The Journal of Systems and Software 97 (2014) 128–139

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

RGB color image encryption based on Choquet fuzzy integral

Seyed Mohammad Seyedzadeh [a,*], Benyamin Norouzi [b], Sattar Mirzakuchaki [b]

[a] Department of Computer Science, University of Pittsburgh, PA 15260, USA
[b] Department of Electrical Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran

ARTICLE INFO

Article history:
Received 15 August 2013
Received in revised form 12 May 2014
Accepted 13 July 2014
Available online 19 July 2014

Keywords:
Image encryption
Choquet fuzzy integral
Mathematical analysis

ABSTRACT

In recent years, one can see an increasing interest in the security of digital images. This research presents a new RGB color image encryption using keystream generator based on Choquet fuzzy integral (CFI). The properties of the dynamical keystream generator with mathematical analysis are presented in this work. In the proposed method, the CFI is first used to generate pseudo-random keystreams. Then, each of the color pixels is decomposed into three gray-level components. The output of the CFI is used to randomly shift the bits of three gray-level components. Finally, three components of RGB color pixels and the generated keystream are coupled to encrypt the permuted components. Performance aspects of the proposed algorithm such as the entropy analysis, differential analysis, statistical analysis, cipher random analysis, and cipher sensitivity analysis are introduced to evaluate the security of the new scheme. The experimental results reveal the fact that the proposed algorithm is suitable for practical use in protecting the security of digital image information distributed via the Internet.

1. Introduction

1.1. Background

With the rapid development of information technology and network communication, the transmission of a wide range of digital data, from digital images to audio and video files, through the Internet or wireless networks has been increased. In this virtual environment, problems associated with image security are becoming progressively important. In recent years, many image encryption methods have been presented (Chen and Chen, 2011; Pareek et al., 2013; Behnia et al., 2013; Mazloom and Eftekhari-Moghadam, 2009; Norouzi et al., 2013a). Generally, the image encryption architectures consist of two processes: pixel permutation and diffusion (El-Latif et al., 2013; Bakhshandeh and Eslami, 2013). The permutation process changes the position of image pixels. This process greatly reduces the high correlation among pixels but it does not alter the frequency distribution of RGB color pixel values. The diffusion process modifies the pixel values so that a tiny change in one pixel can distribute to almost all pixels in the whole image. However, in most of the proposed encryption architectures, there remains some weaknesses to be resolved before a cryptosystem can gain possible pervasive applications.

1.2. Related work

Part of the literature, which is relevant to the present work, is provided herein. One of the recent methods that has focused on the permutation step is suggested in Zhu et al. (2011), where the pixel-level permutation is replaced by a bit-level permutation. Such a method changes pixels' positions and gray-levels in one permutation step. In this method, authors utilized the Arnold cat map for permutation and the logistic map for diffusion. The Arnold cat map is the cornerstone model of the classical dynamical chaos (Lichtenberg and Lieberman, 1992). This symplectic map which belongs to the class of Anosov systems has the positive Kolmogorov–Sinai entropy of $h \approx 0.96$ and is also fully chaotic. In order to disturb the high correlation among pixels, this map can shuffle the pixel positions of the plain-image. Yuen and Wong (2011) presented a chaos-based fast image encryption by combining permutation and diffusion into one step. The plain-image is divided into blocks, and the one-dimensional chaotic map is used to shuffle the blocks while the diffusion step is simultaneously applied.

To gain high complexity in the encryption methods, multiple chaos-based algorithms have been utilized. Gao and Chen (2008a) applied a substitution matrix to the whole plain-image in the confusion phase, and then did the diffusion operation by compound

* Corresponding author. Tel.: +1 4123877057.
E-mail addresses: ses174@pitt.edu (S.M. Seyedzadeh), benyamin_norouzi@elec.iust.ac.ir (B. Norouzi), m_kuchaki@iust.ac.ir (S. Mirzakuchaki).

http://dx.doi.org/10.1016/j.jss.2014.07.025
0164-1212/

chaos systems. A new bilateral diffusion algorithm based on chaos and Linear Feedback Shift Register (LFSR) was put forward by Tong (2012). These authors employed LFSR to disturb the compound chaotic system and produced more key space. Gao and Chen (2008b) proposed a hyper chaotic cryptosystem based on Chen system to encrypt the gray-level image. Rhouma and Belghith (2008) presented the cryptanalysis of this image encryption algorithm based on hyper-chaos and two different attacks. Liu et al. (2013b) proposed an image encryption algorithm based on hyper-chaotic Lorenz system and the CFI. The major core of their encryption algorithm is a pseudo-random number generator based on the CFI. These authors used the output of the CFI to confuse and diffuse the three RGB color components, respectively. They did not however present any mathematical analysis to prove the randomness of the pseudo-random number generator.

Other researches presented in Wei et al. (2012), Liu and Wang (2013), Seyedzadeh and Mirzakuchaki (2012), Norouzi et al. (2012) and Seyedzadeh et al. (2011) revealed good experimental results, but current chaotic image encryption algorithms have the following flaws more or less: (1) some algorithms utilized Arnold cat map to confuse pixels, but Arnold cat map (Sui and Gao, 2013; Abuturab, 2013) has two fundamental weaknesses (Liu and Wang, 2011). One is that the iteration times are very limited and is not usually more than 1000 times. The other is that the width and height of the original image must be identical; otherwise, the image pre-processing must be done. (2) Block design rules are fixed. (3) In order to get perfect encryption effect, the algorithm must ensure that encryption procedures are related to a plain-image. But some researchers have not realized that encryption procedures only had low relevance with plain-image. (4) Some algorithms encrypted RGB color components independently and neglected the correlations between R, G and B components and were more vulnerable to attacks. (5) Most chaotic maps are unstable due to the periodicity of the mapping (Lou and Sung, 2004; Huang and Feng, 2009). Systems based on such maps are prone to attacks, such as the broken system shown in Çokal and Solak (2009). (6) Some algorithms can only encrypt square images, and if the image's height and width are not equal, the image cannot be permuted directly.

1.3. Contribution and organization of the paper

The main contribution of this paper is to design a pseudo-random number generator based on Choquet fuzzy integral which has greater sensitivity, higher degree of complexity and larger key space than current state-of-the-art generators. This paper utilizes the proposed generator to present a novel image encryption cryptosystem which consists of three processes: the pseudo-random number generation process based on the CFI, the random circular shift process and the diffusion process. First process acts as a pseudo-random number generator based on the CFI which uses a 128-bit external security key. In the middle process, the bits of each pixel are permuted by the random circular shift. This process effectively reduces the correlation of adjunct pixels and increases the resistance of the algorithm against statistical attacks. In the last process, the encryption transform uses the pseudo-random sequence based on the CFI to encrypt the permuted pixels. The properties of the dynamical keystream generator with mathematical analysis are proved rigorously. The mathematical analysis truly assures that the CFI has high complexity. Besides, the randomness of the sequences generated by this type of the map is very high which extremely increases the security and the sensitivity of the proposed algorithm.

The rest of this paper is organized in the following manner. Section 2 describes the theory of the fuzzy measure and the fuzzy integral. In Section 3, the proposed cryptosystem is explained. Simulation results and security analysis are provided in Section 4. Finally, the conclusions are drawn in Section 5.

2. Fuzzy measure and fuzzy integral

The Choquet fuzzy integral with respect to the fuzzy measure is often used as a nonlinear aggregation tool. The non-additivity of fuzzy measures can effectively describe the interaction among the contributions from each attribute toward the same target. The contribution of this paper is to approach this target toward a pseudo-random output. In this section, we briefly review the major properties of the fuzzy integral and one subset special type called Choquet fuzzy integral.

2.1. Fuzzy measure

Let $X=\{x_1, \ldots, x_n\}$ be a finite set associated with n attributes on information source space, and let $p(X)$ denote the power set consisting of all subsets of X. Sugeno presented the so called γ-fuzzy measure (Sugeno, 1977) satisfying the following property for all $A, B \subset X$ with $A \cap B = \Phi$:

$$g(A \cup B) = g(A) + g(B) + \gamma g(A) g(B), \quad \gamma > -1 \tag{1}$$

In general, the value of γ can be determined owing to the g_γ fuzzy measure and is found by solving the following equation:

$$g_\gamma(X) = \frac{1}{\gamma}\left(\prod_{i=1}^{n}(1+\gamma g_i) - 1\right), \quad \gamma \neq -1 \tag{2}$$

and substituting

$$\gamma + 1 = \prod_{i=1}^{n}(1+\gamma g_i) \tag{3}$$

where $\gamma \in (-1, \infty)$, $\gamma \neq 0$, and $g_i = g(\{x_i\})$ is the value of the fuzzy density function. Eq. (3) can be calculated by solving the $(n-1)$th degree polynomial and finding the unique root >-1.

Let $A_i = \{x_1, \ldots, x_i\}$ be a subset of elements of the universe of discourse. The values of $g(A_i)$ by the fuzzy measure over the corresponding subsets of elements can recursively be determined as follows:

$$g(A_1) = g(\{x_1\}) = g_1 \tag{4}$$

$$g(A_i) = g_i + g(A_{i-1}) + \gamma g_i g(A_{i-1}), \quad 1 < i \leq n \tag{5}$$

In order to obtain γ-fuzzy measure, there are n parameters $g_1, \ldots, g_n$ needed to be determined in advance. There are a number of interesting families of fuzzy integrals in terms of the underlying fuzzy measures (Friedman et al., 1996; Grabisch and Nguyen, 1994; Sugeno and Murofushi, 1987; Wang et al., 1996). One of the particular interests which we consider in this work is Choquet fuzzy integral (Seyedzadeh and Hashemi, 2011; Seyedzadeh and Mirzakuchaki, 2011; Ralescu and Ralescu, 1997; Murofushi and Sugeno, 1989).

2.2. Choquet fuzzy integral

The fuzzy integral (Murofushi and Sugeno, 1989; Chiang, 2000) $\int h \circ g$ of a measurable function $h: A \to (0, +\infty)$ with respect to a fuzzy measure g is defined in the form:

$$\int_A h \circ g \tag{6}$$

For a finite set of $A=\{x_i, \cdots, x_n\}$, Choquet fuzzy integral can be computed as follows:

$$E_g(h) = \sum_{i=1}^{n}[h(x_i) - h(x_{i-1})]g(A_i) \tag{7}$$

where $h(x_1) \le h(x_2) \le \cdots \le h(x_n)$, and $h(x_0) = 0$. Another computation formula for the finite set case can also be represented by:

$$E_g(h) = \sum_{i=1}^{n} h(x_i)[g(A_i) - g(A_{i+1})] \tag{8}$$

where we take $g(A_{i+1})$ to be 0. From Eqs. (5) and (7), it is obvious that the calculation of Choquet fuzzy integral with respect to γ-fuzzy measure requires the knowledge of the fuzzy density g and the input value h.

For n different inputs, Choquet fuzzy integral needs to solve the $(n-1)$th degree polynomial of Eq. (3) and find the unique root >-1. To find the roots >-1, conditions must be considered to restrict n membership grades (see Appendix A). This proposed algorithm solves the 2nd degree polynomial for three different inputs and finds the unique roots >-1 using conditions mentioned in Appendix A. In contrast to Choquet fuzzy integral with three inputs, Choquet fuzzy integral with more than three inputs needs to solve polynomials 3rd degree and higher. Finding conditions that restrict membership grades in polynomials with 3rd degree and higher is very complex. In this paper, a 128-bit key is utilized to generate the initial parameters of the CFI and to increase the security of the proposed algorithm. The nonlinear structure of the CFI will greatly increase the sensitivity of the proposed generator to initial parameters.

3. The scheme of image encryption and decryption

The proposed encryption cryptosystem is based on the permutation–diffusion architecture. There are three cascade single stages in this kind of the image cryptosystem. The first stage utilizes a secret key to generate initial conditions and parameters of the CFI and then the CFI produces pseudo-random keystreams simultaneously. In the middle stage, the bits of gray-level components are randomly shifted by the pseudo-random keystreams. In the last stage, the output of the cipher-image is related not only to the keystreams but also to the components of the RGB color image.

3.1. Pseudo-random keystream generation process

Based on the analysis presented in Section 2, there are three initial inputs (h_1, h_2, h_3) and three membership grades (g_1, g_2, g_3) for generating pseudo-random keystreams by the CFI. To increase the security of the proposed algorithm, a 128 bit-long secret key is used to generate the initial inputs and parameters of the CFI by making some algebraic transformations to the key. This key is divided into 8-bit blocks (k_i) referred to as session keys. The 128-bit external secret-key (K) is given by:

$$K = k_1, k_2, \ldots, k_{16} \tag{9}$$

The initial condition of the CFI is then derived as follows:

$$t_i = \left(\left(\frac{k_{4i-3} + (4i-3)}{k_{4i-2} + (4i-2)} + \frac{k_{4i-1} + (4i-1)}{k_{4i} + (4i)} \right) \times \sum_{j=1}^{j=16} \frac{k_j \times 2^{8\times(j-1)}}{2^{128}} \right) \bmod 1 \tag{10}$$

where $i = 1, 2, 3, 4$ and values t_i are rearranged in increasing order and are set equal to h_1, h_2, h_3 and β, respectively. The dynamic parameter β maps g_3 to a value >2. The range of three membership grades has been analyzed in Appendix A. According to Appendix A, we generate three membership grades as follows:

$$\begin{aligned} g_1 &= 0.5 \\ g_2 &= 2 + \frac{g_3 + 1}{g_3 - 1} \\ g_3 &= 2 + \beta \end{aligned} \tag{11}$$

Obviously, Eqs. (10) and (11) show that initial parameters of the CFI are greatly sensitive to the change in even a single bit of the 128-bit secret key. As a result, the proposed cryptosystem with the total complexity of 2^{128} can resist against any brute-force attack. In order to generate the pseudo-random keystreams, the output of CFI or E is computed by Eqs. (5) and (7), and then pseudo-random keystreams are generated as follows:

$$y = \left(ARS(Int((E \bmod 1) \times 10^{14}), S) + \sum_{i=1}^{i=16} k_i \right) \bmod 256 \tag{12}$$

where $0 \le (E \bmod 1) < 1$ and the values of S determine the number of pseudo-random keystreams generated in each iteration of the CFI.

Remark: ARS (M, N) performs the N-bit arithmetic right shift operation on the binary sequence M. Also, $Int(\alpha)$ returns the integer value of the argument α.

3.2. The rule of fuzzy measures in the formation of the proposed generator

The pseudo-random number generator-based Choquet fuzzy integral depends on inputs h_i and fuzzy densities $g(A_i)$. Fuzzy densities are generated by fuzzy measures g_i and the parameter γ. The values of γ must be real and >-1. For this reason, we select $\Delta > 0$ in Eq. (A.3) (see Appendix A) and then find the range of fuzzy measures g_i to satisfy Eq. (A.3). If values of fuzzy measures g_i do not satisfy Δ, values of γ will be complex conjugates. If these complex numbers are utilized in Eq. (5), the complex exponential function appears in the output of Eq. (7). Since each complex exponential function consists of the trigonometric functions, i.e. cosine and sine, Eq. (7) in the proposed generator always produces periodic outputs. This analysis shows that the determination of the precise range of fuzzy measures plays a significant role in the formation of the pseudo-random generator with non-periodic outputs or at least outputs with very long period.

3.3. Chaotic behavior and complex degree of the proposed pseudo-random generator

In order to determine whether Eq. (12) is chaotic or not, the simplest way is to calculate this equation's Lyapunov exponent. The positive value of Lyapunov exponent expresses the system sensitivity to initial conditions, as well as the randomness of the signals. The basic expression of the discrete Lyapunov exponent is described as (Wolf et al., 1985; Aurell et al., 1997):

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| \acute{f}(y_i) \right| \tag{13}$$

where $\acute{f}(y_i)$ is the differential function of the pseudo-random map. As the function of Eq. (12) is not differentiable, the value λ is given by:

$$\lambda(h) = \frac{1}{L} \sum_{i=1}^{L} \ln \left| \frac{y_h[i] - y_{h'}[i]}{h[i] - h'[i]} \right| \tag{14}$$

where y_h and $y_{h'}$ correspond to keystream vectors generated by sets of initial conditions h and h' with length L, respectively.

With initial values of fuzzy inputs h_i where $1 \le i \le L$ and $L = 102,400$, the Lyapunov exponents $\lambda(h_i)$ are computed for two near initial values h_i and h_i'. The sensitivity to the initial conditions is depicted in Fig. 1a. Fig. 1b reveals an example of the sensitivity on the initial conditions for two sets of $y_h[i]$ and $y_{h'}[i]$ as a function of the position i. It clearly appears that two sets of pseudo-random keystreams are very sensitive to the initial conditions (i.e. initial values h and h'). For these two sets of keystreams, we present the

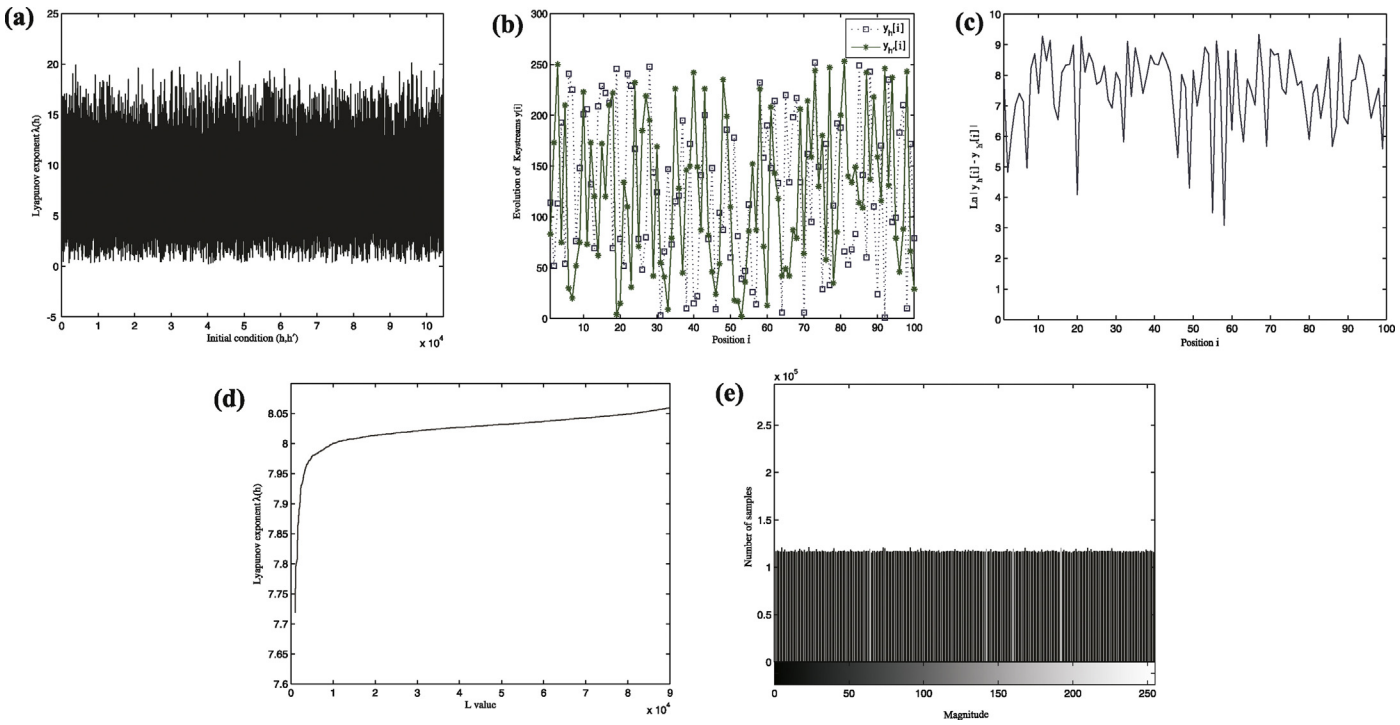

Fig. 1. Chaotic behavior of the pseudo-random function of Eq. (12): (a) Lyapunov exponent between generated keystreams for sets of initial conditions h and h',(b) sensitivity on the initial conditions for 100 iterations, (c) log difference between two sequences generated by the keystreams y_h and $y_{h'}$ for 100 iterations, (d) Lyapunov exponent as function of different lengths of the keystream, and (e) histogram of the proposed pseudo-random function.

$Ln|y_h[i] - y_{h'}[i]|$ as a function of the position i (see Fig. 1c). Fig. 1d shows Lyapunov exponent values as a function of different lengths of the keystream. The results based on real simulations showed that all corresponding Lyapunov exponents are positive and approximately belong to the interval (0.1, 20]. The range of the interval for different pseudo-random generators is not the same but all of them have a range >0. The positive values of Lyapunov exponent justify the sensitivity of the system to initial conditions and also the chaotic behavior of the function in Eq. (12). Finally, we have constructed a histogram for thirty million pseudo-random samples to display the frequency in which pseudo-random keystreams fall into a given region in the state space. The histogram of Eq. (12) is uniformly distributed as shown in Fig. 1e.

In order to evaluate the complexity and the unpredictability of the proposed generator, we conduct a randomness complexity experimental which includes Lyapunov exponents and approximate entropy which reveals capacity of data prediction by positive numbers. Larger approximate entropy means higher sequence complexity and randomness with a longer period. The computing formula of approximate entropy is as follows (Tong, 2012; Liu et al., 2013a):

$$d[u(i), u(j)] = Max\{|u(i+k-1) - u(j+k-1)|\}k = 1, 2, \ldots, m \tag{15}$$

$$\phi_m(r) = \frac{\sum_{i=1}^{N-m+1} \ln C_i^m(r)}{N-m+1} \tag{16}$$

$$ApEn(m, r) = \phi^m(r) - \phi^{m+1}(r) \tag{17}$$

where $C_i^m(r)$ is the ratio between the number of $d[u(i), u(j)]$ and the value of $N-m+1$, $d[u(i), u(j)]$ is the distance between vector $u(i)$ and $u(j)$, N is the length of the sequence, and m is the dimension of the vector. Fig. 1 and Table 1 show that the new proposed generator has greater sensitivity to initial values and higher complicated degree than the dynamical compound map (Tong, 2012)

Table 1
Comparison of randomness complexity.

Item	Map name		
	CFI	LFSR (Tong, 2012)	Logistic $4x(1-x)$
Lyapunov exponents	1.841376	1.356059	0.696033
Approximate entropy	1.934287	1.358034	0.771543

and the famous Logistic chaotic map. With respect to key space, the proposed generator needs four initial values, so its key space is larger than that of the chaotic systems based on the dynamical compound map and the logistic map. The key space of the proposed generator, dynamical compound map and the logistic map is 2^{128}, 4×10^{28} and 2×10^{28}, respectively. Therefore, It can be seen that our generator has greater sensitivity to initial values, higher complicated degree and larger key space than the current state-of-the-art generators.

3.4. Encryption algorithm

For a RGB color image of size $W \times H$, we treat its components as one-dimensional vectors $R = r_1, r_2, \ldots, r_{W\times H}$, $G = g_1, g_2, \ldots, g_{W\times H}$ and $B = b_1, b_2, \ldots, b_{W\times H}$. The detailed encryption steps for the proposed algorithm are as follows:

Step 1. Apply the external 128-bit secret key and set $i = 1$ and $L = W \times H$. According to Section 3.1, generate initial inputs of h_1, h_2, h_3 and membership grades g_1, g_2, g_3.

Keystream generation process: This process produces pseudo-random keystreams for encryption of the RGB color image based on the CFI. We set $S = 0, 3, 5$ in Eq. (12) to generate three pseudo-random numbers in each iteration of the CFI. To this end, $W \times H$ pseudo-random numbers are produced for RGB color components of size $W \times H$.

Step 2. Use the new initial conditions to iterate Eq. (12) and store three pseudo-random numbers in a one-dimensional vector of x_i, x_{i+1}, x_{i+2}, respectively. Update the initial inputs of the CFI as follows:

$$\begin{aligned} \beta &= E \bmod 1 \\ g_3 &= 2 + \beta \\ g_2 &= 2 + \frac{g_3 + 1}{g_3 - 1} \\ h_{j=1,2,3}^{update} &= \left(\frac{h_j}{256} + \beta \right) \mod 1 \end{aligned} \tag{18}$$

where E and β are the output of the CFI and the dynamic parameter generated in Eq. (10), respectively. Set $i = i + 3$, and then iterate this step until $i \le L$.

Random circular permutation and diffusion processes: In the random circular permutation process, the bits of each pixel are randomly shifted by the keystream generated by the pseudo-random number generator. In order to make the influence of changing a single pixel in the plain-image on the cipher-image, the encryption transformations are designed so that encrypting each pixel of the plain-image depends on previous encrypted values and subsequent plain values. Therefore, this dependence yields the fact that a swift change in the plain-image results in a significant change in the cipher-image. To begin these processes, we set $i = 1$.

Step 3. Randomly Shift the bits of the three components as follows:

$$\begin{aligned} r_i^{new} = &\left(\left(r_i + \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \ll ((x_i \bmod 7) + 1) \right) \\ &\oplus \left(\left(r_i + \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \gg (8 - ((x_i \bmod 7) + 1)) \right) \end{aligned} \tag{19}$$

$$\begin{aligned} g_i^{new} = &\left(\left(g_i + \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \ll ((r_i^{new} \bmod 7) + 1) \right) \\ &\oplus \left(\left(g_i + \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \gg (8 - ((r_i^{new} \bmod 7) + 1)) \right) \end{aligned} \tag{20}$$

$$\begin{aligned} b_i^{new} = &\left(\left(b_i + \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \ll ((g_i^{new} \bmod 7) + 1) \right) \\ &\oplus \left(\left(b_i + \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \gg (8 - ((g_i^{new} \bmod 7) + 1)) \right) \end{aligned} \tag{21}$$

where notations "⊕", "≪" and "≫" are the bit-wise exclusive OR operator, Left-shift operator and Right-shift operator, respectively.

Step 4. Compute the corresponding pixel data of the cipher-image by values of the current shifted pixel, the previous encrypted pixels, and the subsequent plain pixels as follows:

$$\begin{aligned} cr_i &= \left(\left(r_i^{new} + \sum_{j=1}^{i-1} cr_j + \sum_{j=i+1}^{L} r_j^{new} \right) \bmod 256 \right) \oplus x_i \\ cg_i &= \left(\left(g_i^{new} + \sum_{j=1}^{i-1} cg_j + \sum_{j=i+1}^{L} g_j^{new} \right) \bmod 256 \right) \oplus cr_i \\ cb_i &= \left(\left(b_i^{new} + \sum_{j=1}^{i-1} cb_j + \sum_{j=i+1}^{L} b_j^{new} \right) \bmod 256 \right) \oplus cg_i \end{aligned} \tag{22}$$

Set $i = i + 1$ and go to step 3 and then iterate steps 3 and 4 until $i \le L$.

3.5. Decryption algorithm

The decryption procedure is similar to that of the encryption process except that some steps are followed in a reversed order. Therefore, some remarks should be given in the decryption process as follows:

Remark: We can rewrite Eqs. (19)–(22) to give the pixels' values as follows:

$$\begin{aligned} b_i &= \left(\left(cb_i - \sum_{j=1}^{i-1} cb_j - \sum_{j=i+1}^{L} b_j \right) \bmod 256 \right) \oplus cg_i \\ g_i &= \left(\left(cg_i - \sum_{j=1}^{i-1} cg_j - \sum_{j=i+1}^{L} g_j \right) \bmod 256 \right) \oplus cr_i \\ r_i &= \left(\left(cr_i - \sum_{j=1}^{i-1} cr_j - \sum_{j=i+1}^{L} r_j \right) \bmod 256 \right) \oplus x_i \end{aligned} \tag{23}$$

$$\begin{aligned} b_i^{original} = &\left(\left(b_i - \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \gg ((g_i \bmod 7) + 1) \right) \\ &\oplus \left(\left(b_i - \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \ll (8 - ((g_i \bmod 7) + 1)) \right) \end{aligned} \tag{24}$$

$$\begin{aligned} g_i^{original} = &\left(\left(g_i - \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \gg ((r_i \bmod 7) + 1) \right) \\ &\oplus \left(\left(g_i - \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \ll (8 - ((r_i \bmod 7) + 1)) \right) \end{aligned} \tag{25}$$

$$\begin{aligned} r_i^{original} = &\left(\left(r_i - \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \gg ((x_i \bmod 7) + 1) \right) \\ &\oplus \left(\left(r_i - \sum_{j=1, j\neq i}^{j=L} x_i \right) \bmod 256 \ll (8 - ((x_i \bmod 7) + 1)) \right) \end{aligned} \tag{26}$$

Since the decryption process requires the same keystream to decrypt each RGB color component, so the same secret key $K = k_1, k_2, \ldots, k_{16}$ should be used for decryption to set the same initial conditions h_1, h_2, h_3 and the memberships g_1, g_2, g_3.

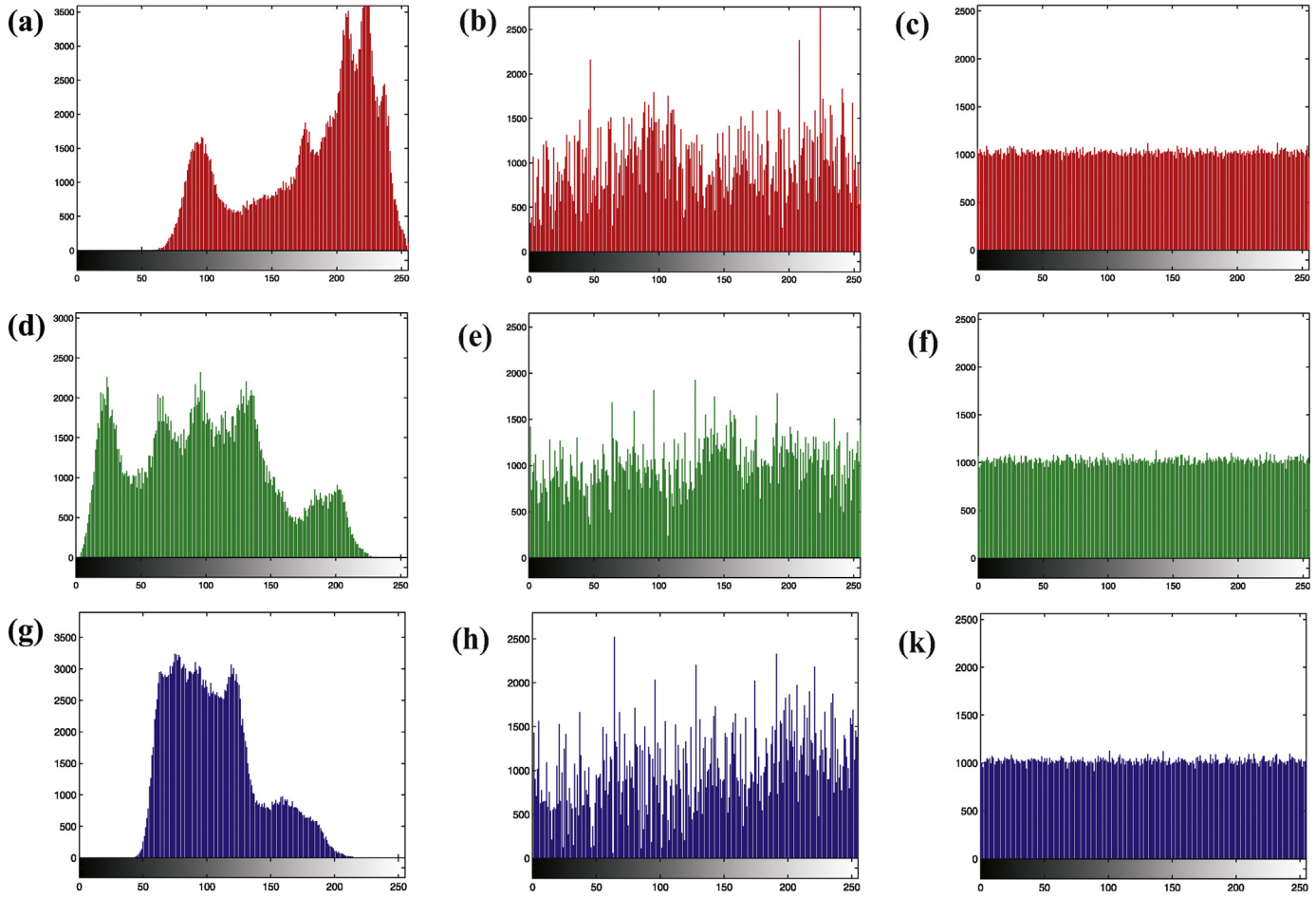

Fig. 2. (a) Histogram of the plain-image Lena – R, (b) histogram of the image Lena – R after random circular shift process, (c) histogram of the image Lena – R after the diffusion process, (d) histogram of the plain-image Lena – G, (e) histogram of the image Lena – G after random circular shift process, (f) histogram of the image Lena – G after the diffusion process, (g) histogram of the plain-image Lena – B, (h) histogram of the image Lena – B after random circular shift process, and (k) histogram of the image Lena – B after the diffusion process.

4. Security analysis for the proposed algorithm

We have carried out several measurements on the CVG-UGR image database and PSU near-regular texture database to check the security and performance of the proposed cryptosystem. These measurements consist of statistical analysis, sensibility analysis, avalanche criterion, and randomness tests for the cipher-images. Each of these measurements is described in detail in the following subsections.

4.1. Statistical analysis

4.1.1. Histogram of encrypted image

Image histogram is a very important feature in image analysis and shows the frequency distribution of gray-level values. Fig. 2 shows the frequency distribution of red, green, and blue components before and after the random circular shift process. It is obvious that histograms of encrypted components are nearly uniform and significantly different from histograms of original components. As a result, the proposed algorithm can resist any statistical analysis attack on the cipher-image.

4.1.2. Correlation of two adjacent pixels

We have analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent pixels in an image. 4000 pairs of two adjacent (in vertical, horizontal, and diagonal direction) pixels from plain-image and cipher-image were randomly selected and the correlation coefficients were calculated using the following equations:

$$
\begin{aligned}
r_{xy} &= \frac{|Cov(x,y)|}{\sqrt{D(x)}\sqrt{D(y)}} \\
E &= \frac{1}{N}\sum x_i \\
D(x) &= \frac{1}{N}\sum (x_i - E(x))^2 \\
Cov(x,y) &= \frac{1}{N}\sum (x_i - E(x))(y_i - E(y))
\end{aligned}
\tag{27}
$$

The x and y show gray-level values of two adjacent pixels. Fig. 3 plots the distribution of these sample pixels and their neighborhood pixels in the horizontal direction. Table 2 shows results of correlation analysis. Fig. 3 and Table 2 show the significant reduction in relevance of adjacent pixels in red, green and blue color components.

Since a high correlation exists among corresponding pixels of R, G and B components in color images, the proposed algorithm encrypts pixels of color components in such a way that these pixels are greatly independent. Tables 3 and 4 show the results of the same position correlations and related adjacent position correlations

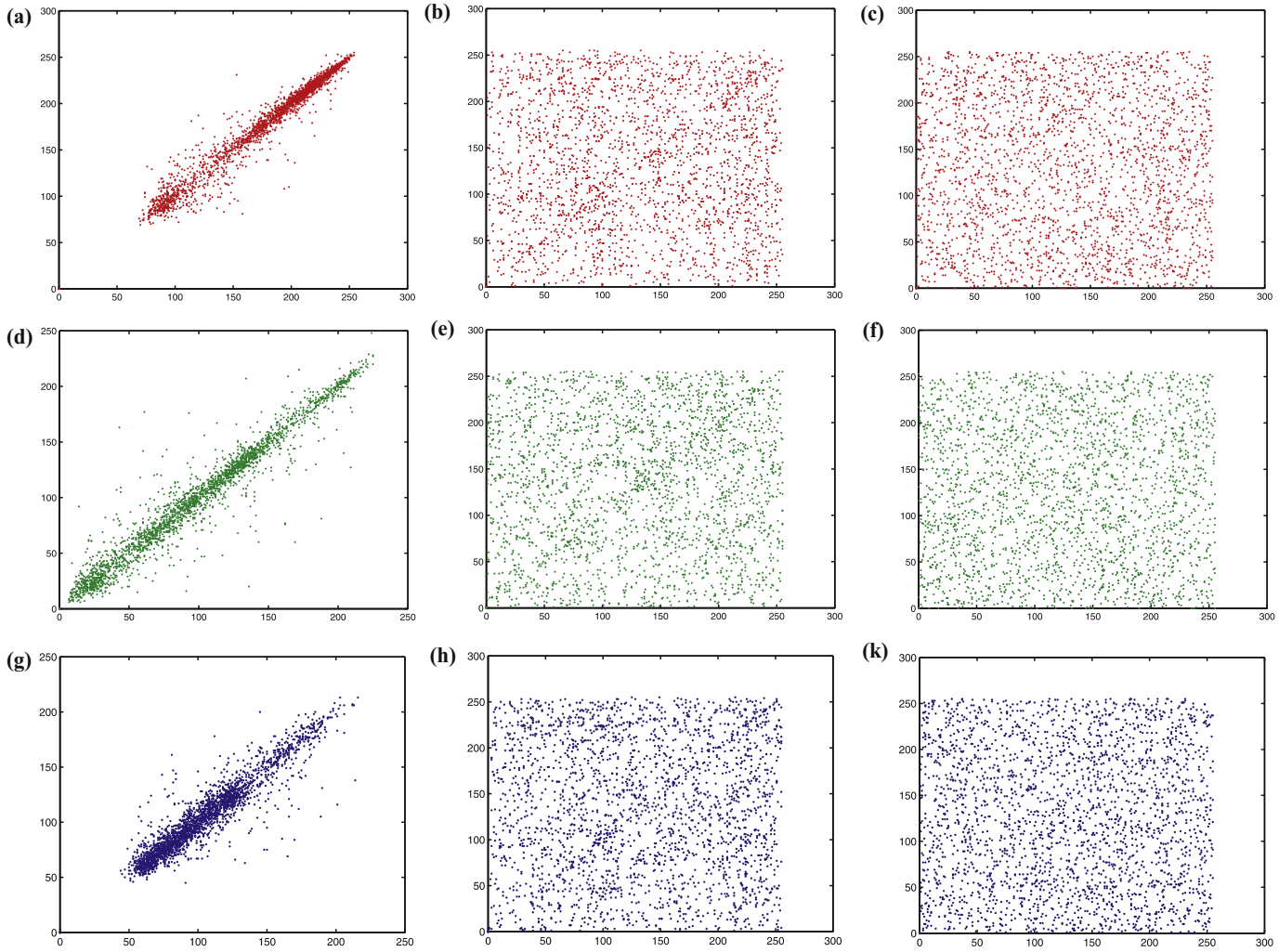

Fig. 3. Correlation analysis of two horizontally adjacent pixels in (a) plain-image Lena – R, (b) image Lena – R after random circular shift process, (c) image Lena – R after the diffusion process, (d) plain-image Lena – G, (e) image Lena – G after random circular shift process, (f) image Lena – G after the diffusion process, (g) plain-image Lena – B, (h) image Lena – B after random circular shift process, and (k) image Lena – B after the diffusion process.

Table 2
The related correlation coefficient between plaintext and ciphertext.

Scan direction	Lena					
	Plain-image			Cipher-image		
	R	G	B	R	G	B
Horizontal	0.982825	0.972527	0.972527	0.004751	0.000534	0.000813
Vertical	0.989545	0.982498	0.982498	0.000594	0.002448	0.002454
Diagonal	0.970403	0.958541	0.958541	0.001418	0.001380	0.000173

between R, G and B components of plain-image and cipher-image. Noticeably, these results prove that the proposed cryptosystem has greatly reduced the correlations among three components. As can be seen in Table 5, the proposed method has the satisfactory performance in the horizontal, diagonal and vertical directions.

4.2. Sensitivity analysis

4.2.1. Difference attacks

As a general requirement for all image encryption algorithms, the encrypted image should be greatly different from its original

Table 3
Similar position correlations between R, G and B components.

Scan direction	R – G	R – B	G – B
Plain-image	0.935783	0.981709	0.961709
Cipher-image	0.000332	0.001941	0.000136

Table 4
Adjacent position correlations between R, G and B components.

Scan direction	R – G	R – B	G – B
Plain-image	0.966437	0.952341	0.974356
Cipher-image	0.000264	0.000302	0.001181

Table 5
Performance analysis of the proposed method with recent methods using Lena image correlation coefficients of pairs of adjacent pixels in different directions.

Scan direction	Horizontal	Vertical	Diagonal
Original image	0.975963	0.984847	0.962495
Proposed algorithm	0.0002032	0.0001832	0.0001511
Mazloom and Eftekhari-Moghadam (2009)	0.007539	0.012878	0.004914
Norouzi et al. (2013a)	0.000821	0.000842	0.000508
El-Latif et al. (2013)	0.002481	0.006847	0.002366
Bakhshandeh and Eslami (2013)	0.005327	0.009524	0.008915
Tong (2012)	0.004146	0.008956	0.016375
Liu et al. (2013b)	0.002425	0.058025	0.017012
Seyedzadeh and Mirzakuchaki (2012)	0.000550	0.000839	0.001124

Table 7
Comparison of the average NPCR and UACI values with one bit different between the plain-images for different schemes.

Algorithm	NPCR	UACI
Proposed	0.996806	0.334911
Mazloom and Eftekhari-Moghadam (2009)	0.996023	0.334451
Norouzi et al. (2013a)	0.996137	0.334594
El-Latif et al. (2013)	0.996498	0.334278
Bakhshandeh and Eslami (2013)	0.993220	0.333161
Tong (2012)	0.994845	0.334228
Liu et al. (2013b)	0.996644	0.334703
Seyedzadeh and Mirzakuchaki (2012)	0.996720	0.334904

form. In order to test the influence of changing a single pixel in the original image on the encrypted image, we have measured the number of pixels change rate by calculating the Number of Pixel Change Rate (*NPCR*) (Seyedzadeh et al., 2010) using Eq. (28), and the Unified Average Changing Intensity (*UACI*) (Norouzi et al., 2013b) for the two encrypted images using Eq. (29):

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100 \tag{28}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C(i,j) - C'(i,j)|}{255}\right] \times 100 \tag{29}$$

where W and H are the width and height of the encrypted image. We use two encrypted images C and C', whose corresponding original images are different in only one pixel. We also define a two-dimensional array D, which has the same size as C and C'. If $C(i, j) = C'(i, j)$, then $D(i, j) = 0$ otherwise, $D(i, j) = 1$. It is clear that in order to resist differential attack, *NPCR* and *UACI* values should be large enough for an ideal cipher system.

To test our proposed algorithm, the RGB color image is first encrypted, then one pixel in the image is randomly selected and toggled. The modified image is encrypted again by the same key so as to generate a new cipher-image. Finally, the *NPCR* and *UACI* values are calculated. This kind of test is performed over 400 times with different images. The resulting average *NPCR* and *UACI* values are listed in Table 6. According to the encryption transformation used in Eq. (22), encrypting each pixel of the original image depends on the sum of previous encrypted values, the sum of subsequent plain values and the generated keystreams. Therefore, as a result of this dependency and also the coupling among encrypted values of gray-level components, a swift change in the original image will result in a significant change in the cipher-image. Table 7 provides the data related to experimental results obtained by the proposed method and other cryptosystems. It is obvious from the simulation results that the proposed cryptosystem achieves high performance by having *NPCR* > 0.996809 and *UACI* > 0.334919.

4.2.2. Key Sensitivity test

Key sensitivity analysis has been performed for the proposed image encryption algorithm and the results are summarized as follows:

Assume that a 16-character cipher key is used. This means that the key consists of 128 bits. A typical key sensitivity test has been performed according to the following steps: First, an image is encrypted using the test key, Key1 = "28,db,a5,60,30,6d, 7b,1e,96,39,62,95,40,4a,9d,43". Then, the least significant bit in the 16th character of the key is changed and Key2 = "28,db,a5,60,30, 6d,7b,1e,96,39,62,95,40,4a,9d,42" is obtained which is used to encrypt the same image. Finally, two cipher-images encrypted by the two slightly different keys are compared. This test shows that although the two keys are different in only one bit, there is a difference of up to 99.60976% in terms of gray-level values between the image encrypted by Key1 and the image encrypted by Key2. Fig. 4a– d shows the test results. Also when a 16-character key is used to encrypt an image and another trivially modified key is used to decrypt the cipher-image, the decryption will completely fail (see Fig. 4e and h). As discussed in Section 3.1, the nonlinear transformation used in Eqs. (10) is designed so that initial conditions and parameters of the CFI are greatly sensitive to the change in even one bit of the secret key. The mathematical analyses shown in Fig. 1 and Table 1 truly assure that the CFI has high complexity and also the randomness of the sequences generated by this type of the map is very high; as a result, the proposed scheme can resist brute-force attack.

The average pixel differences of some well-known images are tabulated in Table 8 using several random keys. All the cases with one-bit difference are computed for each key. Table 9 provides data to quantitatively compare average results of the key sensitivity test performed on the proposed scheme and other schemes. Results

Table 6
The average NPCR and UACI values of the proposed algorithm.

Image name	Image size 512 × 512		Image size 1024 × 1024	
	NPCR	UACI	NPCR	UACI
Sailboat	0.996773	0.335098	0.996875	0.334979
Lena	0.996836	0.334749	0.996738	0.334965
Baboon	0.996734	0.334949	0.996751	0.334879
House	0.996859	0.334885	0.996888	0.334977
Avion	0.996744	0.335044	0.996854	0.334740
Peppers	0.996717	0.334852	0.996764	0.335199
Butfish	0.996771	0.334799	0.996824	0.334760
Blueeye	0.996790	0.334819	0.996885	0.335165
Woodsplint	0.996766	0.334946	0.996868	0.334995
Laurel	0.996839	0.334769	0.996857	0.334829
Baluchisrug	0.996775	0.334895	0.996876	0.334813
Metaldetail	0.996741	0.334820	0.996789	0.334945

Table 8
The average NPCR and UACI values of the proposed algorithm.

Image name	Image size 512 × 512		Image size 1024 × 1024	
	NPCR	UACI	NPCR	UACI
Sailboat	0.996193	0.334661	0.996198	0.334664
Lena	0.996078	0.334771	0.996165	0.334828
Baboon	0.996190	0.334534	0.996246	0.334575
House	0.996191	0.334693	0.996220	0.334509
Avion	0.996145	0.334845	0.996093	0.334533
Peppers	0.996063	0.334671	0.996204	0.334802
Butfish	0.996068	0.334538	0.996132	0.334553
Blueeye	0.996170	0.334785	0.996156	0.334786
Woodsplint	0.996243	0.334540	0.996210	0.334514
Laurel	0.996178	0.334837	0.996072	0.334506
Baluchisrug	0.996085	0.334748	0.996152	0.334614
Metaldetail	0.996099	0.334769	0.996105	0.334847

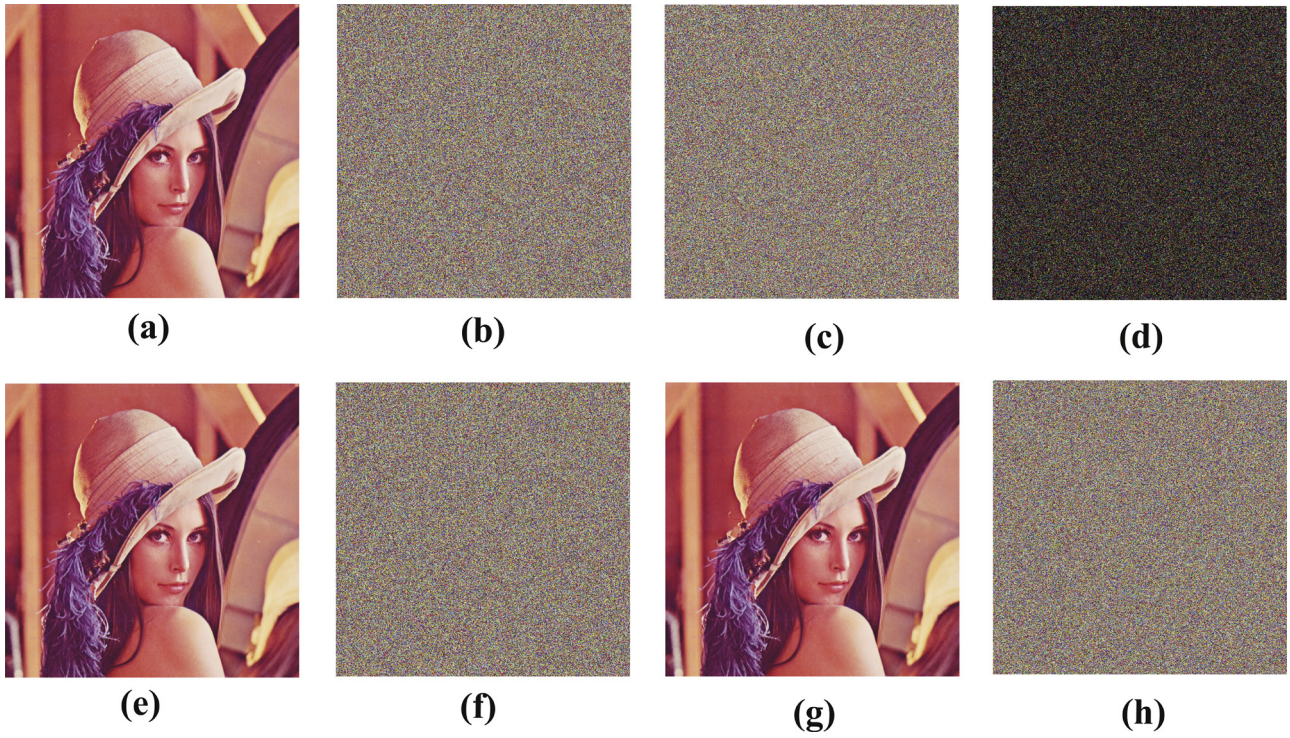

Fig. 4. Key sensitivity result: (a) Lena's the original image, (b) encrypted image with Key1, (c) encrypted image with Key2, (d) difference image, (e) Lena's the original image, (f) encrypted image with Key1, (g) decrypted image with Key1, and (h) decrypted image with Key2.

Table 9
Comparison of pixel difference between images encrypted by random keys with one-bit difference.

Algorithm	NPCR	UACI
Proposed	0.996151	0.334671
Mazloom and Eftekhari-Moghadam (2009)	0.816595	0.265631
Norouzi et al. (2013a)	0.996384	0.336305
El-Latif et al. (2013)	0.996117	0.322345
Bakhshandeh and Eslami (2013)	0.992225	0.332161
Tong (2012)	0.995971	0.332617
Liu et al. (2013b)	0.996024	0.333126
Seyedzadeh and Mirzakuchaki (2012)	0.996093	0.334621

indicate that the sensitivity obtained in the proposed method is very close to the expected value of the pixel difference on two randomly generated images (NPCR = 99.6156% and UACI = 33.4678%). Moreover, almost all the obtained values are in favor of the proposed scheme.

4.2.3. Avalanche criterion

We know the change of one bit in the plaintext should result in theoretically 50% difference in the cipher's bits (Norouzi and Mirzakuchaki, 2014). Hence, for proving so-called sensitivity to plaintext, two plain-images are generated with just one-pixel difference. The changing rate obtained by the proposed algorithm is 49.99998%. Hence, the avalanche criterion of the proposed algorithm is very close to the ideal value of 50%. As mentioned in Section 3, the encryption transformation used in the proposed algorithm is such that the change of one bit in the plaintext results in a significant change in the cipher's bits. The comparison of the results between the proposed algorithm and other algorithms shown in Table 10 prove this claim.

4.3. Information entropy analysis

The entropy (such as K–S entropy, information entropy) is the most outstanding feature of the randomness (Li, 1991). To calculate the entropy $H(s)$ of a source s, we have:

$$H(s) = \sum_{i=0}^{2^N-1} P(s_i)\log_2 \frac{1}{P(s_i)} \tag{30}$$

where $P(s_i)$ represents the probability of symbol s_i. For a purely random source emitting 2^N symbols, the entropy is $H(s) = N$. If the output of a cipher emits symbols with entropy less than N, there is a certain degree of predictability which threatens its security. Table 11 provides the data related to the comparison of the entropy between the proposed algorithm and other algorithms. Apparently,

Table 10
Avalanche test.

Algorithm	Avalanche criterion
Proposed	0.499999
Mazloom and Eftekhari-Moghadam (2009)	0.498201
Norouzi et al. (2013a)	0.499992
El-Latif et al. (2013)	0.499987
Bakhshandeh and Eslami (2013)	0.492646
Tong (2012)	0.498719
Liu et al. (2013b)	0.499923
Seyedzadeh and Mirzakuchaki (2012)	0.500122

Table 11
Results of information entropy.

Algorithm	Entropy
Proposed	7.99991
Mazloom and Eftekhari-Moghadam (2009)	7.99683
Norouzi et al. (2013a)	7.99930
El-Latif et al. (2013)	7.99978
Bakhshandeh and Eslami (2013)	7.99920
Tong (2012)	7.99920
Liu et al. (2013b)	7.99886
Seyedzadeh and Mirzakuchaki (2012)	7.99923

Table 12
DIEHARD tests suite for the Lena image.

Test name	*P*-Value	Result
Birthday spacing	0.895683	Success
Overlapping permutation	0.168750	Success
Binary rank 31 × 31	0.481099	Success
Binary rank 6 × 8	0.555067	Success
Bitstream	0.470586	Success
OPSO	0.631454	Success
OQSO	0.275547	Success
DNA	0.879836	Success
Count the ones 01	0.566558	Success
Count the ones 02	0.324472	Success
Parking lot	0.932247	Success
Minimum distance	0.891747	Success
3DS spheres	0.202790	Success
Squeeze	0.715954	Success
Overlapping sum	0.556131	Success
Runs	0.490834	Success
CrapsRa	0.543112	Success

the proposed algorithm is much closer to the ideal situation. This means that information leakage in the encryption process is negligible and the encryption system is secure against the entropy attack.

4.4. Randomness tests for the cipher

In this paper, we have used DIEHARD (Marsaglia, 1995), ENT test suite (Walker, 2008) and NIST SP 800-22 Tests (Rukhin et al., 2001) to test the randomness of the cipher. The main goal of these tests is to focus on different types of possible non-randomness in the sequence. Some of these tests consist of a number of subsets. To carry on these tests, we have used 200 sequences of ciphers with the length of 1,000,000 bits. The encrypted image is the 24-bit image. To test the randomness of the cipher, a number of initial keys are used. The results of the tests are shown in Tables 12–14 . By analyzing these results, it can be concluded that our proposed image encryption algorithm successfully passes the DIEHARD, ENT and NIST SP 800-22 Tests. Hence, we can claim that the generated ciphers in our cryptosystem are completely random.

4.5. Key space analysis

Key space size is the total number of different keys that can be used in the encryption process. The key space should be large enough to make brute-force attacks infeasible. From the cryptographic point of view, the size of the key space should not be smaller than 2^{100} to provide a high level of security (Schneier, 1996; Stinson, 2006). Since the secret key is 128-bit long, the key space is about 2^{128}, which is sufficient to resist the brute-force attack.

4.6. Image encryption speed test

We have analyzed the speed of the proposed image encryption/decryption scheme using Eclipse 3.5 on a personal computer (PC) with a 3.0 GHz Intel(R) Core2Duo CPU, 3.25 GB RAM, and with Windows XP as the operating system. In Table 15, we have compared our encryption scheme with other encryption schemes in terms of speed. Simultaneous generation of three pseudo-random numbers by the CFI in each round and also the parallel operation of the keystream generation process and the permutation-diffusion process on image arrays lead to high encryption speed. From obtained results, we can see that the proposed scheme still has enough speed for real time image transmission over broadband networks, where the encryption and decryption time should be short relative to the transmission time.

Table 13
Max grade of ENT test suite for the Lena image.

Test name	Average value	Result
Entropy	7.99998	Success
Arithmetic mean	127.434	Success
Monte Carlo	3.13945	Success
Chi square	244.782	Success
SCC	0.00019	Success

Table 14
Results of the SP800-22 tests suite for cipher Lena image.

Test name		*P*-Value	Result
Frequency		0.397709	Success
Runs (M = 10,000)		0.596768	Success
Block-frequency		0.231495	Success
Long runs of ones		0.399734	Success
Rank		0.862254	Success
Spectral DFT		0.029755	Success
No overlapping templates		0.452036	Success
Overlapping templates		0.712751	Success
Universal		0.933784	Success
Lempel ziv complexity		0.095855	Success
Linear complexity		0.244074	Success
Serial	*P*-value1	0.180302	Success
Serial	*P*-value2	0.788686	Success
Approximate entropy		0.548857	Success
Cumulative sums forward		0.143681	Success
Cumulative sums reverse		0.290144	Success
Random excursions	$X=-4$	0.498011	Success
	$X=-3$	0.324424	Success
	$X=-2$	0.865800	Success
	$X=-1$	0.485635	Success
	$X=1$	0.131939	Success
	$X=2$	0.756464	Success
	$X=3$	0.523783	Success
	$X=4$	0.431249	Success
Random excursions variant	$X=-9$	0.729366	Success
	$X=-8$	0.869204	Success
	$X=-7$	0.643319	Success
	$X=-6$	0.404386	Success
	$X=-5$	0.864789	Success
	$X=-4$	0.110367	Success
	$X=-3$	0.246403	Success
	$X=-2$	0.460700	Success
	$X=-1$	0.060552	Success
	$X=1$	0.766127	Success
	$X=2$	0.906333	Success
	$X=3$	0.283179	Success
	$X=4$	0.288749	Success
	$X=5$	0.588453	Success
	$X=6$	0.358960	Success
	$X=7$	0.859502	Success
	$X=8$	0.706422	Success
	$X=9$	0.295954	Success

Table 15
Comparison of encryption/decryption speed of the proposed scheme and other schemes.

Algorithm	Speed (Mbit/s)
Proposed	32.4564
Mazloom and Eftekhari-Moghadam (2009)	9.21151
Norouzi et al. (2013a)	6.75662
El-Latif et al. (2013)	37.1621
Bakhshandeh and Eslami (2013)	24.4879
Tong (2012)	21.5443
Liu et al. (2013b)	28.7123
Seyedzadeh and Mirzakuchaki (2012)	44.9389

5. Conclusions

In this paper, a novel pseudo-random keystream generator based on the CFI for the image encryption is proposed. To achieve

high security and high sensitivity, our proposed scheme presents three solutions: (1) designing a new pseudo-random keystream generator with the non-linearity structure, (2) using a 128-bit long secret key for generating the initial conditions and parameters of the CFI by making some algebraic transformations to the secret key, and (3) coupling components of the RGB color image and keystreams generated by the CFI for increasing the resistance of the cryptosystem against plaintext attack. The mathematical analysis in the paper truly assures that the CFI has high complexity and also the randomness of the sequences generated by this type of the map is very high which extremely increases the security and sensitivity of the proposed algorithm. In this cryptosystem, the three processes of the keystream generation, the random circular shift process and the diffusion process are designed in such a way to strengthen the security and sensitivity of the cryptosystem. Furthermore, simultaneous generation of three pseudo random numbers by CFI in each round causes the proposed cryptosystem to achieve a satisfactory encryption speed. We have carried out several security and performance analysis based on several tests. Results show that the proposed method is suitable for practical uses to protect the security of digital image information over the Internet.

Acknowledgements

The authors would like to thank the Editor and the anonymous Referees for their valuable comments and suggestions to improve this paper.

Appendix A. Determination of membership grades by using γ–fuzzy measure

As described in Section 2, γ-fuzzy measure for three inputs can be obtained by solving the following equation

$$1+\gamma=\prod_{i=1}^{i=3}(1+\gamma g_i)=(1+\gamma g_1)(1+\gamma g_2)(1+\gamma g_3)$$
$$=1+\gamma(g_1+g_2+g_3)+\gamma^2(g_1g_2+g_1g_3+g_2g_3)+\gamma^3(g_1g_2g_3) \quad (A.1)$$

Eq. (A.1) can be written as

$$\gamma^2(g_1g_2g_3)+\gamma(g_1g_2+g_1g_3+g_2g_3)+(g_1+g_2+g_3)=0 \quad (A.2)$$

For having the real value of γ, Δ should be bigger than zero

$$\Delta=(g_1g_2+g_1g_3+g_2g_3)^2-4g_1g_2g_3(g_1+g_2+g_3)>0$$
$$=g_1^2g_2^2+g_2^2g_3^2+g_1^2g_3^2-2(g_1^2g_2g_3+g_1g_2^2g_3+g_1g_2g_3^2)>0$$
$$=(g_2g_3-g_1g_2-g_1g_3)^2-4g_1^2g_2g_3>0 \quad (A.3)$$

We set $g_1=0.5$ to solve Eq. (A.3)

$$\Delta=\left(g_2g_3-\frac{g_2+g_3}{2}\right)^2-g_2g_3>0 \quad (A.4)$$

Eq. (A.4) is true for $g_2>2$ and $g_3>2$,

According to Eq. (A.1), we solve Eq. (A.2) so that it satisfies $\gamma>-1$

$$\gamma=\frac{\sqrt{\left(g_2g_3-((g_2+g_3)/2)\right)^2-g_2g_3}-\left(((g_2+g_3)/2)+g_2g_3\right)}{g_2g_3}>-1 \quad (A.5)$$

Eq. (A.5) can be written as

$$\left(g_2g_3-\frac{g_2+g_3}{2}\right)^2-g_2g_3>\left(\frac{g_2+g_3}{2}\right)^2 \quad (A.6)$$

This gives

$$g_2g_3(g_2g_3-g_2-g_3-1)>0 \quad (A.7)$$

Recalling that g_2 and g_3 are positive integers

$$g_2>\frac{g_3+1}{g_3-1},\qquad g_3\neq 1 \quad (A.8)$$

From Eqs. (A.4) and (A.8), we can conclude that membership grades should be selected as $g_1=0.5$, $g_2>2$ and $g_3>2$ to satisfy the real value of $\gamma>-1$.

References

Abuturab, M.R., 2013. Color information security system using Arnold transform and double structured phase encoding in gyrator transform domain. Opt. Laser Technol. 45, 525–532.
Aurell, E., Boffetta, G., Crisanti, A., Paladin, G., Vulpiani, A., 1997. Predictability in the large: an extension of the concept of Lyapunov exponent. J. Phys. A: Math. Gen. 30, 1–26.
Bakhshandeh, A., Eslami, Z., 2013. An authenticated image encryption scheme based on chaotic maps and memory cellular automata. Opt. Lasers Eng. 51, 665–673.
Behnia, S., Akhavan, A., Akhshani, A., Samsudin, A., 2013. Image encryption based on the Jacobian elliptic maps. J. Syst. Softw. 86, 2429–2438.
Chen, W., Chen, X., 2011. Optical color image encryption based on an asymmetric cryptosystem in the Fresnel domain. Opt. Commun. 284, 3913–3917.
Chiang, J.-H., 2000. Aggregating membership values by a Choquet-fuzzy-integral based operator. Fuzzy Sets Syst. 114, 367–375.
Çokal, C., Solak, E., 2009. Cryptanalysis of a chaos-based image encryption algorithm. Phys. Lett. A 373, 1357–1360.
El-Latif, A.A.A., Li, L., Wang, N., Han, Q., Niu, X., 2013. A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces. Signal Process. 93, 2986–3000.
Friedman, M., Wang, M., Kandel, A., 1996. Numerical methods for calculating the fuzzy integral. Fuzzy Sets Syst. 83, 57–62.
Gao, T., Chen, Z., 2008a. Image encryption based on a new total shuffling algorithm. Chaos Soliton Fractals 38, 213–220.
Gao, T., Chen, Z., 2008b. A new image encryption algorithm based on hyper-chaos. Phys. Lett. A 372, 394–400.
Grabisch, M., Nguyen, H.T., 1994. Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference. Kluwer Academic Publishers, Dordrecht.
Huang, F., Feng, Y., 2009. Security analysis of image encryption based on two-dimensional chaotic maps and improved algorithm. Front. Electr. Electron. Eng. China 4, 5–9.
Li, W., 1991. Some Lagrange multiplier tests for seasonal differencing. Complex Syst. 5, 381–384.
Lichtenberg, A.J., Lieberman, M.A., 1992. Regular and Chaotic Dynamics. Springer-Verlag, New York.
Liu, H., Wang, X., 2011. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt. Commun. 284, 3895–3903.
Liu, H., Wang, X., 2013. Triple-image encryption scheme based on one-time key stream generated by chaos and plain images. J. Syst. Softw. 86, 826–834.
Liu, Y., Tong, X., Hu, S., 2013a. A family of new complex number chaotic maps based image encryption algorithm. Signal Process. Image Commun. 28, 1548–1559.
Liu, H., Wang, X., Kadir, A., 2013b. Color image encryption using Choquet fuzzy integral and hyper chaotic system. Optik 124, 3527–3533.
Lou, D.-C., Sung, C.-H., 2004. A steganographic scheme for secure communications based on the chaos and Euler theorem. IEEE Trans. Multimed. 6, 501–509.
Marsaglia, G., 1995. Diehard: A Battery of Tests for Random Number Generators, CD-ROM. Department of Statistics and Supercomputer Computations Research Institute, Florida State University http://www.stat.fsu.edu/pub/diehard/
Mazloom, S., Eftekhari-Moghadam, A., 2009. Color image encryption based on coupled nonlinear chaotic map. Chaos Soliton Fractals 42, 1745–1754.
Murofushi, T., Sugeno, M., 1989. An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure. Fuzzy Sets Syst. 29, 201–227.
Norouzi, B., Mirzakuchaki, S., 2014. A fast color image encryption algorithm based on hyper-chaotic systems. Nonlinear Dynam., 1–21.
Norouzi, B., Mirzakuchaki, S., Seyedzadeh, M., Mosavi, S., 2012. A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process. Multimed. Tools Appl., 1–29.
Norouzi, B., Seyedzadeh, S., Mirzakuchaki, S., Mosavi, M., 2013a. A novel image encryption based on hash function with only two-round diffusion process. Multimed. Syst., 1–20.
Norouzi, B., Seyedzadeh, S., Mirzakuchaki, S., Mosavi, M., 2013b. A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos. Multimed. Tools Appl., 1–31.
Pareek, N.K., Patidar, V., Sud, K.K., 2013. Diffusion and substitution based gray image encryption scheme. Digit. Signal Process. 23, 894–901.
Ralescu, A., Ralescu, D., 1997. Extensions of fuzzy aggregation. Fuzzy Sets Syst. 86, 321–330.
Rhouma, R., Belghith, S., 2008. Cryptanalysis of a new image encryption algorithm based on hyper-chaos. Phys. Lett. A 372, 5973–5978.
Rukhin, A., et al., 2001. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22.

Schneier, B., 1996. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. John Wiley Sons, New York, 2nd ed.
Seyedzadeh, S.M., Hashemi, Y., 2011. Image encryption algorithm based on Choquet fuzzy integral with self-adaptive pseudo-random number generator. In: 2011 IEEE International Conference on Intelligent Systems Design and Applications, pp. 642–647.
Seyedzadeh, S.M., Mirzakuchaki, S., 2011. Image encryption scheme based on Choquet fuzzy integral with pseudo-random keystream generator. In: 2011 IEEE International Symposium on Artificial Intelligence and Signal Processing, pp. 101–106.
Seyedzadeh, S.M., Mirzakuchaki, S., 2012. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. Signal Process. 92, 1202–1215.
Seyedzadeh, S.M., Mirzakuchaki, S., Atani, R., 2010. A novel image encryption algorithm based on hash function. In: 2010 IEEE Iranian Machine Vision and Image Processing, pp. 1–6.
Seyedzadeh, S.M., Moosavi, S.M.S., Mirzakuchaki, S., 2011. Using self-adaptive coupled piecewise nonlinear chaotic map for color image encryption scheme. In: 2011 IEEE Iranian Conference on Electrical Engineering, pp. 1–6.
Stinson, D.R., 2006. Cryptography: Theory and Practice, 3rd ed. CRC Press, Boca Raton.
Sugeno, M., Murofushi, T., 1987. Pseudo-additive measures and integrals. J. Math. Anal. Appl. 122, 197–222.
Sugeno, M., 1977. Fuzzy Measures and Fuzzy Integrals. North Holland, Amsterdam, The Netherlands, pp. 89–102.
Sui, L., Gao, B., 2013. Color image encryption based on gyrator transform and Arnold transform. Opt. Laser Technol. 48, 530–538.
Tong, X.j., 2012. The novel bilateral diffusion image encryption algorithm with dynamical compound chaos. J. Syst. Softw. 85, 850–858.
Walker, J., 2008. Ent: A Pseudorandom Number Sequence Test Program, Fourmilab. ch. http://www.fourmilab.ch/random/
Wang, Z., Klir, G., Wang, W., 1996. Monotone set functions defined by Choquet integral. Fuzzy Sets Syst. 81, 241–250.
Wei, X., Guo, L., Zhang, Q., Zhang, J., Lian, S., 2012. A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. J. Syst. Softw. 85, 290–299.
Wolf, A., Swift, J., Swinney, H., Vastano, J., 1985. Determining Lyapunov exponents from a tile series. Physica 16, 285–317.
Yuen, C.-H., Wong, K.-W., 2011. A chaos-based joint image compression and encryption scheme using dct and sha-1. Appl. Soft Comput. 11, 5092–5098.
Zhu, Z., Zhang, W., Wong, K., Yu, H., 2011. A chaos-based symmetric image encryption scheme using a bit-level permutation. Inform. Sci. 181, 1171–1186.

Seyed Mohammad Seyedzadeh received the B.S and M.S degrees in electrical engineering from the Shiraz University of Technology and Iran University of Science and Technology, respectively. He has been a member of the Electronic Research Center of Electrical Engineering at the Iran University of Science and Technology. He is currently a Ph.D. student in computer engineering at University of Pittsburgh.. His main research interests include coding theory with applications to non-volatile memories, associative memories and data storage.

Benyamin Norouzi received the B.S and M.S degrees in electrical engineering from the Hakim Sabzevari University, Sabzevar, Iran and Iran University of Science and Technology, Tehran, Iran in 2010 and 2012, respectively. He is currently a Ph.D. Candidate of Electrical Engineering at Iran University of Science and Technology. His research interest includes: Cryptography, Multimedia Security, Network Security and Image Processing.

Sattar Mirzakuchaki received the B.S. degree in electrical engineering from University of Mississippi in USA in 1989 and the M.Sc. and Ph.D. also in Electrical Engineering from the University of Missouri in 1991 and 1996 respectively. He is currently an assistant professor in the electrical engineering department at IUST. His research interests include Cryptography, Image Processing, growth and characterization of semiconductor devices and VLSI design.