_{y,y'} = \frac{\sum_{i=1}^{N}(y_i-\bar{y}).(y'_i-\bar{y})}{[\sum_{i=1}^{N}(y_i-\bar{y})^2]^{\frac{1}{2}}.\left[\sum_{i=1}^{N}(y'_i-\bar{y'})^2\right]^{\frac{1}{2}}}$$

$$C_{z,z'} = \frac{\sum_{i=1}^{N}(z_i-\bar{z}).(z'_i-\bar{z})}{[\sum_{i=1}^{N}(z_i-\bar{z})^2]^{\frac{1}{2}}\left[\sum_{i=1}^{N}(z'_i-\bar{z'})^2\right]^{\frac{1}{2}}} \quad (5)$$

For example, if $C_{x,x'} = \pm 1$, there is a strong correlation between x and x', and if $C_{x,x'} = 0$, there is no correlation between the two sequences. The calculated correlation coefficients on the very small changes (10^{-14}) in the initial conditions (x_0, y_0, z_0) and the control parameters(α, μ) are shown in Table 2. The results obtained from 200 iterated steps in this table, indicate that the proposed dynamical system is highly sensitive to the initial conditions and the control parameter.

Fig. 7 shows the plot obtained from the very small change in the initial value of x_0 on the three sequences x, y and z.

2.7. Cobweb plot

In dynamic systems, the Cobweb plot is used to represent the number of successful iterations of a function such as $y = f(x)$. In fact, a line is plotted between two points [(x, $f(x)$), ($f(x)$, $f(f(x))$], and this iteration is continued for other generated values. In the interpretation of the cobweb plot, we can say that an inward spiral is the result of a stable fix point, and a rectangle is a period of two orbits. A chaotic orbit consists of a number of non-repeatable values that can be seen in “filled out” form.

Fig. 8 shows the cobweb plot of the x sequence for different values of μ. As seen in the Fig. 8, for $\mu = 0.81$, the result of the graph is a periodic form, and in $\mu = 0.50$ the cobweb plot is fully chaotic.

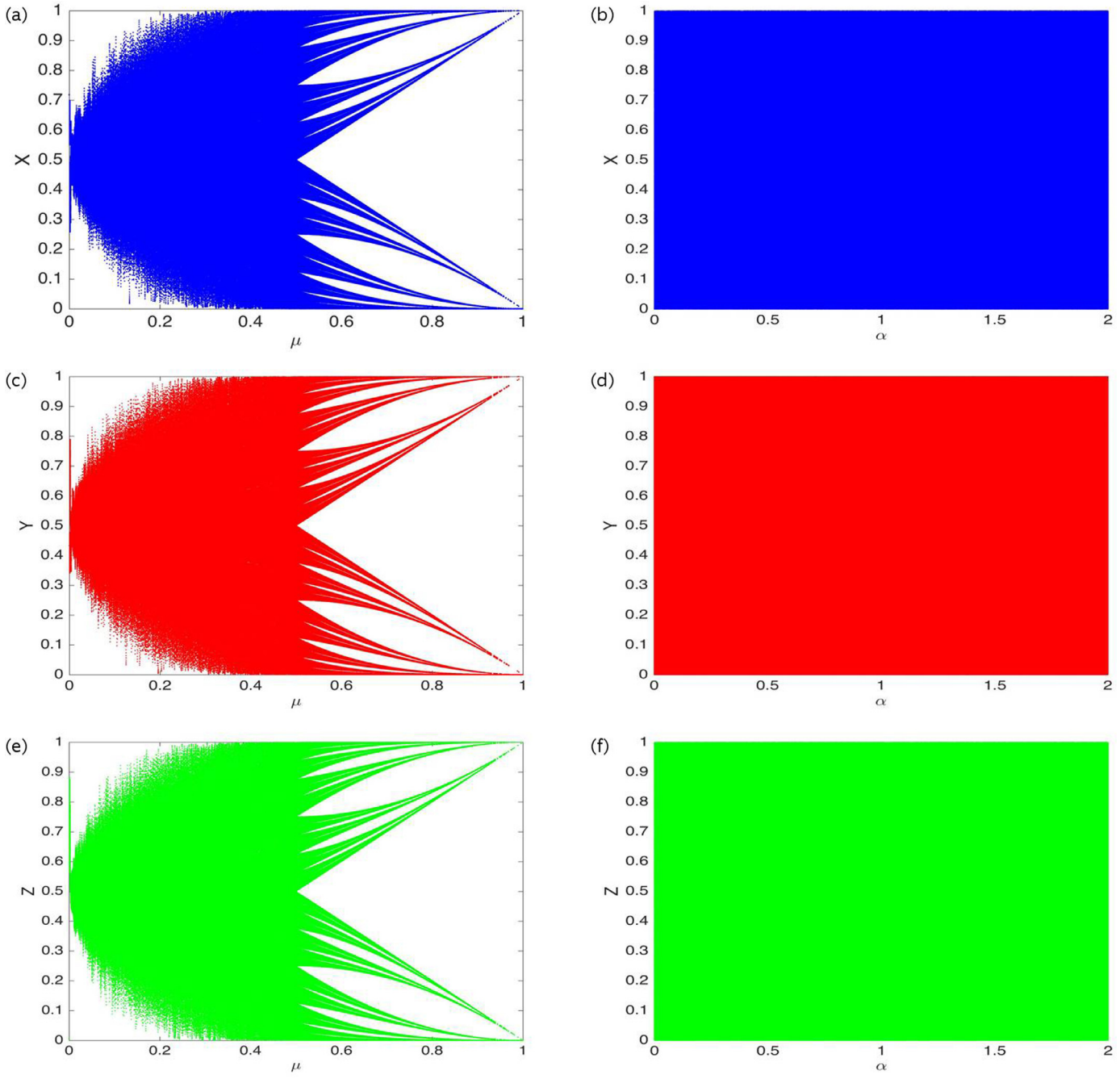

Fig. 5. Bifurcation diagram in cubic organization for control parameters (μ, α), (a-b) x sequence (c-d) y sequence (e-f) z sequence.

Table 2
Sensitivity Analysis for very small changes in initial conditions and control parameters.

Initial conditions						Control parameters		Correlation Coefficients		
x_0	x'_0	y_0	y'_0	z_0	z'_0	α	μ	$C_{x,x'}$	$C_{y,y'}$	$C_{z,z'}$
$x_0 = 0.6534$	$x'_0 = x_0 + 10^{-14}$	$y_0 = 0.3496$	$y'_0 = y_0$	$z_0 = 0.7135$	$z'_0 = z_0$	$\alpha = 1.234$	$\mu = 0.51$	-0.030055	0.103294	-0.0251208
$x_0 = 0.6534$	$x'_0 = x_0$	$y_0 = 0.3496$	$y'_0 = y_0 + 10^{-14}$	$z_0 = 0.7135$	$z'_0 = z_0$	$\alpha = 1.234$	$\mu = 0.51$	-0.032904	0.030389	-0.168912
$x_0 = 0.6534$	$x'_0 = x_0$	$y_0 = 0.3496$	$y'_0 = y_0$	$z_0 = 0.7135$	$z'_0 = z_0 + 10^{-14}$	$\alpha = 1.234$	$\mu = 0.51$	-0.030055	0.103294	-0.0251208
$x_0 = 0.6534$	$x'_0 = x_0$	$y_0 = 0.3496$	$y'_0 = y_0$	$z_0 = 0.7135$	$z'_0 = z_0$	$\alpha = 1.234 + 10^{-13}$	$\mu = 0.51$	0.156399	0.019085	0.137711
$x_0 = 0.6534$	$x'_0 = x_0$	$y_0 = 0.3496$	$y'_0 = y_0$	$z_0 = 0.7135$	$z'_0 = z_0$	$\alpha = 1.234$	$\mu = 0.51 + 10^{-14}$	-0.035268	0.131318	-0.009227

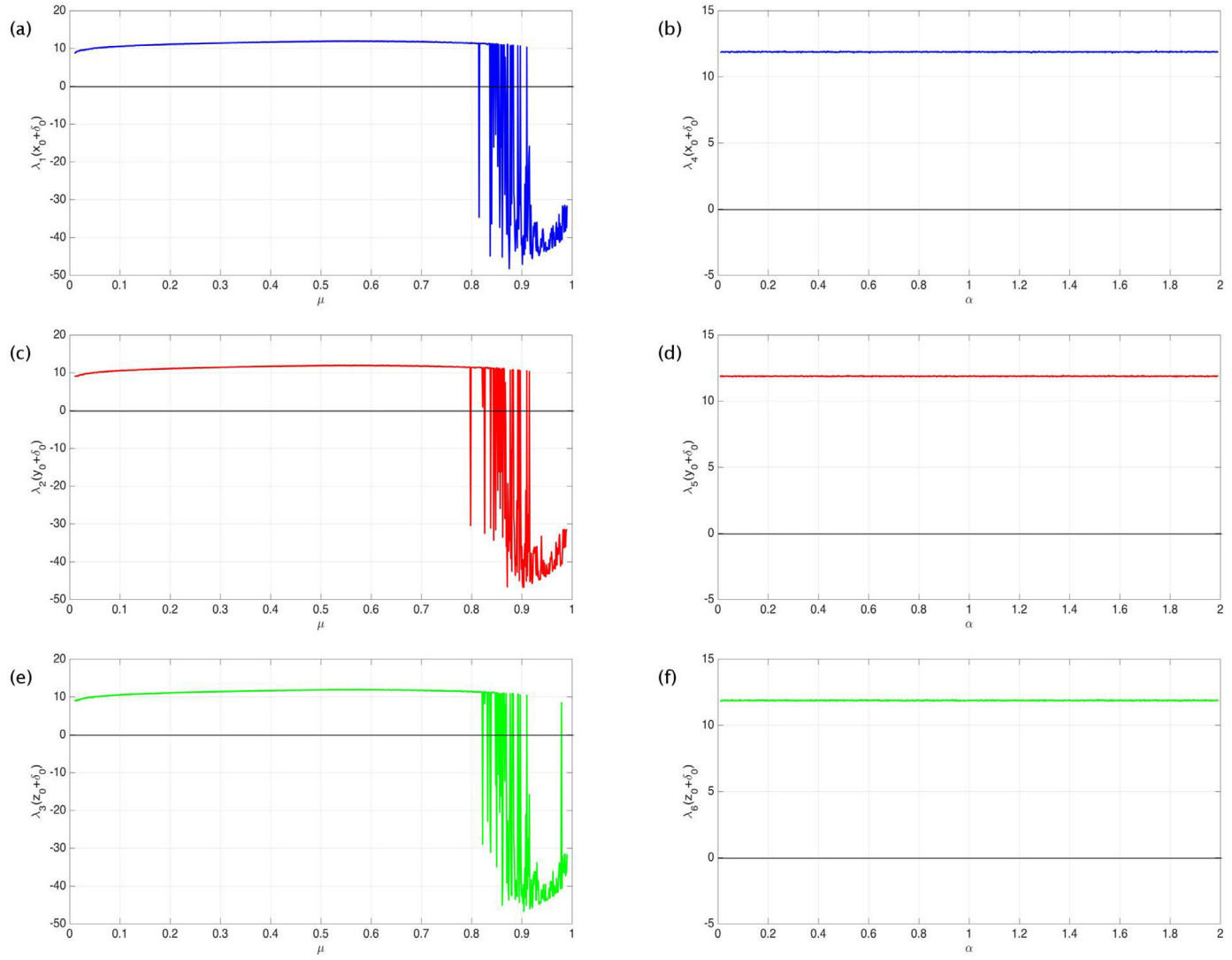

Fig. 6. The results of Lyapunov exponent analysis based on configured parameters in Table 1 (a) $\lambda_1(x_0+\delta_0)$, (b) $\lambda_2(x_0+\delta_0)$ (c) $\lambda_3(y_0+\delta_0)$, (d) $\lambda_4(y_0+\delta_0)$, (e) $\lambda_5(z_0+\delta_0)$, (f) $\lambda_6(z_0+\delta_0)$.

2.8. Randomness test

A pseudo-random number generator needs to pass through a set of randomness statistical tests that can be identified as a secure sequence in the cryptographic system. In this paper, four common statistical tests (NIST, Diehard, ENT, and TESTU01) will be used to prove the randomness of chaotic sequences. Test suites of NIST, ENT, and Diehard have used a file containing the sequence of 32-bit strings as the test input. But in TESTU01, the chaos game function is considered as a code programmed in the C language to the input of this test.In these tests, the results on the *x, y*, and *z* sequences are calculated independently. In the following, we will have a brief overview of the statistical tests.

- **NIST:** The NIST[1] statistical test set contains 15 tests that are used to test the randomness of binary sequences. These sequences can be generated by software or random encryption hardware or pseudo-random number generators. The NIST test was released in 2001, which is the result of the collaboration between the computer security department and the NIST statistics department [10]. The results of this test on the chaos game are shown in Table 3.
- **Diehard:** The Diehard[2] tests include a group of statistical tests to evaluate the quality of random numbers. This set of experiments was published in 1995 by the *Marsaglia* from the University of Florida for the first time on a CD-ROM of random numbers [11,12]. Diehard's collection includes 18 several tests. The results of the Diehrad test suite are presented in the Table 3.
- **ENT:** The ENT[3] program, presented by *John Walker* in 1998, is another set of statistical tests used to evaluate the randomness of sequences [13]. The ENT program is a useful tool for evaluating pseudo-random generators that are used in cryptographic science, compression algorithms and statistical sampling, which requires a density of information. Table 3 shows the results of these experiments
- **TestU01:** TestU01 is a software library that is implemented in ANSI C. This test is a set of tools for the experimental statistical testing of the uniform random generators [14].

[1] National Institute of Standards and Technology, Computer code available at. URL http://csrc.nist.gov/rng/SP800-22b.pdf.

[2] Computer code of DIEHARD, URL available at: http://stat.fsu.edu/pub/diehard/.

[3] ENT, A Pseudo Random Number Sequence Test Program, available at: http://www.fourmilab.ch/random/.

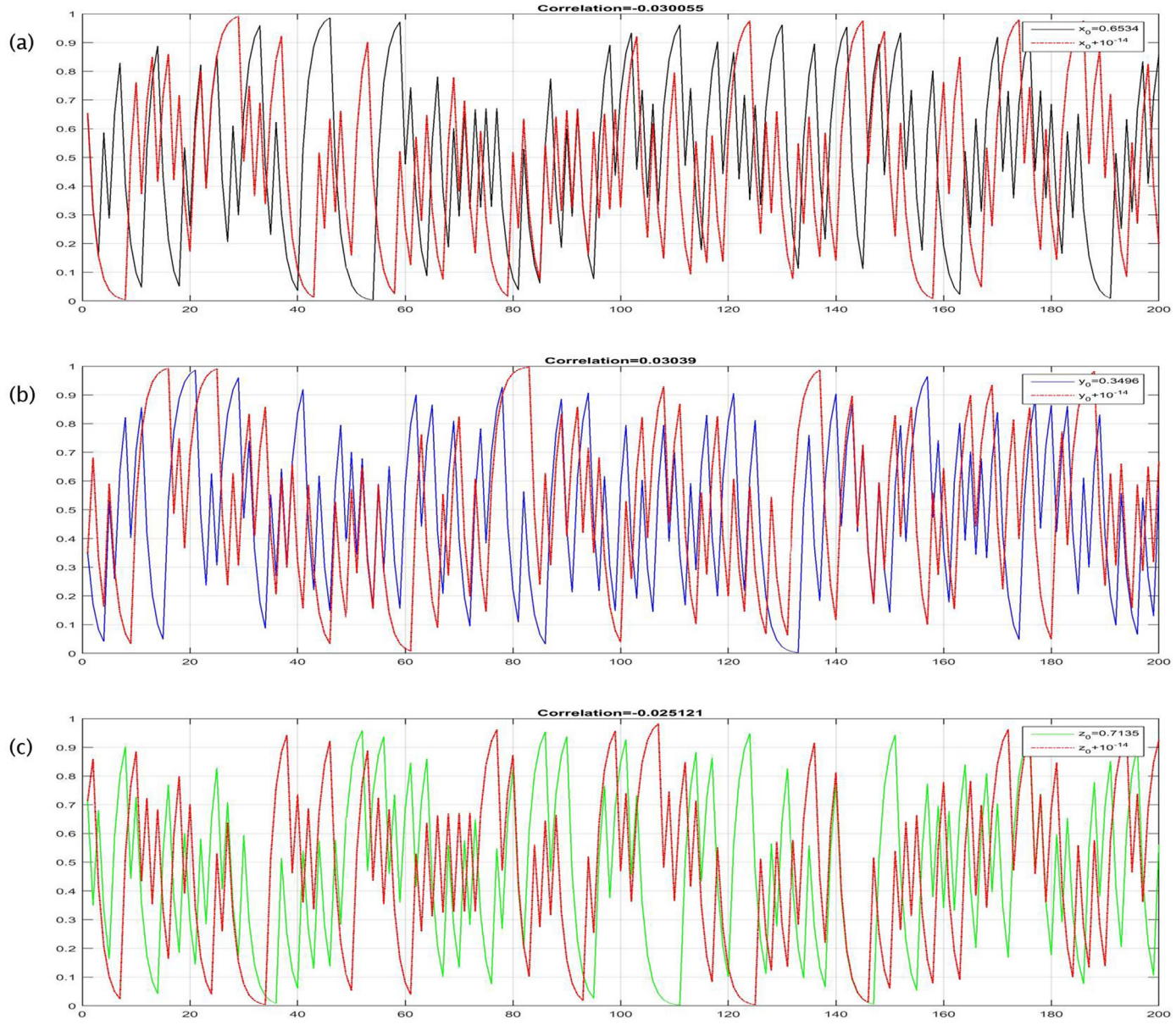

Fig. 7. Sensitivity to small changes (10^{-14}) in initial conditions for : (a) $(x_0, x_0 + 10^{-14})$ sequences, (b) $(y_0, y_0 + 10^{-14})$ sequences, (c) $(z_0, z_0 + 10^{-14})$ sequences.

This library presents several types of random number generators in general, which widely used in the literature of research. Three batteries of statistical tests are implemented by TESTU01: Big Crush (45 tests), Crush (60 tests) and Small Crush (10 tests). Table 3 shows the results of these tests.

2.9. Key space analysis

Sensitivity to initial conditions and control parameters is one of the most important characteristics of chaotic sequences. In fact, the initial conditions and control parameters are used as keys in the encryption system. In high-key encrypted systems, the attacker will need more time to search for key space. In encryption systems, the key space greater than *128 bits* is safe [74]. Table 4 shows how to calculate the key space in the proposed chaotic sequences. Based on this table, the calculated key is *232 bits*, which will be sufficiently secure.

3. Proposed ROI image encryption

The proposed encryption and decryption process for a region of interest in the digital image is described by the following steps:

- **Step 1:** The input image, the initial conditions (x_0, y_0, z_0), the control parameters (μ, α), the bases of chaos game(B[.][.]) and the maximum iteration(*MaxItr*) are entered as input parameters by users.
- **Step 2:** A three-dimensional chaotic map with cubic organizing will run 100 times to produce values within the region of interest.
- **Step 3:** The chaotic sequence is selected for two pixels, and then two pixels are permuted together. The variables (x_i, y_i) are used to select the location of the pixels. To change the pixel value, the pixel intensity is changed by the z_i (XOR operation). A mask (M) is used to the selection process of pixels, so that pixels do not repeat in the encryption and decryption process.

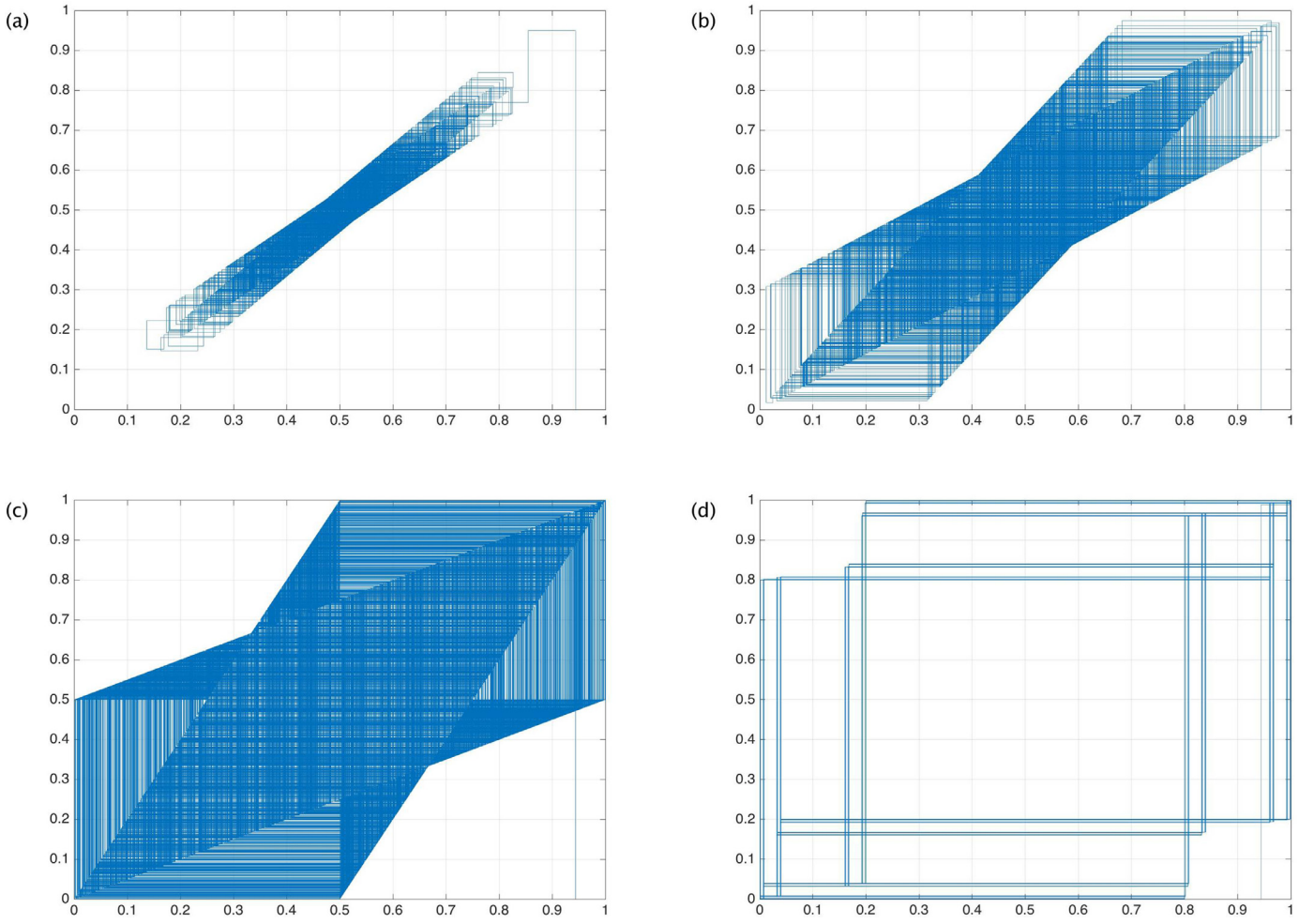

Fig. 8. Cobweb plot of x sequence for (a) $\mu = 0.10$, (b)$\mu = 0.30$, (c) $\mu = 0.50$, (d) $\mu = 0.81$.

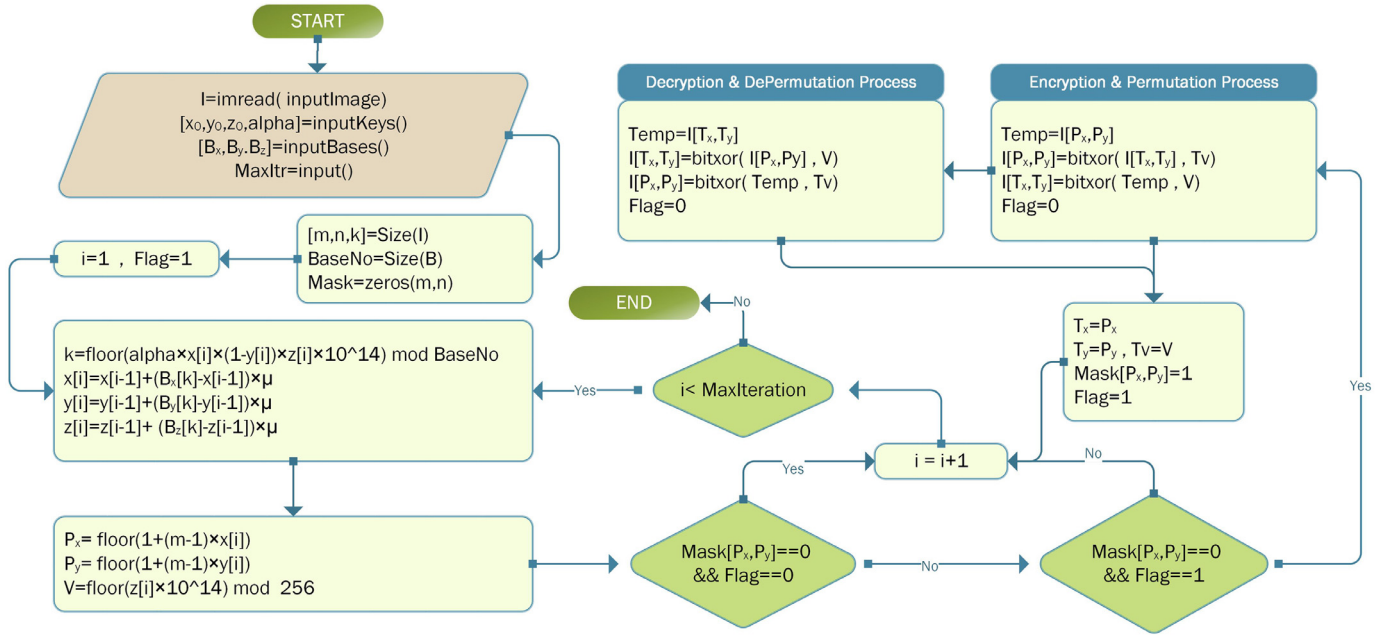

Fig. 9. Flowchart of proposed ROI image encryption & decryption.

Table 3
The results of randomness test suites on proposed PRNG.

NIST : Test name	x			y			z		
	P-Value	Proportion	Result	P-Value	Proportion	Result	P-Value	Proportion	Result
Frequency Test	0.534938	100/100	Success	0.838077	100/100	Success	0.461456	100/100	Success
Block Frequency Test (m=128)	0.246887	100/100	Success	0.876032	100/100	Success	0.341397	100/100	Success
Cumulative-Forward	0.851256	100/100	Success	0.605764	100/100	Success	0.396377	100/100	Success
Cumulative-Reverse	0.578499	100/100	Success	0.436805	100/100	Success	0.730385	100/100	Success
Run Test	0.827371	100/100	Success	0.70784	100/100	Success	0.209223	100/100	Success
Long Runs of Ones	0.223972	100/100	Success	0.848257	100/100	Success	0.807929	100/100	Success
Rank	0.27824	100/100	Success	0.617716	100/100	Success	0.058802	100/100	Success
Spectral DFT	0.778984	100/100	Success	0.453578	100/100	Success	0.503909	100/100	Success
Non-Overlapping-Min	0.579255	100/100	Success	0.451964	100/100	Success	0.209223	100/100	Success
Non-Overlapping-Max	0.579255	100/100	Success	0.451964	100/100	Success	0.209223	100/100	Success
Overlapping Temp. (m=9)	0.303878	100/100	Success	0.829661	100/100	Success	0.182693	100/100	Success
Universal	0.77012	100/100	Success	0.578368	100/100	Success	0.732962	100/100	Success
Approximation Entropy (m=10)	0.533902	100/100	Success	0.759071	100/100	Success	0.541897	100/100	Success
Random Excursions-Min	0.790371	100/100	Success	0.560561	100/100	Success	0.455068	100/100	Success
Random Excursions-Max	0.790371	100/100	Success	0.560561	100/100	Success	0.455068	100/100	Success
Random Excursions Variant-Min	0.020442	100/100	Success	0.693646	100/100	Success	0.378442	100/100	Success
Random Excursions Variant-Max	0.020442	100/100	Success	0.693646	100/100	Success	0.378442	100/100	Success
Serial (1)	0.028293	100/100	Success	0.929005	100/100	Success	0.563538	100/100	Success
Serial (2)	0.028293	100/100	Success	0.929005	100/100	Success	0.563538	100/100	Success
Linear Complexity (M=500)	0.852094	100/100	Success	0.38483	100/100	Success	0.693021	100/100	Success

Diehard : Test Name	x		y		z	
	P-Value	Result	P-Value	Result	P-Value	Result
Diehard Birthdays	0.02340644	Success	0.39965486	Success	0.53923813	Success
Binary Rank 32 × 32	0.20890044	Success	0.42381923	Success	0.86097888	Success
Binary Rank 31 × 31	0.37567665	Success	0.73578776	Success	0.57653232	Success
Binary Rank 6 × 8	0.82861985	Success	0.26149849	Success	0.49735193	Success
Overlapping 5-Permutation	0.97590675	Success	0.944299	Success	0.04255836	Success
Bitstream	0.6585842	Success	0.19812604	Success	0.33542862	Success
Overlapping Pairs Sparse Occupancy	0.46584941	Success	0.60129876	Success	0.91342186	Success
Overlapping Quadruples Sparse Occupancy	0.37409842	Success	0.95709138	Success	0.97761646	Success
DNA	0.78077457	Success	0.69231782	Success	0.48462384	Success
Count The 1's Test a Stream of Bytes	0.13666287	Success	0.77441116	Success	0.20012025	Success
Count The 1's Test a Specific of Bytes	0.3840272	Success	0.93934917	Success	0.94662068	Success
Parking Lot	0.08478625	Success	0.77490362	Success	0.20045759	Success
Minimum Distance	0.36045554	Success	0.40568184	Success	0.42143695	Success
3D-Sphere	0.32447131	Success	0.99900354	Success	0.86652261	Success
Squeeze	0.45135904	Success	0.74639153	Success	0.57777117	Success
Sums	0.20912717	Success	0.41349664	Success	0.01076553	Success
Runs	0.65484546	Success	0.18635408	Success	0.08124332	Success
Craps	0.5842986	Success	0.79812833	Success	0.65262916	Success

ENT : Test Name	x		y		z	
	Value	Result	Value	Result	Value	Result
Entropy	7.999953	Success	7.999948	Success	7.999952	Success
Chi square	259.72	Success	285.81	Success	266.86	Success
Arithmetic mean	127.4522	Success	127.4554	Success	127.4828	Success
Monte Carlo	3.142029142	Success	3.143349143	Success	3.14026514	Success
Serial correlation coefficient	0.000346	Success	-0.000323	Success	0.000005	Success

TestU01 : Battery	Parameters	Number of	Results		
		Statistics	x	y	z
Small Crush	Standard	15	Pass	Pass	Pass
Crush	Standard	144	Pass	Pass	Pass
Big Crush	Standard	160	Pass	Pass	Pass

- **Step 4:** Step 4: Depending on the type of encryption or decryption process, the pixel permutation, and substitution process are different, and the details are fully illustrated in Algorithm. 1.
- **Step 5:** Stages 3 and 4 run at maximum iteration so that the encrypted or decrypted image can be generated as output.

Flowchart of the proposed method for encryption and decryption an image is shown in Fig. 9. The complete pseudo-code of the encryption and decryption process is shown in Algorithm 1.

4. Experimental results

4.1. Image database

To simulate the proposed algorithm on a region of interest, a number of non-standard images are extracted from the Internet. Various regions of the image were selected and various fractals were encrypted on the original images. The results of these encrypted and decrypted images are clearly seen in Fig. 10 and Fig. 11.

Algorithm 1 Proposed pseudo-code for encryption and decryption process.

function J = ENCRYPT / DECRYPT($I, B[][], MaxItr, x_0, y_0, z_0, \mu, \alpha$)
  % x_0, y_0, $z_0 \in [0,1]$, $\mu \in [0.5, 0.55]$ $\alpha \in [1,4]$]
  % I : *InpulmatgE* , J : *Encrypted or Decrypted Image*
  $[m, n, k] = GetSize(I)$
  $BaseNo = size(B)$
  $Mask = zeros(m, n)$
  for $j = 1$ to 100 **do**
    $k = \lfloor \alpha \times x_j \times (1 - y_j) \times z_j \rfloor \ MOD \ BaseNo$
    $x_{j+1} = x_j + (B[k][1] - x_j) \times \mu$
    $y_{j+1} = y_j + (B[k][2] - y_j) \times \mu$
    $z_{j+1} = z_j + (B[k][3] - z_j) \times \mu$
  end for
  $x_0 = x_j, y_0 = y_j, z_0 = z_j$
  $i = 1, Flag = 1$
  while $i < MaxItr$ **do**
    $k = \lfloor \alpha \times x_i \times (1 - y_i) \times z_i \rfloor \ MOD \ BaseNo$
    $x_{i+1} = x_i + (B[k][1] - x_i) \times \mu$
    $y_{i+1} = y_i + (B[k][2] - y_i) \times \mu$
    $z_{i+1} = z_i + (B[k][3] - z_i) \times \mu$
    $P_x = \lfloor x_{i+1} \times 10^{14} \rfloor \ MOD \ m$
    $P_y = \lfloor y_{i+1} \times 10^{14} \rfloor \ MOD \ n$
    $V = \lfloor z_{i+1} \times 10^{14} \rfloor \ MOD \ 256$
    if $Mask[P_x][P_y] == 0$ && $Flag == 0$ **then**
      $T_x = P_x$
      $T_y = P_y$
      $T_v = V$
      $Flag = 1$
    else if $Mask[P_x][P_y] == 0$ && $Flag == 1$ **then**
      if *EncryptionProcess* **then**
        $Temp = I[P_x][P_y]$
        $J[P_x][P_y] = I[T_x][T_y] \oplus T_v$
        $J[T_x][T_y] = Temp \oplus V$
        $Flag = 0$
      else if *DecryptionProcess* **then**
        $Temp = I[T_x][T_y]$
        $J[T_x][T_y] = I[P_x][P_y] \oplus T_v$
        $J[P_x][P_y] = Temp \oplus T_v$
        $Flag = 0$
      end if
    end if
    $i = i + 1$
  end while
  Return J
end function

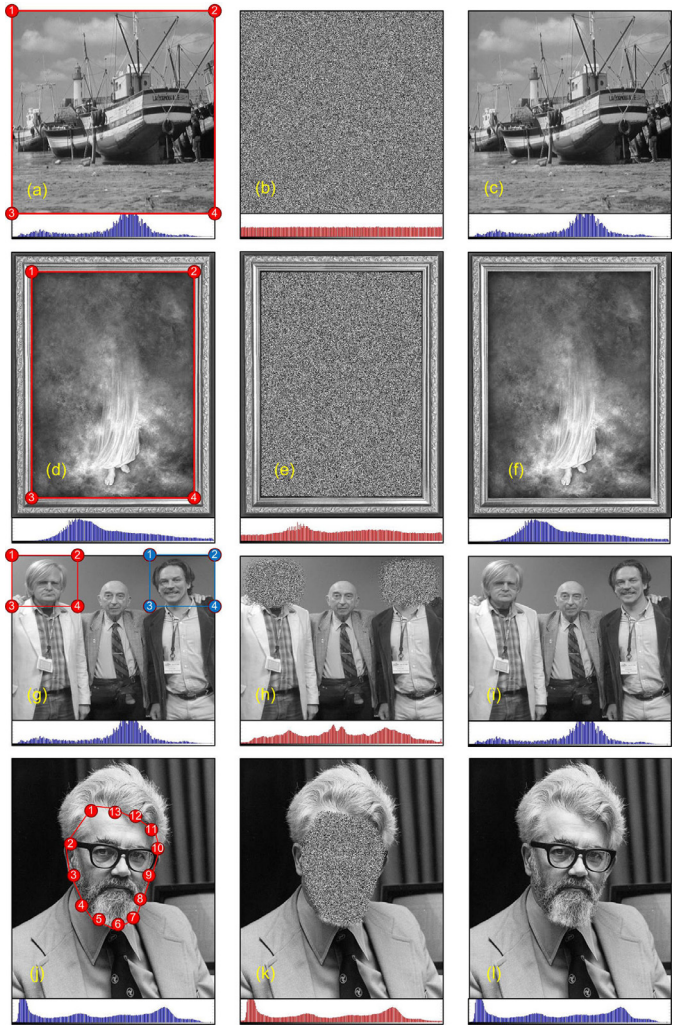

Fig. 10. The results of gray-scale encrypted & decrypted images based on the selected region of interest, (a-c) Selection of all pixels in the image as an ROI, (d-f) Rectangular ROI selection (g-i) Multiple ROI Selection (j-l) Face Selection.

Table 4
The key space Of proposed Chaos Game.

Input parameters	Best range	Precision	
		float	binary
μ	[0.5,0.55]	10^{-13}	2^{45}
α	[0,2]	10^{-14}	2^{47}
x_0	[0,1]	10^{-14}	2^{46}
y_0	[0,1]	10^{-14}	2^{46}
z_0	[0,1]	10^{-14}	2^{46}
Key Space			2^{232}

To evaluate the quality of the proposed algorithm, the Kodak and SIPI[4] datasets are used. The Kodak image collection contains 25 uncompressed PNG images, which size of each image is 768 × 512 pixels [76]. The USC-SIPI collection was first presented in 1977, which includes digitized images for researchers in the field of image processing. From SIPI collection, the nine classic color images have been selected [75]. The image file name and size of each image are shown in Table 5, which separates each image as a dataset name. Selected images from the SIPI database which used to simulating of the proposed algorithm are shown in Fig. 12.

4.2. Software and hardware platform

We have used *Matlab 2017a* to implement the proposed algorithm in this article. A graphical user interface is also designed using Matlab, as shown in Fig. 13. The hardware to simulate a personal computer with its MS-Windows OS has a *CORE-i5 CPU* with *8GB RAM*.

4.3. Histogram analysis

Histogram analysis is one of the algorithm evaluation tools. The histogram is the statistical frequency of pixels. The histogram shows the illustration of the pixel distribution in the image that represents the number of pixels. A good encryption algorithm is an algorithm that collapses the image in such a way that its features are not as visually recognizable, and no information from the original image can be extracted from the comparison of the original image and the encrypted image. If there is no statistical similarity in the histogram of an image, then the attacker will not succeed in extracting the original image.

[4] SIPI Image Database, Volume 3: Miscellaneous (1977 (accessed 2018)). URL http://sipi.usc.edu/database/database.php? volume=misc

Table 5
The experimental results for proposed method after permutation and encryption processes based on the database images.

Dataset	Images	Original image				Encrypted image					
		Size	Entropy	Correlation			Entropy	SSIM	PSNR	Correlation	
				H	V	D				H	V	D
SIPI [75]	Baboon	512 × 512	7.6444	0.8660	0.9159	0.8516	7.9993	0.0091	8.7814	0.0224	0.0115	-0.0025
	Couple 1	256 × 256	6.0483	0.9510	0.9497	0.9158	7.9973	0.0044	6.2539	0.0221	0.0041	-0.0115
	F-16	512 × 512	6.5768	0.9648	0.9750	0.9414	7.9993	0.0104	7.9727	0.0144	-0.0031	0.0020
	Girl 1	256 × 256	6.4155	0.9628	0.9721	0.9479	7.9972	0.0065	7.2840	0.0114	-0.0012	0.0053
	Girl 2	256 × 256	5.6002	0.9195	0.9770	0.9045	7.9973	0.0111	9.9452	0.0319	0.0022	0.0171
	Girl 3	256 × 256	7.1026	0.9885	0.9839	0.9712	7.9971	0.0114	8.8503	0.0259	-0.0090	-0.0522
	House 1	256 × 256	6.4007	0.9423	0.9681	0.9136	7.9969	0.0108	8.9361	0.0025	-0.0069	-0.0037
	House 2	512 × 512	7.3602	0.9589	0.9632	0.9233	7.9993	0.0103	8.4837	0.0048	0.0113	-0.0129
	Jelly Beans 1	256 × 256	5.8346	0.9787	0.9718	0.9533	7.9974	0.0107	8.5887	0.0037	-0.0087	0.0362
	Jelly Beans 2	256 × 256	6.2700	0.9752	0.9742	0.9546	7.9974	0.0107	8.6513	0.0012	0.0104	0.0016
	Lena	512 × 512	7.2719	0.9882	0.9805	0.9653	7.9993	0.0100	8.6157	-0.0031	-0.0293	0.0077
	Peppers	512 × 512	7.2978	0.9675	0.9693	0.9635	7.9993	0.0093	8.0812	0.0282	0.0122	-0.0056
	Sailboat	512 × 512	7.3896	0.9553	0.9595	0.9490	7.9993	0.0095	8.0807	-0.0037	-0.0152	0.0191
	Splash	512 × 512	6.6530	0.9978	0.9949	0.9918	7.9993	0.0092	7.6313	0.0048	0.0158	0.0100
	Tiffany	512 × 512	5.8101	0.9746	0.9639	0.9484	7.9993	0.0104	7.0772	0.0109	-0.0023	0.0012
	Tree	256 × 256	7.1816	0.9419	0.9562	0.9265	7.9969	0.0100	8.1577	-0.0034	-0.0065	0.0003
KODAK [76]	alfons rudolph	512 × 768	7.6366	0.9215	0.8848	0.8130	7.9994	0.0096	8.4021	-0.0091	-0.0199	0.0072
	barn and pond	512 × 768	7.1877	0.9586	0.9558	0.9343	7.9995	0.0106	9.1075	0.0316	-0.0159	0.0040
	bob clemens	768 × 512	7.2118	0.9532	0.9576	0.9338	7.9995	0.0097	8.7814	-0.0053	0.0070	-0.0181
	couple on beach	512 × 768	7.0286	0.9609	0.9747	0.9477	7.9995	0.0112	8.8429	0.0042	-0.0408	-0.0204
	girl with painted face	512 × 768	7.3050	0.9903	0.9834	0.9754	7.9994	0.0086	7.0860	0.0003	0.0190	-0.0244
	hats	512 × 768	7.1256	0.9720	0.9815	0.9740	7.9996	0.0097	8.8102	-0.0177	-0.0388	0.0234
	lighthouse in Maine	768 × 512	7.3641	0.9512	0.9336	0.9035	7.9995	0.0099	9.0644	0.0132	-0.0203	-0.0107
	model in black dress	768 × 512	6.9911	0.9362	0.9236	0.8848	7.9995	0.0082	7.7963	-0.0107	-0.0135	-0.0057
	monument	768 × 512	7.3095	0.9686	0.9673	0.9546	7.9994	0.0080	7.9450	-0.0113	-0.0237	0.0026
	mountain chalet	512 × 768	7.0864	0.9324	0.9464	0.9099	7.9995	0.0098	8.6489	0.0097	-0.0151	0.0071
	mountain stream	512 × 768	7.4468	0.8448	0.8987	0.8202	7.9995	0.0096	8.5087	0.0101	-0.0037	-0.0189
	off-shore sailboat race	768 × 512	7.1476	0.9704	0.9564	0.9372	7.9995	0.0107	9.6705	0.0060	-0.0060	-0.0071
	p51 Mustang	512 × 768	5.8297	0.9901	0.9918	0.9821	7.9995	0.0089	6.2941	0.0031	-0.0055	0.0013
	portland head light	512 × 768	7.0229	0.8980	0.9502	0.8678	7.9995	0.0102	9.3131	0.0015	-0.0206	-0.0114
	red door	512 × 768	5.6608	0.9275	0.9313	0.8899	7.9994	0.0080	7.7654	0.0102	-0.0129	0.0066
	sailboat at anchor	512 × 768	7.4158	0.9488	0.9791	0.9348	7.9995	0.0100	8.5583	-0.0113	0.0059	0.0192
	sailboat at pier	512 × 768	6.9619	0.9182	0.9353	0.8794	7.9995	0.0092	8.7302	0.0255	0.0314	-0.0047
	sailboats under spinnakers	768 × 512	7.1137	0.9649	0.9567	0.9285	7.9995	0.0110	9.7351	0.0019	-0.0090	-0.0004
	shuttered windows	512 × 768	7.0662	0.9598	0.9674	0.9377	7.9995	0.0105	9.2648	0.0081	-0.0107	0.0032
	steve kelly	512 × 768	7.4019	0.9055	0.9088	0.8717	7.9995	0.0087	8.0699	-0.0051	0.0059	-0.0127
	stone building	512 × 768	7.1670	0.7903	0.8535	0.7029	7.9995	0.0107	9.2661	0.0328	0.0142	-0.0155
	tropical key	512 × 768	7.2435	0.9463	0.9831	0.9423	7.9995	0.0099	9.1204	-0.0025	0.0081	-0.0216
	two macaws	512 × 768	7.3721	0.9869	0.9898	0.9814	7.9995	0.0095	8.4059	0.0067	0.0290	0.0338
	white water rafters	512 × 768	7.4783	0.9494	0.9584	0.9293	7.9995	0.0089	8.3312	-0.0107	0.0281	-0.0119
Mean			6.9358	0.9470	0.9561	0.9214	7.9990	0.0096	8.4227	0.0111	0.0139	0.0120

Histogram analysis describes how to distribute pixels by showing the number of each intensity.The cryptographic algorithm should be designed in such a way that it does not present any information for a statistical attack, and it will be the best algorithm when the statistical distribution is uniform. Fig. 14 shows the histogram of the original image and the encrypted image by selecting the entire image as a region of interest. The difference between the histograms of the original image and the encrypted image is clearly visible according to Fig. 14(a) and Fig. 14(b). Also, the uniformity of the histogram of the encrypted image shows the strength of the proposed algorithm in changing the values of pixels in the image.

Also, Fig. 15 shows a three-dimensional histogram of an original and encrypted image by separating the most and least significant bits in pixel values.

4.4. Spatial characteristic

Spatial analysis is very similar to histogram analysis. A spatial graph has three dimensions x, y, and z, whose axes x and y contain the pixel address an image, and its z-axis is the brightness intensity of that pixel. The spatial diagram of an original and encrypted image is shown in Fig. 16. Based on this figure, it can be concluded that encrypted pixels are uniformly distributed in several directions of space. The uniformity of this distribution makes the attackers do not extract useful information from the original image.

4.5. Visual quality measures

Peak Signal to Noise Ratio (PSNR) and structural similarity index (SSIM) are two important measures to calculate the differences between original images and their encrypted form. Generally, PSNR and SSIM show the destruction rate and similarity level between original images and their encrypted form in image encryption methods. In a good image encryption technique, the PSNR and SSIM should be as low as possible. The PSNR is defined as

$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{6}$$

Where MAX_I is the maximum possible pixel value between the original image and the encrypted image. MSE is the mean square error and it is defined as

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - E(i,j)]^2 \tag{7}$$

Table 6
Key sensitivity analysis for decrypted image with small change to initial keys (x_0=0.73, y_0=0.64, z_0=0.34, α=1.32, μ=0.51).

Images	Key sensitivity											
	$x_0 + 10^{-14}$			$y_0 + 10^{-14}$			$z_0 + 10^{-14}$			$\alpha + 10^{-14}$		
	NPCR	UACI	SSIM	NPCR	UACI	SSIM	NPCR	UACI	SSIM	NPCR	UACI	SSIM
Baboon	99.6111	33.4812	0.0055	99.6211	33.4875	0.0052	99.6293	33.4826	0.0048	99.6137	33.4390	0.0069
Couple 1	99.6139	33.4528	0.0062	99.5850	33.5081	0.0041	99.6272	33.5239	0.0039	99.6368	33.5263	0.0035
F-16	99.6227	33.4069	0.0070	99.6175	33.4422	0.0050	99.6189	33.4740	0.0044	99.6072	33.4298	0.0068
Girl 1	99.6307	33.4545	0.0055	99.6206	33.4805	0.0069	99.6226	33.4329	0.0075	99.6170	33.5542	0.0037
Girl 2	99.6002	33.3276	0.0111	99.5997	33.4468	0.0085	99.6053	33.3982	0.0077	99.5926	33.4054	0.0078
Girl 3	99.6033	33.4225	0.0068	99.6104	33.4753	0.0055	99.6007	33.4428	0.0081	99.6038	33.4145	0.0076
House 1	99.5880	33.4409	0.0042	99.6282	33.5256	0.0024	99.5890	33.3380	0.0094	99.5895	33.4593	0.0061
House 2	99.6344	33.4390	0.0066	99.6221	33.4554	0.0064	99.6202	33.4640	0.0063	99.6115	33.4203	0.0084
Jelly Beans 1	99.6348	33.4455	0.0065	99.6175	33.4224	0.0062	99.6175	33.5605	0.0008	99.5844	33.3960	0.0067
Jelly Beans 2	99.5997	33.4392	0.0060	99.6043	33.4168	0.0059	99.6241	33.4371	0.0075	99.6124	33.4259	0.0054
Lena	99.6100	33.4515	0.0066	99.6125	33.5015	0.0040	99.6254	33.4705	0.0053	99.6166	33.4594	0.0054
Peppers	99.6240	33.4487	0.0062	99.6122	33.4917	0.0040	99.6162	33.5292	0.0039	99.6181	33.4674	0.0050
Sailboat	99.6127	33.4601	0.0062	99.6302	33.4814	0.0050	99.6230	33.4498	0.0074	99.6264	33.4983	0.0047
Splash	99.6291	33.4851	0.0038	99.6200	33.4641	0.0055	99.6231	33.4388	0.0064	99.6226	33.4511	0.0059
Tiffany	99.6180	33.4755	0.0063	99.6227	33.4676	0.0048	99.6298	33.4485	0.0053	99.6150	33.4150	0.0077
Tree	99.6134	33.5064	0.0045	99.6190	33.4224	0.0076	99.6043	33.4194	0.0066	99.6256	33.4605	0.0063
alfons rudolph	99.6225	33.4559	0.0062	99.6292	33.4589	0.0062	99.6241	33.4289	0.0066	99.6319	33.4269	0.0071
barn and pond	99.6165	33.4707	0.0044	99.6202	33.4388	0.0065	99.6250	33.4839	0.0055	99.6280	33.4339	0.0069
bob clemens	99.6235	33.4582	0.0056	99.6174	33.4661	0.0053	99.6230	33.4858	0.0060	99.6226	33.4860	0.0048
couple on beach	99.6207	33.4624	0.0057	99.6236	33.4871	0.0046	99.6174	33.5032	0.0034	99.6310	33.5012	0.0045
girl with painted face	99.6174	33.4204	0.0071	99.6220	33.4459	0.0057	99.6219	33.4376	0.0066	99.6305	33.4731	0.0055
hats	99.6221	33.4774	0.0044	99.6238	33.4475	0.0060	99.6162	33.4691	0.0046	99.6248	33.4862	0.0041
lighthouse in Maine	99.6196	33.4340	0.0068	99.6150	33.4312	0.0059	99.6078	33.4131	0.0074	99.6210	33.4495	0.0057
model in black dress	99.6250	33.4459	0.0059	99.6192	33.4366	0.0066	99.6307	33.4600	0.0051	99.6162	33.4343	0.0066
monument	99.6141	33.4506	0.0059	99.6189	33.4442	0.0058	99.6280	33.4646	0.0055	99.6208	33.4531	0.0060
mountain chalet	99.6251	33.4664	0.0053	99.6247	33.4562	0.0056	99.6090	33.4446	0.0068	99.6200	33.4662	0.0056
mountain stream	99.6212	33.4475	0.0052	99.6201	33.4600	0.0053	99.6085	33.4350	0.0068	99.6207	33.4646	0.0058
off-shore sailboat race	99.6169	33.4740	0.0050	99.6164	33.4805	0.0041	99.6231	33.4669	0.0058	99.6161	33.5189	0.0032
p51 Mustang	99.6067	33.4635	0.0056	99.6165	33.4657	0.0050	99.6240	33.4570	0.0055	99.6196	33.4512	0.0068
portland head light	99.6200	33.4026	0.0079	99.6204	33.4508	0.0060	99.6172	33.4615	0.0059	99.6249	33.4579	0.0063
red door	99.6170	33.4479	0.0057	99.6239	33.4546	0.0067	99.6251	33.4843	0.0046	99.6294	33.4779	0.0059
sailboat at anchor	99.6314	33.4465	0.0062	99.6190	33.4784	0.0046	99.6235	33.4707	0.0057	99.6240	33.4322	0.0058
sailboat at pier	99.6134	33.4596	0.0052	99.6269	33.4422	0.0062	99.6236	33.4541	0.0058	99.6178	33.4241	0.0067
sailboats under spinnakers	99.6165	33.4543	0.0056	99.6199	33.4320	0.0061	99.6252	33.4333	0.0066	99.6175	33.4667	0.0057
shuttered windows	99.6254	33.4401	0.0060	99.6228	33.4399	0.0059	99.6228	33.4740	0.0052	99.6252	33.4170	0.0075
steve kelly	99.6314	33.4974	0.0034	99.6223	33.4327	0.0065	99.6249	33.4650	0.0052	99.6230	33.4634	0.0052
stone building	99.6161	33.4266	0.0060	99.6158	33.4734	0.0051	99.6108	33.4544	0.0055	99.6165	33.4726	0.0050
tropical key	99.6146	33.4407	0.0065	99.6064	33.4579	0.0055	99.6260	33.4858	0.0056	99.6104	33.4903	0.0040
two macaws	99.6241	33.4549	0.0055	99.6192	33.4418	0.0069	99.6260	33.4861	0.0046	99.6242	33.4707	0.0055
white water rafters	99.6193	33.4399	0.0061	99.6292	33.4406	0.0059	99.6308	33.4612	0.0060	99.6250	33.4709	0.0054
Mean	99.6182	33.4494	0.0059	99.6184	33.4588	0.0056	99.6198	33.4598	0.0058	99.6184	33.4578	0.0058

Where I and E are the original image and encrypted image, respectively. The m and n are the sizes of the original image. The SSIM is calculated by

$$SSIM(I_1, I_2) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{8}$$

Where σ_x and σ_y are mean intensity of x and y, σ_x^2 and σ_y^2 are the variance of x and y, σ_{xy} is covariance of x and y, respectively. The averages of x and y are μ_y and μ_x, C_1 and C_2 are variables which stabilize the division with weak denominator. Table 5 shows the results of PSNR and SSIM in permutation and encryption processes. This table demonstrates the capability of the proposed method in encrypting images and the direct impact of each process of the proposed scheme.

4.6. Security analysis

4.6.1. Correlation analysis

Statistical research on an image shows that the 8 to 16 adjacent pixels have correlations along the vertical, horizontal and diagonal. An efficient encryption algorithm should be able to greatly reduce this correlation. To evaluate the correlation between two adjacent pixels in a ciphered image, we first randomly select 2,000 pair of pixels. Then the correlation coefficient for the two adjacent pixels (x and y) is calculated by

$$r_{xy} = \frac{N^2 \,.cov(x, y)}{\sum_{i=1}^{N} (x_i - E_x)^2 . \sum_{i=1}^{N} (y_i - E_y)^2} \tag{9}$$

Where x and y are two adjacent pixels sequences of the original image and the encrypted image. E_x, E_y and $cov(x, y)$ are defined as

$$E_x = \frac{\sum_{i=1}^{N} x_i}{N} \;\; and \;\; E_y = \frac{\sum_{i=1}^{N} y_i}{N} \tag{10}$$

$$cov(x, y) = E((x - E_x)(y - E_y)) \tag{11}$$

Fig. 14(e-f) shows the correlation diagram between the original image (peppers) and its encrypted image for three channels R, G, and B in horizontal, vertical, and diagonal directions, respectively. The correlation coefficients of the original and encrypted image pixels in the vertical (V), horizontal (H), and diagonal (D) direction on the dataset images are shown in Table 5.

Table 7
The PSNR results for robust analysis against to image processing attacks.

Attacks	Dataset										Mean
	SIPI					Kodak					
	Baboon	Barbara	Lena	Peppers	Tiffany	Girl	Hats	P51	Red door	Two macaws	
Cropping (5%)	22.0209	21.7162	21.5834	21.1839	19.8792	19.9966	21.6316	19.2671	20.7240	21.2584	20.9261
Cropping (10%)	18.9056	18.6414	18.5372	18.0768	16.8008	17.0128	18.6476	16.2777	17.7272	18.2505	17.8878
Cropping (25%)	14.7664	14.7092	14.5775	14.0297	12.8294	13.0894	14.6864	12.2411	13.7199	14.2065	13.8855
Salt & Pepper (1%)	28.7040	28.7270	28.4956	27.9630	26.9521	26.9961	28.8606	26.2085	27.7259	28.4236	27.9056
Salt & Pepper (5%)	21.7797	21.7656	21.6242	21.0556	20.0028	20.1014	21.7952	19.2617	20.7977	21.3824	20.9566
Salt & Pepper (10%)	18.7216	18.7604	18.5825	18.0163	17.0366	17.0437	18.7697	16.2677	17.7168	18.3513	17.9267
Salt & Pepper (30%)	13.9818	14.0240	13.7892	13.2730	12.2574	12.2747	13.9887	11.5053	12.9573	13.6065	13.1658
Gaussian Noise($\sigma^2 = 0.01$)	14.8491	14.8892	14.6608	14.1389	12.7154	12.9094	14.7487	11.7308	13.6195	14.5449	13.8807
Gaussian Noise($\sigma^2 = 0.001$)	19.2472	19.2992	19.0502	18.5794	16.9816	17.4571	19.1509	15.9494	18.4167	19.0328	18.3165
Jpeg Compression (QF=10)	9.4672	9.5170	9.3480	8.7830	7.8159	7.8007	9.4625	6.9744	8.4125	9.0788	8.6660
Jpeg Compression (QF=50)	10.3144	10.3715	10.2509	9.6931	8.7282	8.7082	10.2816	7.8338	9.2530	9.9190	9.5354
Jpeg Compression (QF=75)	10.6368	10.6933	10.5650	10.0376	9.0422	9.0414	10.6130	8.1231	9.5912	10.2410	9.8584
Jpeg Compression (QF=90)	10.7113	10.7625	10.6391	10.1165	9.0999	9.1127	10.6861	8.1816	9.6732	10.3115	9.9294
Jpeg 2000 (Ratio=10)	10.9861	11.0296	10.7265	10.3288	9.0497	9.2805	10.9693	8.3270	10.0157	10.6252	10.1338
Jpeg 2000 (Ratio=20)	9.9597	10.0031	9.7680	9.3053	8.1664	8.2754	9.9653	7.4062	8.9610	9.6060	9.1416
Median Filter [3 × 3]	9.8659	9.9073	9.7169	9.2236	8.1974	8.2669	9.8717	7.4500	8.9236	9.5398	9.0963
Histogram Equalization	36.7725	44.1595	30.7128	32.1093	25.8990	30.4382	31.8441	33.8927	27.5306	30.1627	32.3521
Gamma Correction ($\gamma = 0.5$)	10.9199	10.8149	10.6780	9.8371	9.0776	8.6084	10.8176	8.0849	9.2054	10.2772	9.8321
Sharpning	13.3483	13.3503	13.2477	13.0321	12.1729	12.4484	13.2483	11.5085	12.9235	13.2470	12.8527
Blurring (len=10,$\theta = 45$)	9.6214	9.6450	9.4418	8.8964	7.8139	7.8595	9.5993	7.0068	8.5430	9.2532	8.7680
Complement	7.1154	7.2489	6.6949	5.4228	3.4639	3.4976	7.2020	2.2082	4.7725	6.1685	5.3795
Resizing (scale=0.5)	9.9845	10.0033	9.8127	9.2606	8.1729	8.2050	9.9590	7.3406	8.8876	9.6271	9.1253
Resizing (scale=2)	13.9310	13.9391	13.7563	13.3346	11.9699	12.2720	13.7890	11.0700	12.9346	13.7394	13.0736

Table 8
The comparison results of the proposed algorithm with similar encryption techniques.

Image	Method	Chaotic map	Entropy	Correlation			Differantial attack	
				Hor.	Ver.	Diag.	NPCR (%)	UACI (%)
Lena	Proposed	Chaos Game	7.9993	-0.0031	-0.0293	0.0077	99.6100	33.4515
	Chai 2019 [48]	DNA & Hyper Chaos	7.9973	-0.0029	0.0013	-0.0026	99.6000	33.5600
	Asgari 2019 [52]	Combined chaotic systems	7.9984	-0.0022	0.0013	0.0029	99.7024	33.5249
	Alawida 2019 [40]	Tent-Logistic-Tent system	7.9975	-0.0017	-0.0084	-0.0019	99.6200	33.8120
	Chen 2018 [49]	DNA & Hyper Lorenz	7.9993	0.0003	-0.0064	0.0110	99.6067	33.4951
	Wang 2018 [45]	Sine-Sine system	-	0.0003	-0.0054	0.0016	99.6413	33.4801
	Wu 2018 [43]	2D Hénon-Sine map	7.9994	0.0032	0.0016	0.0023	99.6002	33.5079
	Pak 2017 [51]	Mixed map	-	-0.0038	-0.0026	0.0017	99.6552	33.4846
	Wu 2017 [41]	4D Arnold cat map	7.9912	-0.0001	0.0089	0.0091	99.9999	33.7400
	Ratnavelu 2017 [54]	FCNN	7.9978	0.0089	-0.0346	0.0259	99.9100	33.3500
	Belazi 2016 [38]	Improved Logistic map	7.9974	-0.0362	-0.0141	-0.0464	99.6143	33.6532
	Murillo 2015 [37]	Logistic map	7.9975	0.0135	-0.0835	-0.0170	99.6100	33.3600
	Lio 2015 [44]	Henon map	7.9896	0.0009	0.0009	-0.0015	99.6140	33.4747
	Hsiao 2015 [53]	Chaotic APFM	7.9896	0.0031	0.0010	0.0014	99.6078	33.4211
	Wang 2012 [36]	Logistic map	-	-0.0038	-0.0509	0.0122	99.6475	33.4274
Baboon	Proposed	Chaos Game	7.9993	0.0224	0.0115	-0.0025	99.6111	33.4812
	Hsiao 2015 [53]	Chaotic APFM	7.9974	0.0016	0.0029	0.0024	99.6029	33.4931
	Ratnavelu 2017 [54]	FCNN	7.9982	-0.0182	0.0105	0.0003	99.9100	33.4400
	Cao 2018 [50]	2D-LICM hyperchaos	7.9972	0.0054	0.0003	0.0043	99.6059	33.4161
	Wu 2018 [43]	2D Hénon-Sine map	7.9992	0.0029	0.0033	0.0062	99.5903	33.5281
	Alawida 2019 [40]	Tent-Logistic-Tent system	7.9971	-0.0015	-0.0026	0.0014	99.6010	33.7120
Peppers	Proposed	Chaos Game	7.9993	0.0282	0.0122	-0.0056	99.6240	33.4487
	Alawida 2019 [40]	Tent-Logistic-Tent system	7.9970	0.0024	-0.0131	0.0002	99.6170	33.3910
	Cao 2018 [50]	2D-LICM hyperchaos	7.9993	0.0003	0.0014	0.0007	99.6582	33.4550
	Wu 2018 [43]	2D Hénon-Sine map	7.9993	0.0006	0.0038	0.0010	99.6112	33.5265
	Belazi 2016 [38]	Improved Logistic map	7.9974	-0.0413	-0.0043	-0.0002	99.6190	33.6126
House	Proposed	Chaos Game	7.9969	0.0025	-0.0069	-0.0037	99.5880	33.4409
	Chai 2019 [48]	DNA & Hyper Chaos	7.9974	0.0010	-0.0019	0.0011	99.6100	33.4900
	Ratnavelu 2017 [54]	FCNN	7.9983	0.0078	-0.0088	0.0221	99.9300	33.3300
Tiffany	Proposed	Chaos Game	7.9993	0.0109	-0.0023	0.0012	99.6180	33.4755
	Ratnavelu 2017 [54]	FCNN	7.9977	0.0125	0.0197	0.0198	99.9900	33.3900

Table 9
The key space comparison of the proposed algorithm with similar chaotic methods.

Method	Chaotic map	Key space
Ref. [54]	Fuzzy cellular neural network	2^{900}
Ref. [53]	Chaotic APFM	2^{651}
Ref. [38]	Improved logistic map	2^{624}
Ref. [48]	DNA & Hyper Chaos	2^{614}
Ref. [43]	2D Hénon-Sine map	2^{372}
Ref. [40]	Tent-Logistic-Tent system	2^{312}
Ref. [41]	4D Arnold cat map	2^{280}
Ref. [52]	Combined chaotic systems	2^{256}
Ref. [50]	2D-LICM hyperchaotic map	2^{208}
Ref. [49]	DNA & Hyper Lorenz	2^{199}
Ref. [55]	Integrated Chaotic Systems	2^{186}
Ref. [36]	Logistic map	2^{186}
Ref. [44]	Henon Map	2^{140}
Ref. [51]	Mixed map	2^{138}
Ref. [37]	Logistic map	2^{128}
Ref. [45]	Sine-Sine system	2^{128}
proposed	Chaos Game	2^{232}

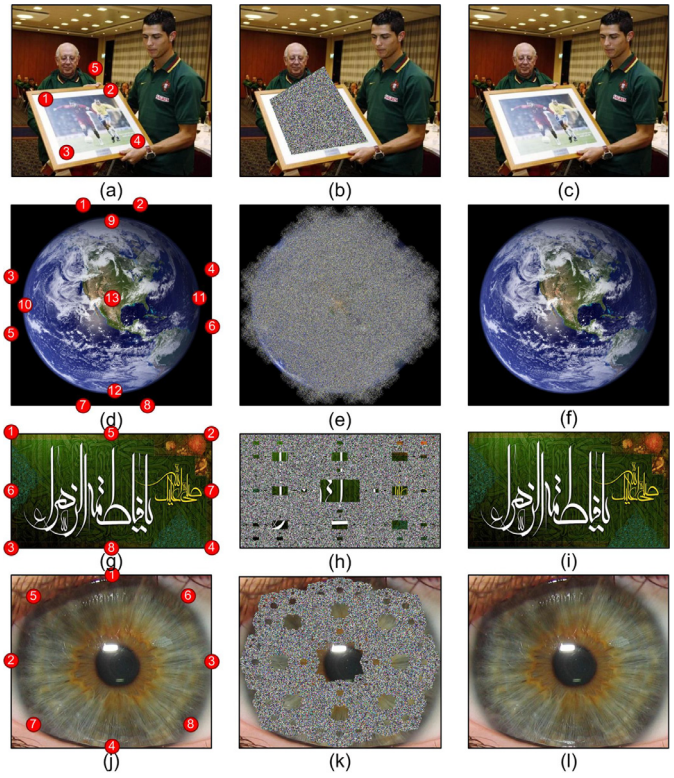

Fig. 11. The results of encrypted & decrypted color images based on the selected region of interest, (a-c) pyramid selection, (d-f)Spherical Selection, (g-i) Sirpienski carpet, (j-l) Iris Encryption.

4.7. Information entropy

One of the ways to detect randomness property in a signal is to use entropy analysis. The entropy was introduced in 1949 by Shannon for data communication and storage [77]. The concept of entropy in physics is related to the degree of irregularity and uncertainty. The entropy of a digital image is the randomized estimate that is used to measure the sharpness of the histogram peaks. The entropy rate of a digital image is defined by

$$H(m) = \sum_{i=0}^{2^n-1} P(m_i) Log_2 \frac{1}{P(m_i)} \tag{12}$$

Where the probability of value m_i is shown by $P(m_i)$ and $N-1$ is the number of gray value. If the entropy is 8, a completely random image is created which is considered as an ideal value. If entropy is closer to 8, it can be less predictable and more secure. The obtained values from the computed entropy for the original and encrypted image on the data-set images are shown in Table 5.

4.8. Key sensitivity

In order to evaluate the key sensitivity, an arbitrary key is considered and the encryption process is performed with that key. Encryption is repeated again with the same keys but with a very small change in one of the keys. As shown in Fig. 14(b) and Fig. 14(c), this sensitivity is visible to the key with a very small change in one of the keys.

This paper uses the number of pixels change rate (NPCR) and unified average changing intensity (UACI) to evaluate the sensitivity of the proposed method to the small change on the input keys. NPCR and UACI refer to diffusion and they are defined as

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\% \tag{13}$$

$$UACI = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|C(i,j) - C'(i,j)|}{255 \times M \times N} \times 100\% \tag{14}$$

where M and N are size of plain image. $C(i, j)$ and $C'(i, j)$ are plain image and changed image, respectively. $D(i, j)$ is the differences array and it is defined as $D(i,j) = \begin{cases} 0, & if\ C(i,j) = C'(i,j); \\ 1, & otherwise; \end{cases}$

The key sensitivity analysis is shown based on the very small changes (10^{-14}) to each of the key parameters in Table 6.

4.9. Robustness against to image processing attacks.

In practical applications, the encrypted image must be resistant to image-processing attacks. These attacks could include issues such as noise in computer networks and the transmission of images on these networks. Conventional noise in computer networks can be noted to gaussian noise and salt and pepper noise. Attacks can also be caused by tampering by the attacker manually or automatically so that the encrypted image does not reach the destination correctly. These attacks include image cropping, lossy compression (JPEG, JPEG2000), median filtering, histogram equalization, gamma correction, image sharpening, image complement, and image scaling. To test the resistance of encrypted images against attacks, visual measures such as PSNR are used. Table 7 shows the results of the obtained PSNR after the attacks with different input parameters on the 10 image samples of the dataset images. The impact of the cropping attack on the performance of the decryption process with a variety of this particular type of attack is illustrated in Fig. 17. Also, the visual quality of the Baboon encrypted image after the image processing attacks is shown in Fig. 18. As shown in Fig. 18, the encrypted image after the attacks is in many cases relatively good and recognizable.

4.10. Comparisons with similar methods

To analyze the efficiency of the proposed algorithm, this algorithm is investigated with other similar algorithms. The results

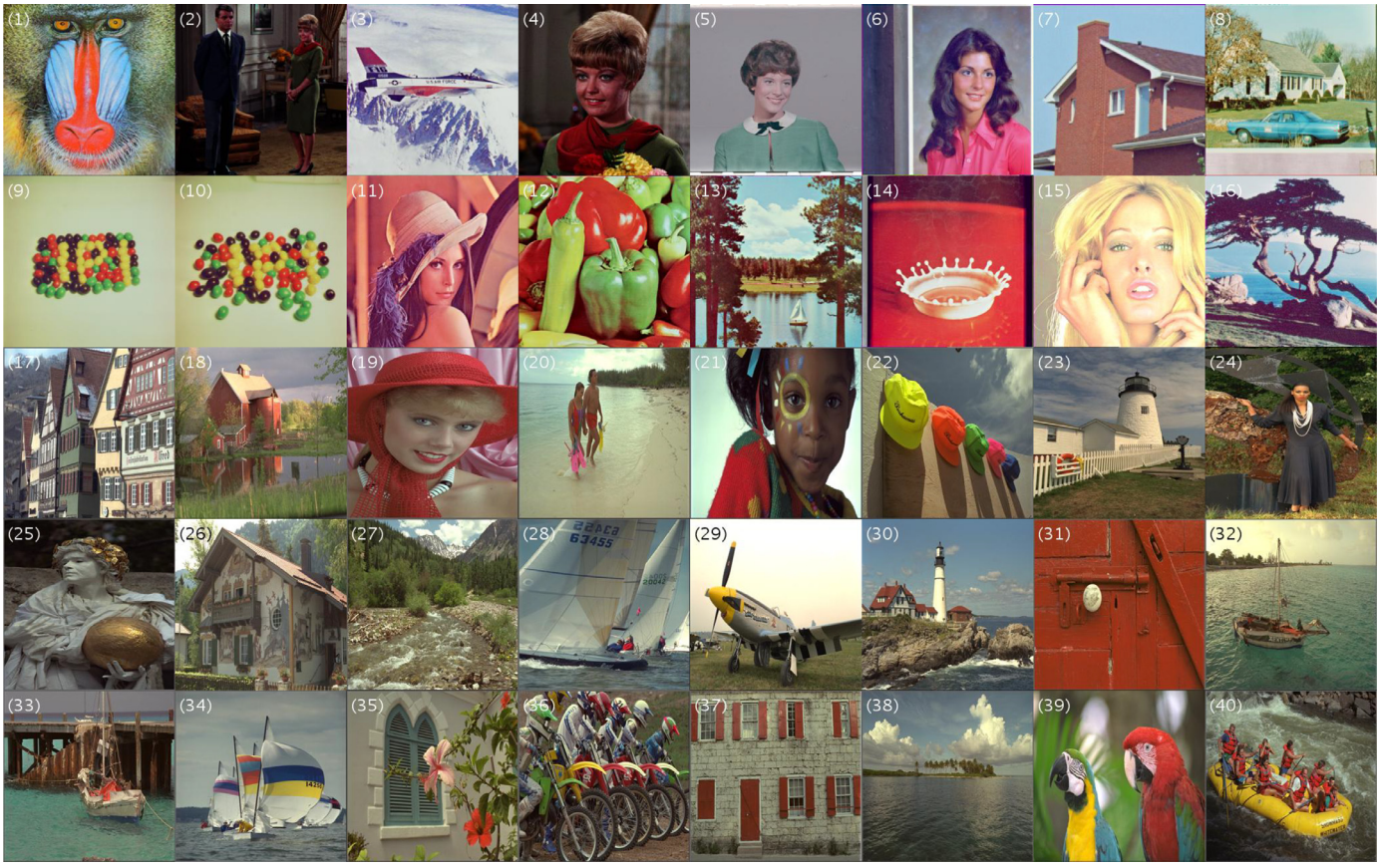

Fig. 12. The SIPI [75] & kODAK Image Database [76] : (1) Baboon, (2) Couple 1, (3) F-16, (4) Girl 1, (5) Girl 2, (6) Girl 3, (7) House 1, (8) House 2, (9) Jelly Beans 1, (10) Jelly Beans 2, (11) Lena, (12) Peppers, (13) Sailboat, (14) Splash, (15) Tiffany, (16) Tree, (17) alfons rudolph, (18) barn and pond, (19) bob clemens, (20) couple on beach, (21) girl with painted face, (22) hats, (23) lighthouse in Maine, (24) model in black dress, (25) monument, (26) mountain chalet, (27) mountain stream, (28) off-shore sailboat race, (29) p51 Mustang, (30) portland head light, (31) red door, (32) sailboat at anchor, (33) sailboat at pier, (34) sailboats under spinnakers, (35) shuttered windows, (36) steve kelly, (37) stone building, (38) tropical key, (39) two macaws, (40) white water rafters.

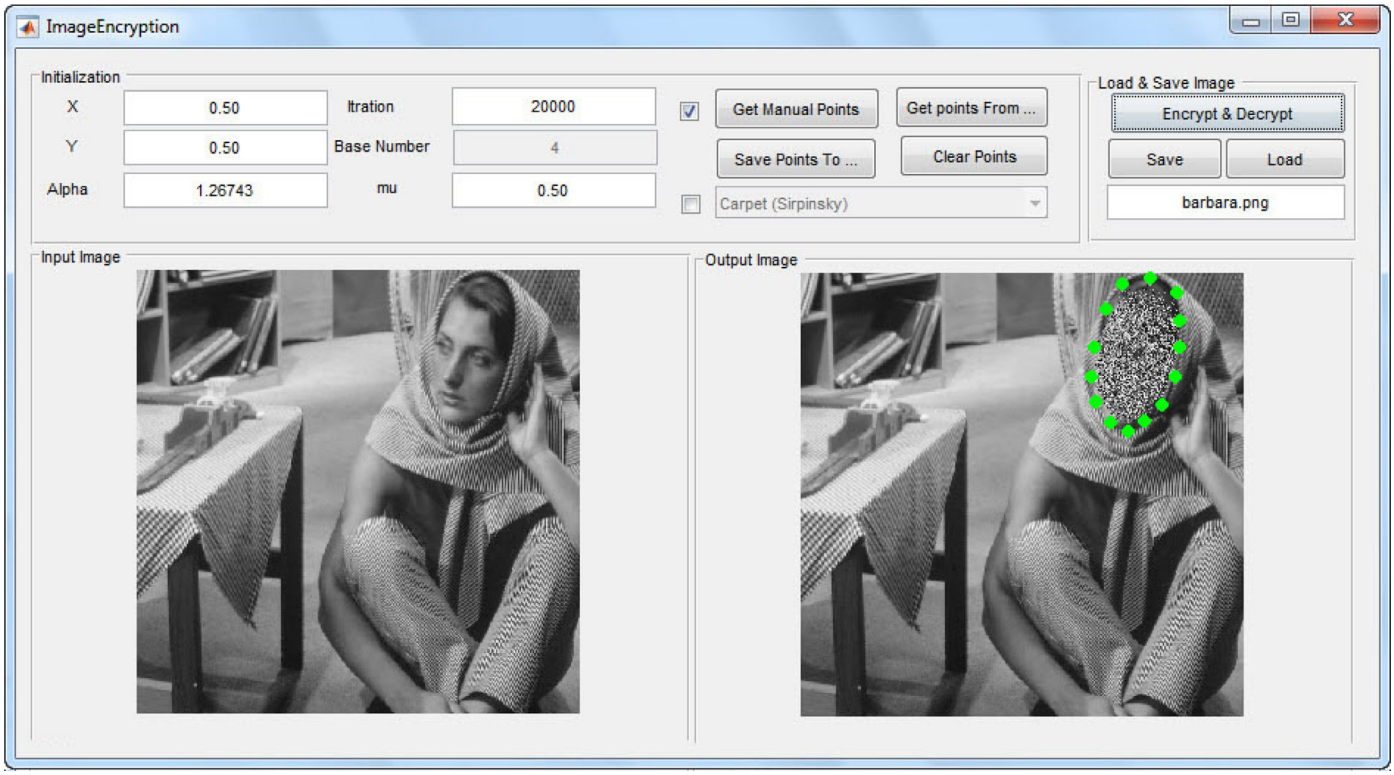

Fig. 13. The designed Matlab GUI for proposed image encryption.

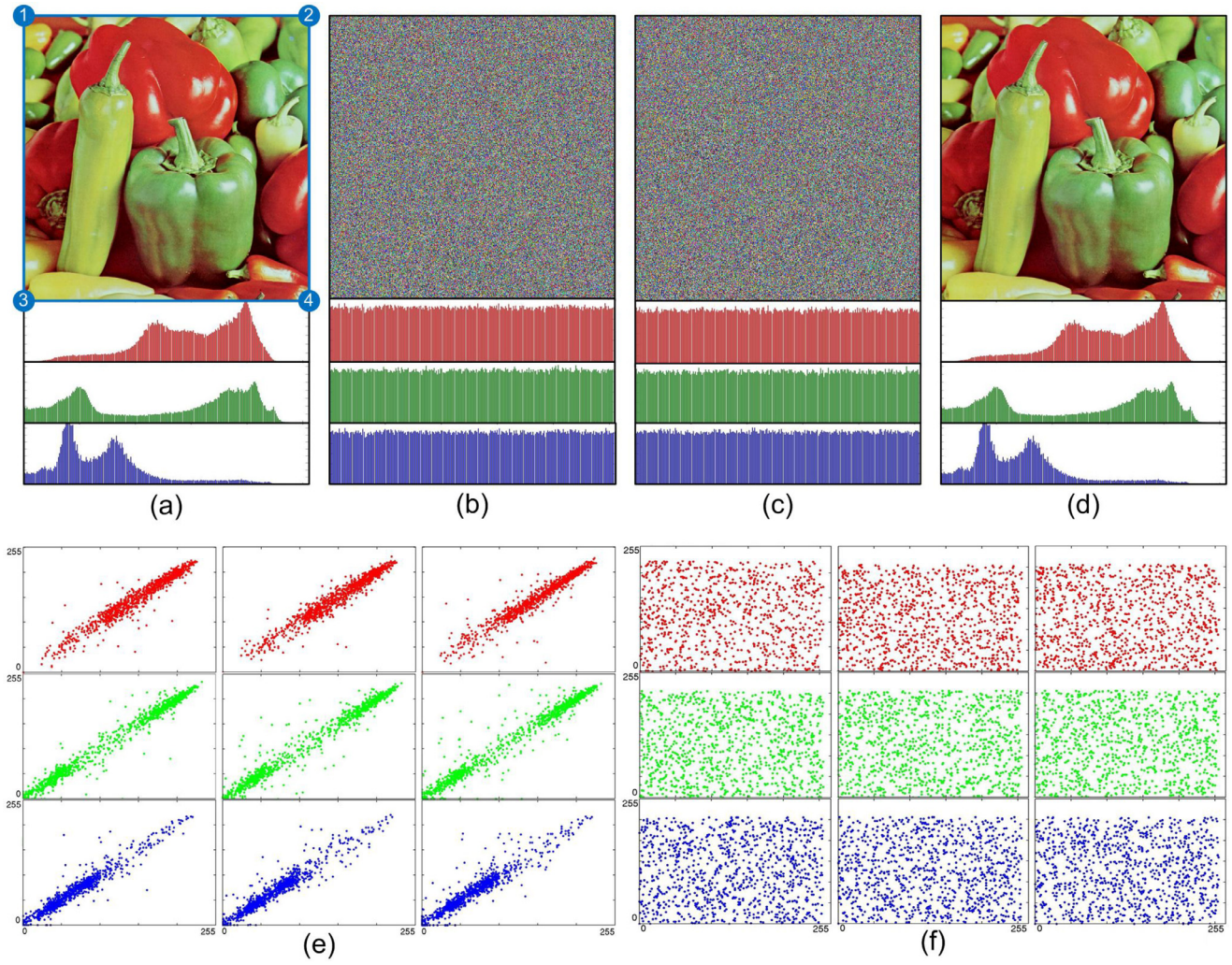

Fig. 14. Histogram analysis based on RGB channels : (a) Original Image (b) Encrypted Image (c) decrypted image by incorrect keys (d) decrypted image by correct keys, and Correlation analysis of two adjacent pixels in horizontal (left), Vertical(Mid) and diagonal(Right) for (e) Original Image (f) Encrypted Image.

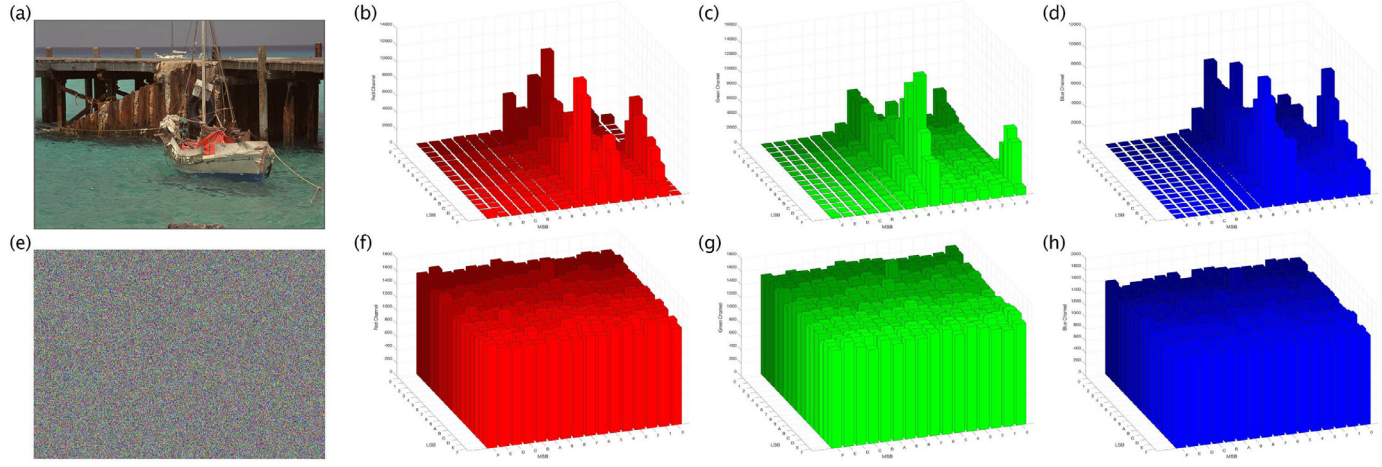

Fig. 15. The 3D histogram analysis based on the discrimination of MSB and LSB : (a) original image (b-d) 3D histogram of the original image for RGB channels, (e) encrypted image (f-h) 3D histogram of the encrypted image for RGB channels, respectively.

of this study are shown in Table 8 based on entropy analysis, correlation analysis, and differential attacks. Based on the results of this table, the proposed algorithm has no significant difference with other methods and can be accepted as an efficient algorithm.

Also, from the perspective of security analysis, the key space of the proposed chaos game is compared with other chaos-based encryption methods. As previously mentioned, the key length larger than 128 bits is secure. Therefore, the proposed algorithm is in good agreement with the security analysis.

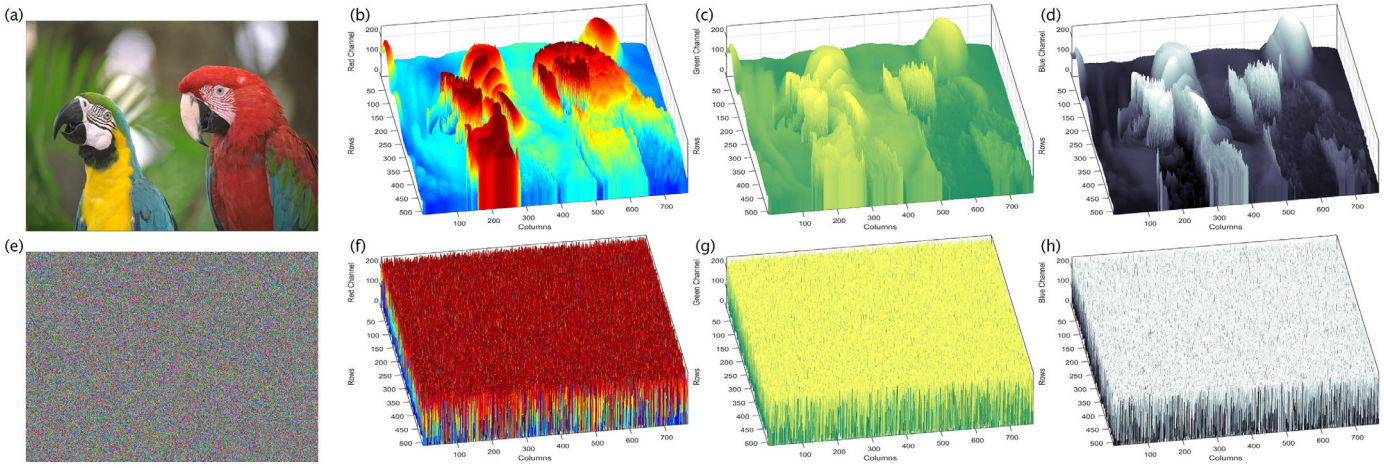

Fig. 16. Spatial characteristics analysis : (a) original image (b-d) spatial diagram of the original image for RGB channels, (e) encrypted image (f-h) spatial diagram of the encrypted image for RGB channels, respectively.

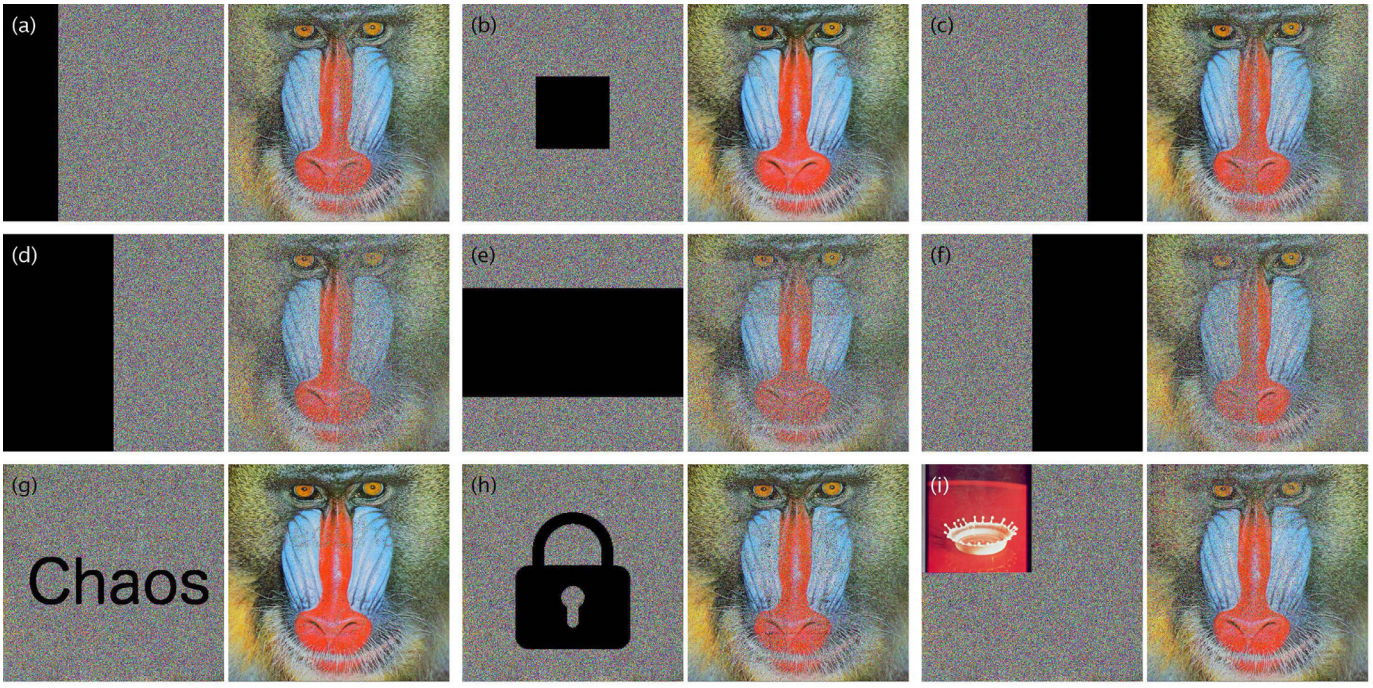

Fig. 17. The robustness analysis against cropping attacks: (a-c) 25% cropping (d-f) 50% cropping (g) tampering by 'CHAOS' text, (h) tampering by a lock shape (f) tampering by another image.

5. Conclusion

In this paper, a generalization of a dynamical system called the chaos game is presented. The chaos game is a chaotic fractal sequence that can be used as an encryptor in the ROI or non-ROI from the digital image. Chaos game was used as a pseudo-random number generator in the encryption and decryption process, with a key length of 232 bits. To demonstrate the chaotic behavior of the proposed dynamic system, the Lyapunov exponent and the bifurcation diagram were used and also exploited from statistical test suites to test random behavior of proposed PRNG such as NIST, ENT, Diehard, and TESTU01. The results of these tests showed that the proposed chaos game is fully chaotic and has a randomness feature. To evaluate the efficiency of the proposed algorithm, histogram analysis, entropy analysis, correlation analysis, and key sensitivity analysis were used. The performance analysis criteria were calculated on standard datasets from the digital images and the results were compared with other similar methods. The

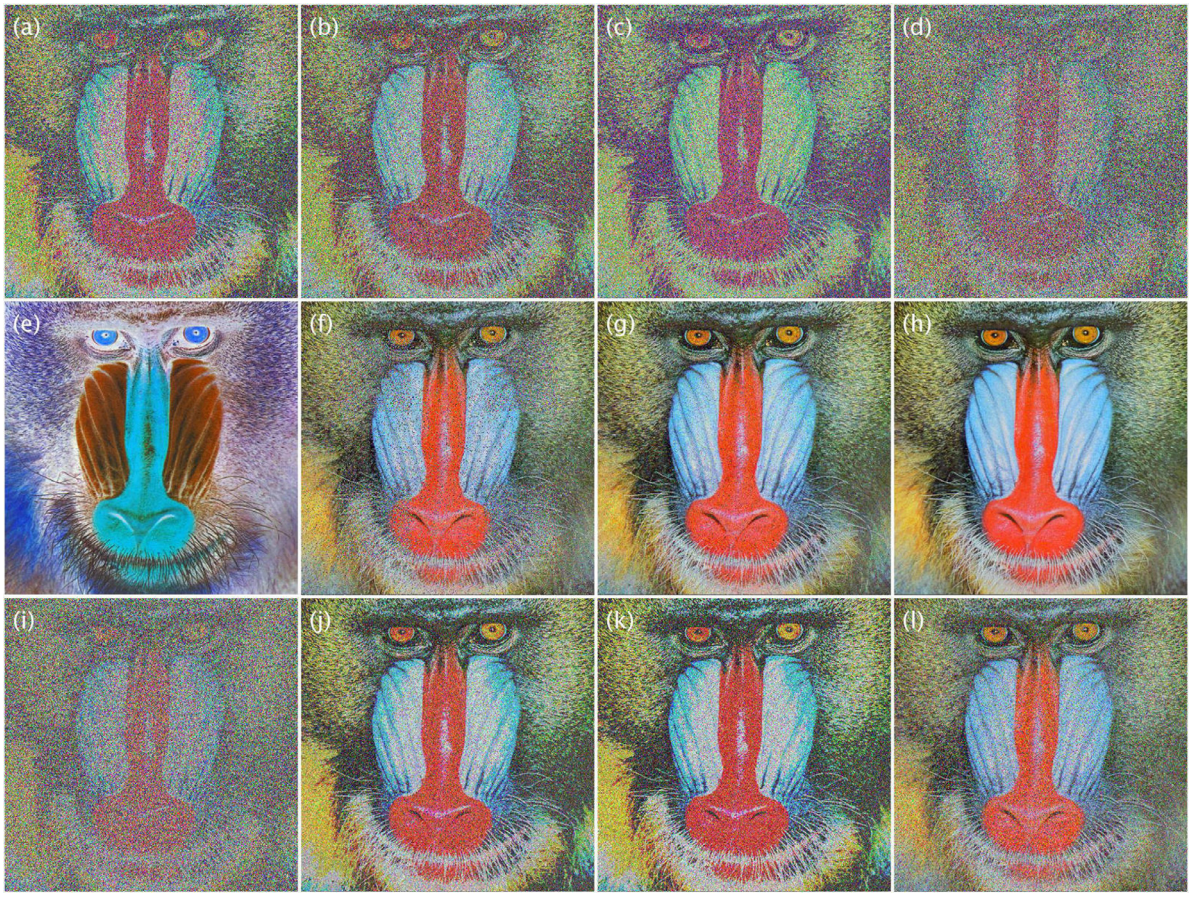

Fig. 18. The results of Common Image processing Attacks : (a) Gamma Correction ($\gamma = 0.5$), (b) Jpeg 2000 (Ratio=10), (c) Jpeg Compression (QF=75), (d) Blurring (len=10,$\theta = 45$), (e) Complement, (f) Cropping (25%), (g) Gaussian Noise($\sigma^2 = 0.001$), (h) Histogram Equalization, (i) Median Filter [3 × 3], (j) Resizing (scale=2), (k) Sharpning, (l) Salt & Pepper (30%).

comparison results showed that the proposed algorithm does not have a significant difference with other efficient methods and can be used as a safe algorithm.

Compliance with ethical standards

All authors declare that they have no conflict of interest.

Declaration of Competing Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

CRediT authorship contribution statement

Peyman Ayubi: Conceptualization, Methodology, Software, Investigation, Writing - original draft, Visualization. **Saeed Setayeshi:** Writing - original draft, Writing - review & editing, Project administration. **Amir Masoud Rahmani:** Writing - original draft, Writing - review & editing.

Acknowlgedgments

This article is dedicated to Imam Hussein, who has all my scientific life from his love.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jisa.2020.102472.

References

[1] Faragallah OS, El-Samie FEA, Ahmed HEH, Elashry IF, Shahieen MH, El-Rabaie E-SM, et al. Image encryption: a communication perspective. CRC Press; 2013.
[2] Kaur M, Kumar V. A comprehensive review on image encryption techniques. Arch Comput Methods Eng 2018:1–29.
[3] Song Y, Zhu Z, Zhang W, Guo L, Yang X, Yu H. Joint image compression–encryption scheme using entropy coding and compressive sensing. Nonlinear Dyn 2018:1–27.
[4] Tong X-J, Wang Z, Zhang M, Liu Y. A new algorithm of the combination of image compression and encryption technology based on cross chaotic map. Nonlinear Dyn 2013;72(1–2):229–41.
[5] Paar C, Pelzl J. Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media; 2009.
[6] Tuchman W. A brief history of the data encryption standard. In: Internet besieged. ACM Press/Addison-Wesley Publishing Co.; 1997. p. 275–80.
[7] Daemen J, Rijmen V. Aes proposal: Rijndael; 1999.
[8] Kumari M, Gupta S, Sardana P. A survey of image encryption algorithms. 3D Research 2017;8(4):37.
[9] Kneusel RT. Random Numbers and Computers, 239. Springer; 2018.
[10] Rukhin A, Soto J, Nechvatal J, Smid M, Barker E. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. Rep.. Booz-allen and hamilton inc mclean va; 2001.
[11] Marsaglia G. Diehard: a battery of tests of randomness. Open Source software library 1996.
[12] Brown RG, Eddelbuettel D, Bauer D. Dieharder: a random number test suite. Open Source software library 2013.
[13] Walker J. Ent: a pseudorandom number sequence test program; 2008.
[14] L'Ecuyer P, Simard R. Testu01: Ac library for empirical testing of random number generators. ACM Trans Math Software (TOMS) 2007;33(4):22.

[15] Kocarev L, Lian S. Chaos-based cryptography: theory, algorithms and applications, 354. Springer Science & Business Media; 2011.
[16] Özkaynak F. Brief review on application of nonlinear dynamics in image encryption. Nonlinear Dyn 2018;92(2):305–13.
[17] Ahmad M, Doja M, Beg MS. A new chaotic map based secure and efficient pseudo-random bit sequence generation. In: International symposium on security in computing and communication. Springer; 2018. p. 543–53.
[18] Lambić D, Janković A, Ahmad M. Security analysis of the efficient chaos pseudo-random number generator applied to video encryption. J Electr Testing 2018;34(6):709–15.
[19] Valandar MY, Barani MJ, Ayubi P. A fast color image encryption technique based on three dimensional chaotic map. Optik 2019;193:162921.
[20] Irani BY, Ayubi P, Jabalkandi FA, Valandar MY, Barani MJ. Digital image scrambling based on a new one-dimensional coupled sine map. Nonlinear Dyn 2019:1–29.
[21] Behnia S, Ayubi P, Soltanpoor W. Image encryption based on quantum chaotic map and fsm transforms. In: 2012 15th International telecommunications network strategy and planning symposium (NETWORKS). IEEE; 2012. p. 1–6.
[22] Ahmad M, Ahmad T. Securing multimedia colour imagery using multiple high dimensional chaos-based hybrid keys. Int J Commun Networks Distrib Syst 2014;12(1):113–28.
[23] Verma OP, Nizam M, Ahmad M. Modified multi-chaotic systems that are based on pixel shuffle for image encryption. J Inf Process Syst 2013;9(2):271–86.
[24] Valandar MY, Ayubi P, Barani MJ. A new transform domain steganography based on modified logistic chaotic map for color images. J Inf Secur Appl 2017;34:142–51.
[25] Valandar MY, Barani MJ, Ayubi P, Aghazadeh M. An integer wavelet transform image steganography method based on 3d sine chaotic map. Multim Tools Appl 2019;78(8):9971–89.
[26] Valandar MY, Ayubi P, Barani MJ. High secure digital image steganography based on 3d chaotic map. In: 2015 7th Conference on information and knowledge technology (IKT). IEEE; 2015. p. 1–6.
[27] Behnia S, Ahadpour S, Ayubi P. Design and implementation of coupled chaotic maps in watermarking. Appl Soft Comput 2014;21:481–90.
[28] Farri E, Ayubi P. A blind and robust video watermarking based on iwt and new 3d generalized chaotic sine map. Nonlinear Dyn 2018;93(4):1875–97.
[29] Hadi RM, Ayubi P. Blind digital image watermarking based on ct-svd and chaotic cellular automata. In: 2012 2nd International eConference on computer and knowledge engineering (ICCKE). IEEE; 2012. p. 301–6.
[30] Khorrami N, Ayubi P, Behnia S, Ayubi J. A svd-chaos digital image watermarking scheme based on multiple chaotic system. In: International joint conference on advances in signal processing and information technology. Springer; 2012. p. 9–18.
[31] Panahi N, Amirani M, Behnia S, Ayubi P. A new colour image watermarking scheme using cellular automata transform and schur decomposition. In: 2013 21st Iranian conference on electrical engineering (ICEE). IEEE; 2013. p. 1–5.
[32] Behnia S, Teshnehlab M, Ayubi P. Multiple-watermarking scheme based on improved chaotic maps. Commun Nonlinear Sci NumerSimul 2010;15(9):2469–78.
[33] Barani MJ, Valandar MY, Ayubi P. A secure watermark embedding approach based on chaotic map for image tamper detection. In: 2015 7th Conference on information and knowledge technology (IKT). IEEE; 2015. p. 1–5.
[34] Barani MJ, Ayubi P, Jalili F, Valandar MY, Azariyun E. Image forgery detection in contourlet transform domain based on new chaotic cellular automata. Secur Commun Networks 2015;8(18):4343–61.
[35] Barani MJ, Valandar MY, Ayubi P. A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3d quantum map. Optik 2019;187:205–22.
[36] Wang X, Teng L, Qin X. A novel colour image encryption algorithm based on chaos. Signal Process 2012;92(4):1101–8.
[37] Murillo-Escobar M, Cruz-Hernández C, Abundiz-Pérez F, López-Gutiérrez RM, Del Campo OA. A rgb image encryption algorithm based on total plain image characteristics and chaos. Signal Process 2015;109:119–31.
[38] Belazi A, El-Latif AAA, Belghith S. A novel image encryption scheme based on substitution-permutation network and chaos. Signal Process 2016;128:155–70.
[39] Ahmad M, Alam MZ, Umayya Z, Khan S, Ahmad F. An image encryption approach using particle swarm optimization and chaotic map. Int J Inf Technol 2018;10(3):247–55.
[40] Alawida M, Samsudin A, Teh JS, Alkhawaldeh RS. A new hybrid digital chaotic system with applications in image encryption. Signal Process 2019;160:45–58.
[41] Wu J, Liao X, Yang B. Color image encryption based on chaotic systems and elliptic curve elgamal scheme. Signal Process 2017;141:109–24.
[42] Chopra A, Ahmad M, Malik M. An enhanced modulo-based image encryption using chaotic and fractal keys. In: 2015 International conference on advances in computer engineering and applications. IEEE; 2015. p. 501–6.
[43] Wu J, Liao X, Yang B. Image encryption using 2d hénon-sine map and dna approach. Signal Process 2018;153:11–23.
[44] Liu H, Kadir A. Asymmetric color image encryption scheme using 2d discrete–time map. Signal Process 2015;113:104–12.
[45] Wang H, Xiao D, Chen X, Huang H. Cryptanalysis and enhancements of image encryption using combination of the 1d chaotic map. Signal Process 2018;144:444–52.
[46] Li C, Lin D, Lü J, Hao F. Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography. IEEE MultiMedia 2018;25(4):46–56.
[47] Li C, Feng B, Li S, Kurths J, Chen G. Dynamic analysis of digital chaotic maps via state-mapping networks. IEEE Trans Circuits Syst I 2019;66(6):2322–35.
[48] Chai X, Fu X, Gan Z, Lu Y, Chen Y. A color image cryptosystem based on dynamic dna encryption and chaos. Signal Process 2019;155:44–62.
[49] Chen J, Zhu Z-l, Zhang L-b, Zhang Y, Yang B-q. Exploiting self-adaptive permutation–diffusion and dna random encoding for secure and efficient image encryption. Signal Process 2018;142:340–53.
[50] Cao C, Sun K, Liu W. A novel bit-level image encryption algorithm based on 2d-licm hyperchaotic map. Signal Process 2018;143:122–33.
[51] Pak C, Huang L. A new color image encryption using combination of the 1d chaotic map. Signal Process 2017;138:129–37.
[52] Asgari-Chenaghlu M, Balafar M-A, Feizi-Derakhshi M-R. A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation. Signal Process 2019;157:1–13.
[53] Hsiao H-I, Lee J. Color image encryption using chaotic nonlinear adaptive filter. Signal Process 2015;117:281–309.
[54] Ratnavelu K, Kalpana M, Balasubramaniam P, Wong K, Raveendran P. Image encryption method based on chaotic fuzzy cellular neural networks. Signal Process 2017;140:87–96.
[55] Lan R, He J, Wang S, Gu T, Luo X. Integrated chaotic systems for image encryption. Signal Process 2018;147:133–45.
[56] Singla P, Sachdeva P, Ahmad M. A chaotic neural network based cryptographic pseudo-random sequence design. In: 2014 Fourth international conference on advanced computing & communication technologies. IEEE; 2014. p. 301–6.
[57] Hua Z, Xu B, Jin F, Huang H. Image encryption using josephus problem and filtering diffusion. IEEE Access 2019;7:8660–74.
[58] Akhshani A, Akhavan A, Lim S-C, Hassan Z. An image encryption scheme based on quantum logistic map. Commun Nonlinear Sci NumerSimul 2012;17(12):4653–61.
[59] Alzaidi AA, Ahmad M, Ahmed HS, Solami EA. Sine-cosine optimization-based bijective substitution-boxes construction using enhanced dynamics of chaotic map. Complexity 2018;2018.
[60] Alzaidi AA, Ahmad M, Doja MN, Al Solami E, Beg MS. A new 1d chaotic map and β-hill climbing for generating substitution-boxes. IEEE Access 2018;6:55405–18.
[61] Ahmad M, Doja M, Beg MS. Cryptanalysis and improvement of an image encryption scheme using fourier series. 3D Res 2017;8(4):40.
[62] Li C, Zhang Y, Xie EY. When an attacker meets a cipher-image in 2018: a year in review. J Inf Secur Appl 2019;48:102361.
[63] Ahmad M, Doja M, Beg MS. Security analysis and enhancements of an image cryptosystem based on hyperchaotic system. J King Saud Univ-ComputInf Sci 2018.
[64] Ahmad M, Alam MZ, Ansari S, Lambić D, AlSharari HD. Cryptanalysis of an image encryption algorithm based on pwlcm and inertial delayed neural network. J Intell Fuzzy Syst 2018;34(3):1323–32.
[65] Ahmad M, Al Solami E, Wang X-Y, Doja M, Beg M, Alzaidi A. Cryptanalysis of an image encryption algorithm based on combined chaos for a ban system, and improved scheme using sha-512 and hyperchaos. Symmetry 2018;10(7):266.
[66] Feldman DP. Chaos and fractals: an elementary introduction. Oxford University Press; 2012.
[67] Falconer K. Fractal geometry: mathematical foundations and applications. John Wiley & Sons; 2004.
[68] Arjunan SP, Kumar D, Aliahmad B. Fractals: applications in biological Signalling and image processing. CRC Press; 2017.
[69] Mandelbrot BB. The fractal geometry of nature, 173. WH Freeman New York; 1983.
[70] Bouallegue K. Gallery of chaotic attractors generated by fractal network. Int J Bifurc Chaos 2015;25(1):1530002.
[71] Barnsley MF. Fractals everywhere. Academic Press, San Diego; 1988.
[72] Strogatz SH. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Westview Press; 2014.
[73] Akhshani A, Akhavan A, Mobaraki A, Lim S-C, Hassan Z. Pseudo random number generator based on quantum chaotic map. Commun Nonlinear Sci NumerSimul 2014;19(1):101–11.
[74] Smart N. Ecrypt ii yearly report on algorithms and keysizes (2010-2011). Katholieke Universiteit Leuven (KUL) Deliverable SPA-17 rob June 2011.
[75] Weber A. The usc-sipi image database, signal and image processing institute of the university of southern california. Volume 3: Miscellaneous 1997.
[76] Ponomarenko N, Jin L, Ieremeiev O, Lukin V, Egiazarian K, Astola J, et al. Image database tid2013: Peculiarities, results and perspectives. Signal Process 2015;30:57–77.
[77] Shannon CE. Communication theory of secrecy systems. Bell SystTechnJ 1949;28(4):656–715.