



# Bit-level color image encryption algorithm based on coarse-grained logistic map and fractional chaos

Yu-jie Sun<sup>1</sup> · Hao Zhang<sup>2</sup> · Xing-yuan Wang<sup>3</sup> · Ming-xu Wang<sup>4</sup>

Received: 14 September 2019 / Revised: 14 September 2020 / Accepted: 22 December 2020 /  
Published online: 9 January 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

## Abstract

A novel color image encryption algorithm based on coarse-grained fractional chaotic system signals is proposed in this paper. First, color images are divided into three channels, which are encrypted based on the corresponding three states of the chaotic system. Second, the chaotic systems are defined as fractional chaotic, in which the fractional order enlarges the parameter space. Third, the fractional chaotic signals are handled with unfixed coarse-grained methods instead of being utilized directly. In addition, the original image and the chaotic signals are divided into bit signals from the pixel values, and the high and low bits are encrypted, respectively. To demonstrate the effectiveness and robustness of the proposed color image encryption algorithm, its properties, including the key space, information entropy, correlation analysis, key sensitivity, and resistance to differential attacks, are provided using a numerical simulation.

**Keywords** Image encryption · Fractional systems · Coarse-grained · Bit-level · Color image

## 1 Introduction

Owing to the rapid development of computer science and big data technology, text processing, particularly digital image processing, has recently attracted widespread attention of researchers and has become a hot research field [1, 18, 21, 29, 30]. Digital images typically show various texts; in addition, encryption is needed in almost all aspects of modern life, and the

---

✉ Hao Zhang  
zhangh545@126.com

<sup>1</sup> School of Computer Science & Technology, BeiJing Institute of Technology, Beijing 100081, China

<sup>2</sup> College of Information & Computer, Taiyuan University of Technology, Taiyuan 030024, China

<sup>3</sup> School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

<sup>4</sup> Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China

requirements for security and humanity are increasing. Because the amount of data is large and the pixels in adjacent areas are strongly correlated in general images, traditional image processing algorithms, including RSA [5] and DES [6], are unsuitable, and the development of new image encryption schemes is challenging.

By contrast, a chaotic signal is widely utilized as an external key in image encryption owing to its pseudo-randomness and sensitivity to the initial conditions. Chaotic keys refer to the uncertain signals generated by certain systems, and they have potential applications in physics, biology, chemistry, and medical analysis [4, 7, 12, 14, 15, 23, 25, 27, 28]. For chaotic image encryption, many studies have been conducted, based upon which, it has been proven that the use of chaos is a superior method for image encryption [11, 13, 24, 33].

Traditionally, image encryption has been divided into pixel- and bit-levels [8, 16, 20, 31, 32, 34]. Although pixel-level image encryption is fast, bit-level encryption achieves better results; thus, both methods have been utilized in recent studies. For example, Yao et al. [31] obtained an encrypted image by scrambling the image pixels and diffusing the image information using a chaotic logistic map. However, other researchers have focused on bit-level image encryption. For instance, Teng et al. [20] proposed an image encryption algorithm with bit-level scrambling. Gray image encryption [8, 11, 13, 16, 20, 24, 32, 33] and color image encryption [9, 17, 19, 26, 31] have also been extensively researched. For example, Faragallah et al. [9] proposed a color image cryptosystem with a chaotic baker map, and Parvaz et al. [17] designed a combined chaotic system and applied it to color image encryption.

In general, chaotic signals are directly used in image encryption. However, the generated signals are fixed once the initial values and parameters of the chaotic system are selected. Differing from previous studies, in the present study, coarse-grained methods are applied to chaotic signals before the signals are used as keys, which makes the signals unpredictable while applying different coarse-grained methods. In addition, to obtain various chaotic signals, a discrete logistic map and a coupled fractional lattice chaotic system (CFLCS) are considered during the encryption process. In addition, image information is divided into high and low bits throughout the entire encryption process. During the scrambling, with the generated coarse-grained fractional chaotic signals, the high bits are row- and column-scrambled. By contrast, the low bits are global scrambled using coarse-grained logistic chaotic map signals. During the diffusion process, the bit streams are also handled through a cyclic shift and XOR operation with coarse-grained chaotic signals. Motivated by the aforementioned analysis, a novel bit-level color image encryption algorithm with different coarse-grained chaotic signals is proposed in this paper. The contributions of this paper are as follows: 1. The coarse-grained method is utilized to handle different kinds of chaotic signals and make the signals unpredictable. 2. Different scrambling methods are used in the high and low bits simultaneously. 3. The bit planes are decomposed, and parallel processing is utilized in different planes for both scrambling and diffusion. 4. Chaotic signals for the encryption of different planes in a color image are concurrently generated using CFLCS. The remainder of this paper is organized as follows. In the following section, the main image encryption algorithm is proposed. Section 3 provides the experimental results and corresponding analysis. Finally, some concluding remarks are given in section 4.

## 2 The coarse-grained chaotic system and encryption algorithm

In this section, the coarse-grained chaos-based image encryption algorithm is described. For greater efficiency and security, a discrete chaotic map and continuous fractional chaotic system

are considered. First, take the following logistic map (1) as an example of a discrete chaotic map. The equation can be defined as

$$x(k+1) = \mu x(k)(1-x(k)), \quad k = 0, 1, 2, \dots \quad (1)$$

where discrete state  $x(k) \in (0, 1)$  and control parameter  $\mu \in [0, 4]$ . When  $3.5699456 < \mu \leq 4$ , the Logistic map is chaotic.

In the previous study, once the initial value and system parameter of system (1) are fixed, the chaotic signals generated are fixed because of the fine-grained iteration step. If the initial value and system parameter are known, the keys will no longer be safe. To make the chaos signal more secure, coarse-grained methods will be applied in the signals before encryption. For example, a coarse-grained parameter  $C$  can be brought in, and a key signal with  $C$  chaotic signals can be generated. For the sake of simplicity, the mean value of the chaotic signals is calculated as the final key in this paper. With the same initial value,  $x_0 = 0.3874926$ , and system parameter,  $\mu = 3.999999$ , but different coarse-grained parameters 4 and 8, the derived key signals are as shown in Fig. 1. The corresponding NIST test results are shown as following Table 1.

From Table 1, it is exhibited that all  $p$ -values for different coarse-grained parameters are greater than 0.01, and NIST tests are all passed. This means that the proposed coarse-grained signals are sufficiently good and it can be utilized in the code applications which require randomness.

However, for a continuous chaotic system signal, the coarse-grained method will be more suitable for dealing with an analog signal because the discretization will be completed during the coarse-grained process. As a result, the derived key signal can be directly utilized to encrypt the digital images. In the following section, a coarse-grained simulation based on fractional chaotic signals is described. Before the simulation, the following fractional definition should first be introduced.

Definition 1 (Caputo differential definition) [3]:

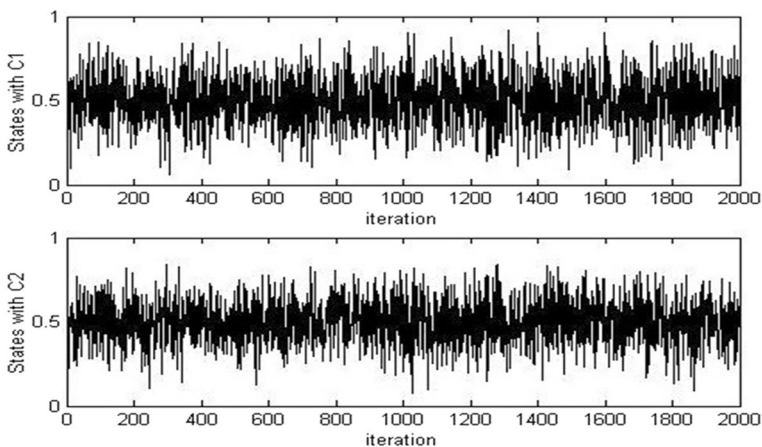


Fig. 1 Discrete key signals with different coarse-grained parameters

**Table 1** The test results of NIST SP800–22 for different coarse-grained parameters

Test	P-Values(C=4)	Pass or not	P-Values(C=8)	Pass or not
Frequency (Monobit) test	0.3202	Pass	0.6284	Pass
Frequency test	0.6320	Pass	0.2166	Pass
Runs test	0.2770	Pass	0.2484	Pass
Longest-run-of-ones in a block	0.9371	Pass	0.0105	Pass
Binary matrix rank test	0.2583	Pass	0.0192	Pass
Discrete Fourier transform test	0.3098	Pass	0.2577	Pass
Non-overlapping template matching test	0.5391	Pass	0.1977	Pass
Overlapping template matching test	0.9866	Pass	0.2781	Pass
Maurer’s universal statistical test	0.6392	Pass	0.9512	Pass
Liner complexity test	0.7062	Pass	0.0671	Pass
Serial test (p-value1)	0.1149	Pass	0.0168	Pass
Serial test (p-value2)	0.2121	Pass	0.0113	Pass
Approximate entropy test	0.6512	Pass	0.8727	Pass
Cumulative sums test (Forward)	0.9992	Pass	1	Pass
(Backward)	1.0000	Pass	0.7967	Pass
Random excursions test (x=-4)	0.8077	Pass	0.9132	Pass
Random excursions test (x=-3)	0.4781	Pass	0.6853	Pass
Random excursions test (x=-2)	0.3439	Pass	0.7431	Pass
Random excursions test (x=-1)	0.0614	Pass	0.7231	Pass
Random excursions test (x=1)	0.5443	Pass	0.9737	Pass
Random excursions test (x=2)	0.5234	Pass	0.6656	Pass
Random excursions test (x=3)	0.0764	Pass	0.4223	Pass
Random excursions test (x=4)	0.3546	Pass	0.7851	Pass
Random excursions variant test (x=-9)	0.3358	Pass	0.7863	Pass
Random excursions variant test (x=-8)	0.2515	Pass	0.6650	Pass
Random excursions variant test (x=-7)	0.1730	Pass	0.6198	Pass
Random excursions variant test (x=-6)	0.2315	Pass	0.9462	Pass
Random excursions variant test (x=-5)	0.6366	Pass	0.7942	Pass
Random excursions variant test (x=-4)	0.7479	Pass	0.6420	Pass
Random excursions variant test (x=-3)	0.4725	Pass	0.5157	Pass
Random excursions variant test (x=-2)	0.7025	Pass	0.4386	Pass
Random excursions variant test (x=-1)	0.9247	Pass	0.3143	Pass
Random excursions variant test (x=1)	0.5708	Pass	0.8231	Pass
Random excursions variant test (x=2)	0.4450	Pass	0.6514	Pass
Random excursions variant test (x=3)	0.4220	Pass	1	Pass
Random excursions variant test (x=4)	0.4320	Pass	0.5828	Pass
Random excursions variant test (x=5)	0.7290	Pass	0.6280	Pass
Random excursions variant test (x=6)	0.9773	Pass	0.5666	Pass
Random excursions variant test (x=7)	0.9791	Pass	0.5558	Pass
Random excursions variant test (x=8)	0.7327	Pass	0.5444	Pass
Random excursions variant test (x=9)	0.6141	Pass	0.4641	Pass

The Caputo fractional differential definition is most commonly expressed in the following form:

$D^q u(t) = J^{s-q} u^{(s)}(t) (q > 0)$ , where  $s$  is an integer which is not less than  $q$ ,  $u^{(s)}$  is the  $s$ -order derivative, and  $J^q$  is the  $q$ -order Reimann–Liouville integral operator in the following form:

$$J^q v(t) = \frac{\int_0^t (t-\tau)^{q-1} v(\tau) d\tau}{\Gamma(q)},$$

where  $\Gamma$  is the Gamma function, and  $D^q$  represents a  $\theta$ -order Caputo differential operator.

Based on Definition 1, the CFLCS [22] can be expressed as follows:

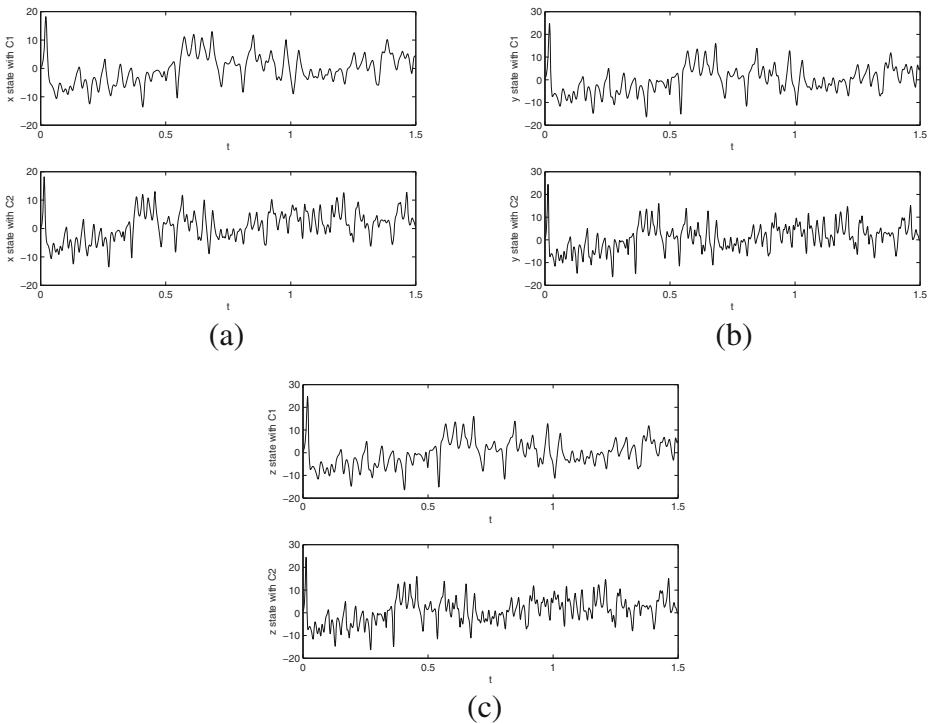
$$D^q \mathbf{w}_i(t) = (1-\varepsilon)f(\mathbf{w}_i(t)) + \frac{1}{2}\varepsilon f(\mathbf{w}_{i+1}(t)) + \frac{1}{2}\varepsilon f(\mathbf{w}_{i-1}(t)) \tag{2}$$

where  $\varepsilon$  is the coupled intensity,  $\mathbf{w}$  is the state vector,  $i(i = 1, 2, \dots, n)$  is the lattice index, and  $n$  is the lattice length. As a whole, system (2) is periodically connected. In detail, local system  $f$  is set as the following Lorenz system form

$$\begin{cases} \dot{x} = a(y-x) \\ \dot{y} = bx - y - xz \\ \dot{z} = xy - cz \end{cases} \tag{3}$$

where  $x$ ,  $y$ , and  $z$  are state variables of the Lorenz system, and the system parameters are  $a = 10$ ,  $b = 28$ , and  $c = \frac{8}{3}$ . With randomly generated initials, by applying the coarse-grained method to the chaotic signals from system (2), the derived key signals for different coarse-grained parameters  $C_1 = 2$  and  $C_2 = 3$  are shown in Fig. 2.

With the coarse-grained keys generated using the above logistic map and the CFLCS signals, the digital images can be encrypted with proper scrambling and diffusion algorithms. It should be noted that, because fractional system (2) can generate three different coarse-grained keys simultaneously, as shown in Fig. 3, the keys can be applied in the encryption of



**Fig. 2** The key signals generated with coarse-grained parameters  $C_1$  and  $C_2$  from different chaotic state signals in parameters  $C_1$  and  $C_2$ : **a** The first system states generated keys, **b** The second system states generated keys, **c** The third system states generated keys

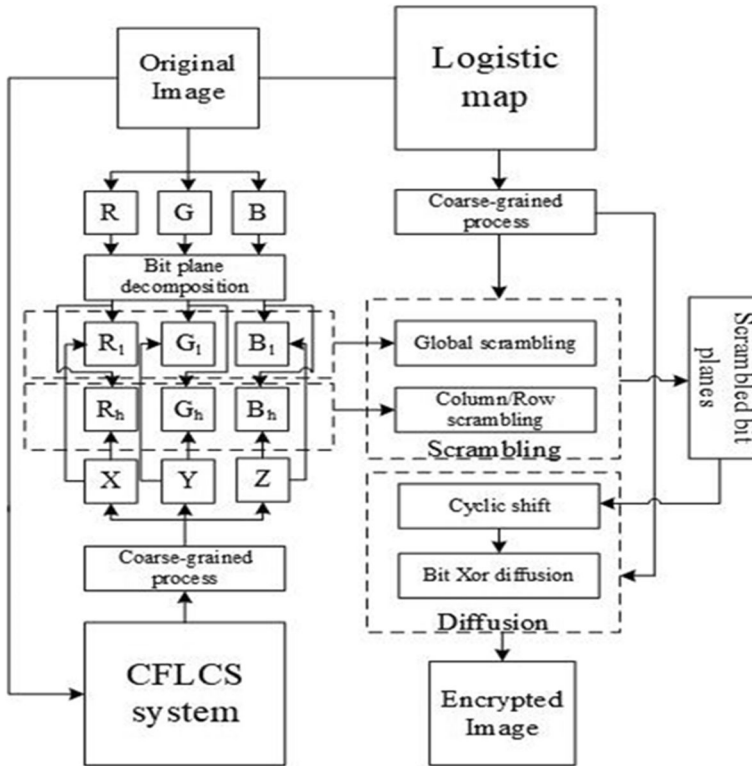


Fig. 3 The global flow diagram for color image encryption with coarse-grained keys

R, G, and B channels in parallel for color images. The entire encryption can be applied as indicated in Fig. 3. For the proposed method, images can be recovered using a symmetrical decryption method and a detailed encryption analysis, as described in the following subsections.

### 2.1 Scrambling analysis

In this section, the pixels of a color image in different channels are translated into different bit streams, and these bits are scrambled using different methods. The detailed steps are provided as follows:

- Step 1. For a given color image  $F$  of  $height \times width$ , generate  $R$ ,  $G$ , and  $B$  vectors  $F_r$ ,  $F_g$ , and  $F_b$ , respectively. For each  $F_r$ ,  $F_g$ , or  $F_b$  vector, the following steps are applied.
- Step 2. Extract the high bit planes in the corresponding channel and reconstruct the bit matrix. For the fifth through the eighth bits, the sizes of the extracted bit planes  $BP_i (i = 5, 6, 7, 8)$  are all  $height \times width$ . Combine these bit planes as vector  $H = [BP_5 \ BP_6 \ BP_7 \ BP_8]_{height \times 4width}$
- Step 3. With the above coarse-grained methods and CFLCS (2), generate encryption keys  $k_1$  and  $k_2$ . The lengths of  $k_1$  and  $k_2$  are  $height$  and  $4width$ , respectively. Using the MATLAB function  $sort(\bullet)$ , obtain index vectors  $Index_1$  and  $Index_2$  from  $k_1$  and  $k_2$ , respectively.

- Step 4. From  $i = 1$  to  $i = \text{height}$ , apply the loop operation  $Q(:, i) = P(:, \text{index}_2(i))$ . From  $i = 1$  to  $i = 4 \times \text{width}$ , apply the loop operation  $Q(:, i) = P(:, \text{index}_2(i))$ . Then, the high bit planes are obtained as a row vector  $H = \text{reshape}(Q, 1, 4 \times \text{height} \times \text{width})$ .
- Step 5. Extract the low bit planes in the corresponding channel and reconstruct the bit matrix. For the first through the fourth bits,  $BP_i (i = 1, 2, 3, 4)$  can be obtained and reshaped as  $BP_i = \text{reshape}(BP_i, 1, \text{height} \times \text{width})$ . Then, the low bit planes can be expressed as  $L = [BP_1 \ BP_2 \ BP_3 \ BP_4]_{1 \times (4 \times \text{height} \times \text{width})}$ .
- Step 6. With the above coarse-grained methods and logistic map (1), generate a  $4 \times \text{height} \times \text{width}$  encryption key  $k$ , and sort  $k$  to obtain the corresponding vector  $\text{Index}$ . From  $i = 1$  to  $i = 4 \times \text{height} \times \text{width}$ , conduct the loop operation  $O(1, i) = L(1, \text{index}(i))$ . Then, reset  $L = O$ , and complete the scrambling process.

It can be observed that both the CFLCS and the logistic map chaotic signals are utilized in the scrambling. In addition, high bits and low bits are scrambled using different methods according to the importance of different bits to make the scrambling robust. For the CFLCS, three different keys based on coarse-grained chaotic signals in different states can be applied in the scrambling of the R, G, and B channels simultaneously, thereby making the encryption more efficient.

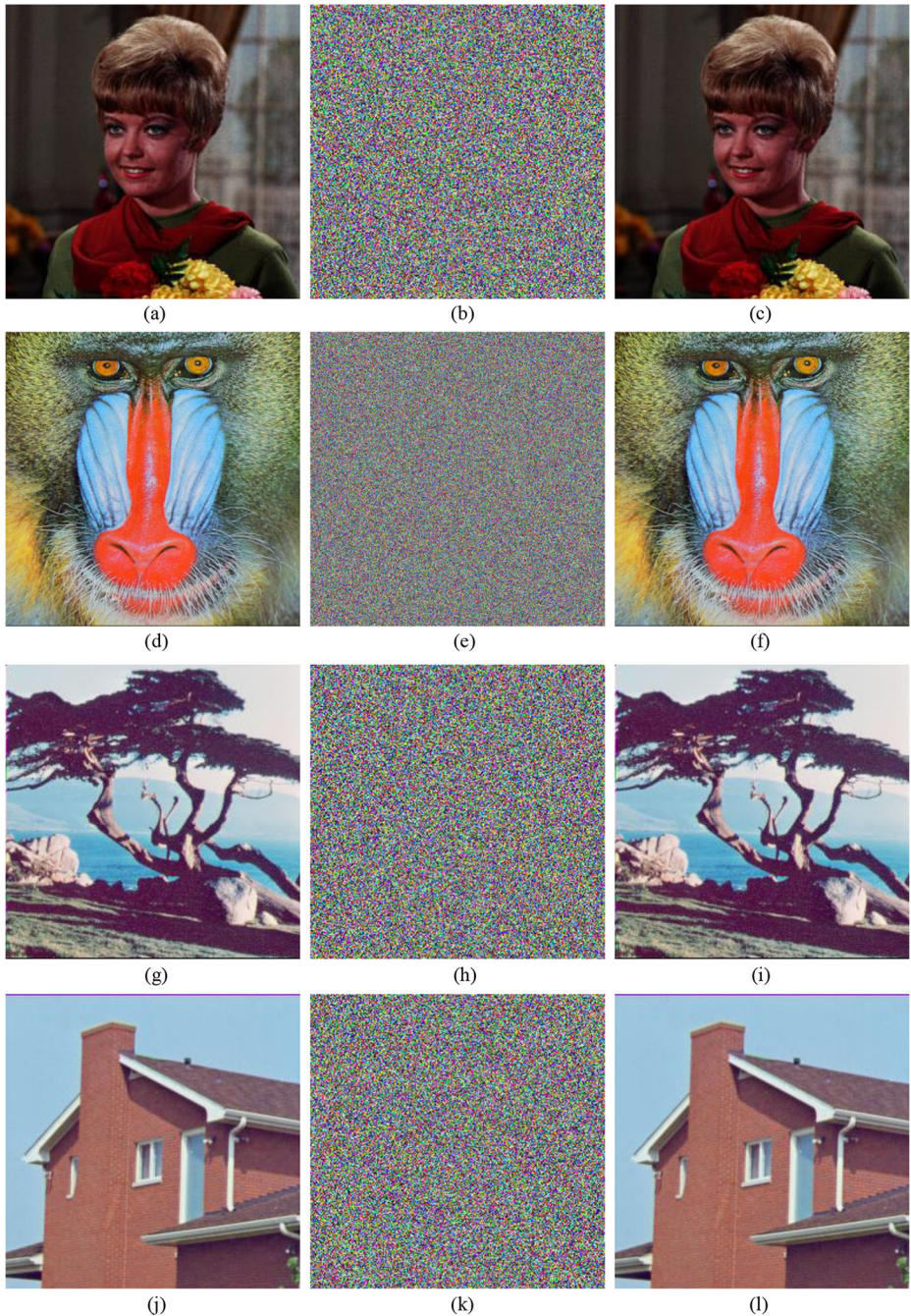
## 2.2 Diffusion analysis

In any R, G, or B channel, for bit vectors  $H$  and  $L$  after scrambling, the following diffusion operations should be applied:

- Step 1. With a logistic map (1), generate chaotic signals, and apply a coarse-grained process with parameter  $C$  to obtain diffusion key  $T$ . Reset  $T_s = \text{mod}(\text{floor}(T \times 10^{14}), 256)$ , and extract the bit vectors  $BC_i (i = 1, 2, \dots, 8)$  from  $T_s$ . Then, reconstruct vectors  $B_1$  and  $B_2$  as  $B_1 = [BC_1 \ BC_3 \ BC_5 \ BC_7]_{1 \times (4 \times \text{height} \times \text{width})}$  and  $B_2 = [BC_2 \ BC_4 \ BC_6 \ BC_8]_{1 \times (4 \times \text{height} \times \text{width})}$ , respectively.
- Step 2. Based on the calculation, obtain the summation of all bits in  $BP_i (i = 5, 6, 7, 8)$  as  $\text{sum}_1$ . Next, obtain the summation of all bits in  $BP_i (i = 1, 2, 3, 4)$  as  $\text{sum}_2$ . Apply a cyclic shift operation as  $L = \text{circshift}(L^T, C \times \text{sum}_1 + \text{sum}_2)^T$ .
- Step 3. Move on to the diffusion with the XOR operation as follows:

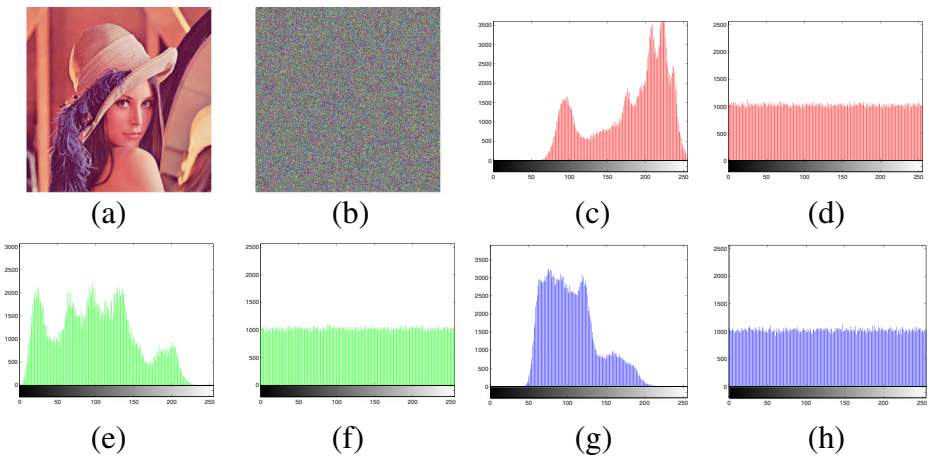
$$\begin{cases} L(1) = L(1) \oplus H(1) \oplus L(4 \times \text{height} \times \text{width}) \oplus B_1(1) \\ L(i) = L(i) \oplus H(i) \oplus L(i-1) \oplus B_1(i), i = 2, 3, \dots, 4 \times \text{height} \times \text{width}, \end{cases} \quad (4)$$

- Step 4. Analogously, for high bits, apply a cyclic shift operation as  $H = \text{circshift}(H^T, C \times \text{sum}_2 + \text{sum}_1)^T$ . Then do the XOR operation as:



**Fig. 4** Encryption and decryption results: **a** Original image of Girl, **b** Encrypted image of Girl, **c** Decrypted image of Girl, **d** Original image of Baboon, **e** Encrypted image of Baboon, **f** Decrypted image of Baboon, **g** Original image of Tree, **h** Encrypted image of Tree, **i** Decrypted image of Tree, **j** Original image of House, **k** Encrypted image of House, **l** Decrypted image of House





**Fig. 5** Histogram of original images and encrypted Lena images: **a** Original image of Lena, **b** Encrypted image of Lena, **c** Red component of (a), **d** Red component of (b), **e** Green component of (a), **f** Green component of (b), **g** Blue component of (a), **h** Blue component of (b)

$$\begin{cases} H(1) = L(1) \oplus H(1) \oplus H(4 \times height \times width) \oplus B_2(1) \\ H(i) = L(i) \oplus H(i) \oplus H(i-1) \oplus B_2(i), i = 2, 3, \dots, 4 \times height \times width, \end{cases} \quad (5)$$

Step 5. Recombine the color image with the derived bit streams in the R, G, and B channels after the diffusion. The encryption is then complete.

Similar to the scrambling, the diffusion of high bit streams and low bit streams are different. In addition, the proposed encryption is a symmetrical encryption method, and the encrypted image can be recovered with a completely reversed decryption process.

### 3 Performance analysis

In this section, simulations to analyze the encryption algorithm are conducted using MATLAB R2014a on identical Thinkpad L470 laptops running Microsoft Windows 10, with a 2.70 or

**Table 2** Information entropies of encrypted images

Image	R component	G component	B component
Lena	7.9975	7.9972	7.9977
Girl	7.9973	7.9973	7.9974
Peppers	7.9993	7.9994	7.9994
Tree	7.9968	7.9975	7.9972
Airplane	7.9993	7.9993	7.9993

**Table 3** Information entropy comparisons of “Lena image” with some Ref.

Methods	R component	G component	B component
Our algorithm	7.9975	7.9972	7.9977
Zhang [34]	7.9966	7.9966	7.9968
Suryanto [19]	7.7325	7.7339	7.7264
Parvaz [17]	7.9975	7.9970	7.9970

2.90 GHz Intel<sup>(R)</sup> Core<sup>(TM)</sup> i7-7500U CPU and 16.0 GB of RAM. The color images included “Girl.tiff,” “Tree.tiff,” “House.tiff,” and “Baboon.tiff.” and these images are from test collections USC-SIPI. Using the aforementioned methods, the original, encrypted, and decrypted images are shown in Fig. 4.

Figures 4a–c present a Girl image, an encrypted Girl image, and a decrypted Girl image, respectively. Figures 4d–f depict a Baboon image, an encrypted Baboon image, and a decrypted Baboon image, respectively. Figures 4g–i show a Tree image, an encrypted Tree image, and a decrypted Tree image, respectively. Figures 4j–l present a House image, an encrypted House image, and a decrypted House image, respectively. It can be observed that, for different color images, any information in the encrypted images is difficult to identify, and the encrypted image can be successfully recovered.

### 3.1 Histogram analysis

In this section, histograms of the original image and an encrypted image are described. It should first be noted that a histogram is used to display the distribution pixel information of an image, and an ideal algorithm can generate an encrypted image with uniformly distributed histograms. If the histograms are sufficiently flat, a statistical attack will be useless in the pixel analysis. Taking the Lena image as an example, the distributions of all 256 pixels are as shown in Fig. 5.

Histograms of the encrypted images are presented in Fig. 5d, f, h. Compared with the histograms in Fig. 5c, e, g before encryption, the histograms after encryption are flat and can effectively resist a statistical attack.

### 3.2 Key space analysis

For a robust algorithm, larger key space means the algorithm is more secure because one cannot find out the correct keys from the large key space. From the global encryption process described in sections 2.1 and 2.2, it can be observed that three discrete logistic chaotic series

**Table 4** NPCR and UACI for encrypted color images

Image	NPCR(%)			UACI(%)		
	R component	G component	B component	R component	G component	B component
Lena	99.5590	99.5895	99.6063	33.5696	33.4967	33.5644
Girl	99.6109	99.6170	99.6750	33.4030	33.5253	33.4748
Peppers	99.6273	99.6010	99.5945	33.4462	33.4547	33.3990
Tree	99.6124	99.6033	99.6201	33.4277	33.5333	33.6586
Airplane	99.6231	99.6284	99.6098	33.4997	33.4394	33.4807

**Table 5** NPCR and UACI comparisons of “Lena image” with some Refs

Methods	NPCR(%)			UACI(%)		
	R component	G component	B component	R component	G component	B component
Our method	99.5590	99.5895	99.6063	33.5696	33.4967	33.5644
Zhang [34]	99.6170	99.6231	99.6704	33.6606	33.4486	33.4149
Suryanto [19]	99.45	99.46	99.49	26.12	26.24	26.12
Parvaz [17]	99.6078	99.6140	99.6033	33.4457	33.5589	33.5243

and a fractional chaotic series will be generated. Let us suppose that the computational accuracy is  $10^{-15}$ , the key space of the discrete initial values for logistic chaos is  $10^{15 \times 3}$ , and the key space of the parameter for logistic chaos is  $(4 - 3.57) \times 10^{15 \times 3}$ . Assuming that the coupling number of the lattices is  $n = 10$ , the fractional order is  $0 < q \leq 1$ , and the coupling strength is  $0 < \varepsilon < 1$ . Then, the key space of the parameter for CFLCS chaos is  $10^{15 \times 2}$ , and the key space of the discrete initial values for logistic chaos is  $10^{15 \times 10}$ . The entire key space is at least  $(4 - 3.57) \times 10^{270}$ , and the key space is sufficiently large to resist the force attack effectively. In addition, for the coarse-grained chaotic methods, different coarse-grained

**Table 6** Correlations of the original image (OI) and corresponding encrypted image (EI)

Image	Component	Correlation coefficient	Direction to calculation correlation		
			Horizontal	Vertical	Diagonal
Lena	Red	OI	0.9647	0.9639	0.9143
		EI	0.0091	-0.0123	0.0258
	Green	OI	0.9730	0.9407	0.9186
		EI	-0.0012	0.0047	0.0188
	Blue	OI	0.9484	0.8907	0.8434
		EI	-0.0223	-0.0057	-0.0142
Girl	Red	OI	0.9645	0.9728	0.9457
		EI	-0.0016	0.0237	-0.0561
	Green	OI	0.9593	0.9702	0.9517
		EI	-0.0042	0.0031	0.0271
	Blue	OI	0.9457	0.9573	0.9426
		EI	-0.0055	-0.0189	-0.0048
Tree	Red	OI	0.9271	0.9551	0.9100
		EI	-0.0219	-0.0003	0.0225
	Green	OI	0.9461	0.9696	0.9353
		EI	0.0230	0.0003	-0.0240
	Blue	OI	0.9332	0.9635	0.9273
		EI	0.0233	-0.0167	0.0264
Airplane	Red	OI	0.9583	0.9721	0.9180
		EI	-0.0114	0.0038	-0.0188
	Green	OI	0.9748	0.09590	0.9457
		EI	-0.0212	0.0129	0.0123
	Blue	OI	0.9453	0.9615	0.9217
		EI	0.0218	-0.0381	0.0387
Peppers	Red	OI	0.9642	0.9567	0.9560
		EI	-0.0350	-0.0084	0.0419
	Green	OI	0.9849	0.9787	0.9759
		EI	-0.0191	0.0358	-0.0071
	Blue	OI	0.9636	0.9679	0.9528
		EI	-0.0320	-0.0183	0.0448

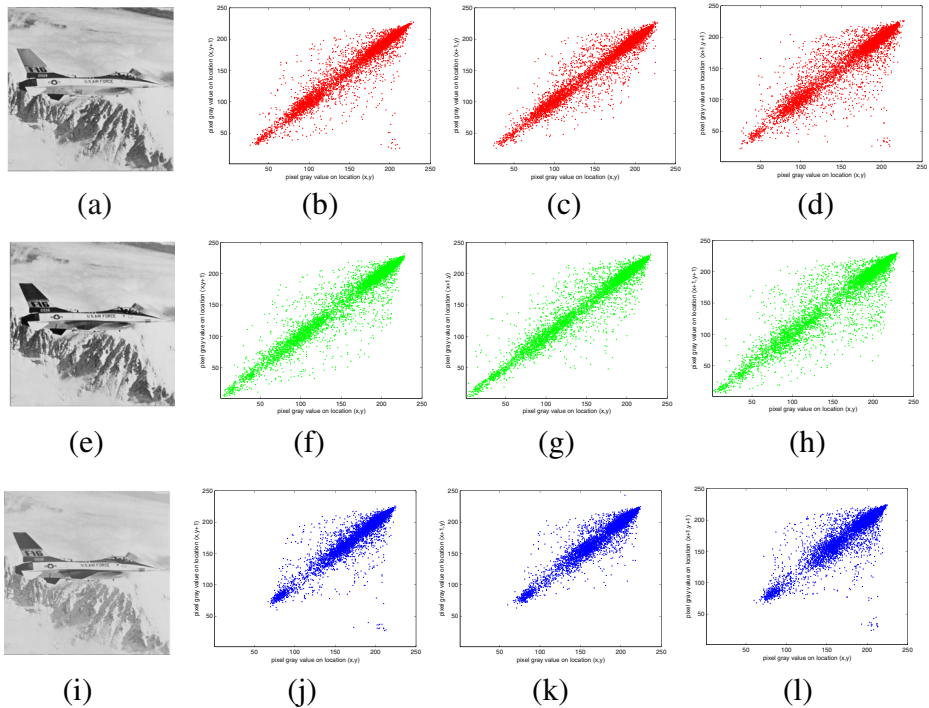
parameters lead to different chaotic signals (from Fig. 1). Thus, the key signals are can vary based on the strength of the security.

### 3.3 Information entropy analysis

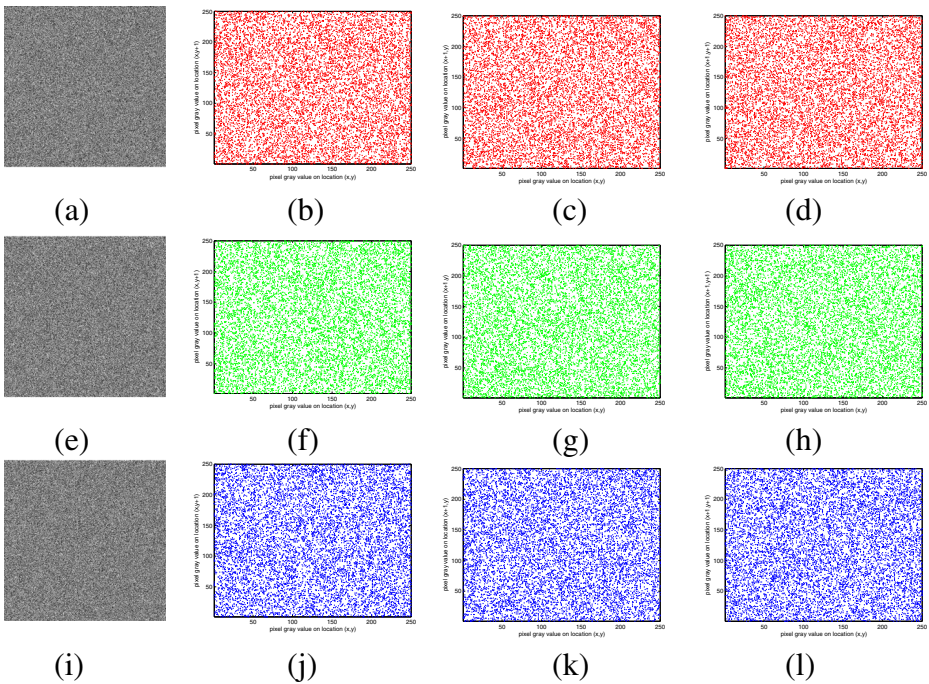
Information entropy is one of the most important properties in the analysis of the encryption performance. By defining the information source as  $m$ , the information entropy formula can be depicted as follows:

$$H(m) = \sum_{i=0}^{L-1} p(m_i) \log \frac{1}{p(m_i)} \tag{6}$$

where  $L$  is the total number of symbols, and function  $p(\bullet)$  is the occurrence probability of the corresponding symbol. In general, the information entropy is close to the ideal value of 8 if the encryption algorithm is good. In this paper, the information entropies of the proposed algorithm for different color images are listed in Table 2, and comparisons of the proposed algorithm with some of the latest effective encryption algorithms are listed in Table 3. Compared with previous studies, our entropy schemes are slightly larger; thus, when our scheme is used, the encrypted images can effectively resist possible statistical analysis attacks.



**Fig. 6** Adjacent pixel correlation of the original Airplane image: **a** R component, **b** Horizontal of (a), **c** Vertical of (a), **d** Diagonal of (a), **e** G component, **f** Horizontal of (e), **g** Vertical of (e), **h** Diagonal of (e), **i** B component, **j** Horizontal of (i), **k** Vertical of (i), **l** Diagonal of (i)



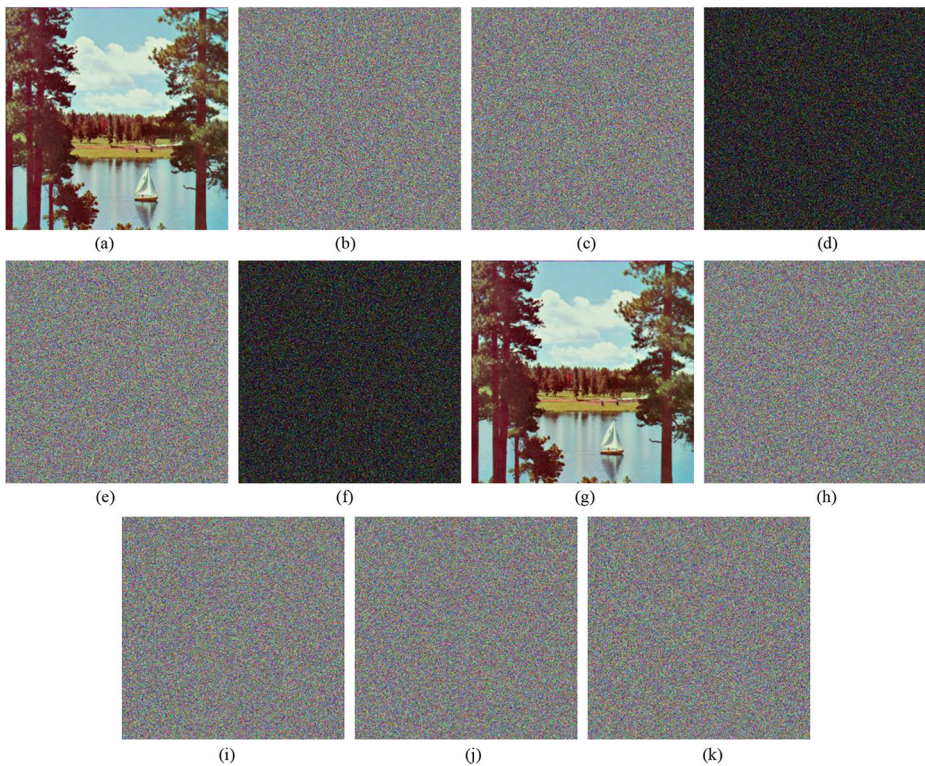
**Fig. 7** Adjacent pixel correlation of the encrypted Airplane image: **a** R component, **b** Horizontal of (a), **c** Vertical of (a), **d** Diagonal of (a), **e** G component, **f** Horizontal of (e), **g** Vertical of (e), **h** Diagonal of (e), **i** B component, **j** Horizontal of (i), **k** Vertical of (i), **l** Diagonal of (i)

### 3.4 Differential attack analysis

To resist the differential attack, two criteria, namely the number of pixels change rate (*NPCR*) and the unified average changing intensity (*UACI*), are utilized to analyze the effectiveness. *NPCR* and *UACI* are based on a test of two encrypted images generated from the original images with just one bit being slightly different. The ideal values for *NPCR* and *UACI* are 99.6094% and 33.4635%, respectively. The *NPCR* and *UACI* of our proposed method for different color images are listed in Table 4, and comparisons of the *NPCR* and *UACI* obtained using our method with those obtained using other methods are listed in Table 5. Compared with previous studies, the values of *NPCR* and *UACI* obtained using the proposed method are closer to the ideal values.

### 3.5 Correlation of adjacent pixels

To test resistance to a statistical attack further, the correlation is selected to test the relationship between two adjacent series. In general, the correlations in the original image are always close to 1, but in the encrypted images, they are close to zero. In the following image analysis, correlations between two adjacent pixels horizontally, vertically, and diagonally for different color images are presented in Table 5, and comparisons of these values with those obtained using other methods are shown in Table 6. The related correlation equations for two pixel series  $x = \{x_i\}$  and  $y = \{y_i\}$  are defined as follows:



**Fig. 8** Key sensitive analysis of the color Lake image: **a** The original image of the Lake, **b** Encrypted image of (a), **c** Encrypted image of (a) with modified  $u$ , **d** Difference of (b) and (c), **e** Encrypted image of (a) with modified  $C$  **f** Difference of (b) and (e), **g** Decrypted image of (a) with  $u$  and  $C$ , **h** Decrypted image of (c), **i** Decrypted image of (e), **j** Decrypted image of (a) with modified  $u$  **k** Decrypted image of (a) with modified  $C$

$$\begin{cases} r_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \\ \text{cov}(x,y) = \frac{1}{N} \sum_{i=1}^N [x_i - E(x)][y_i - E(y)] \\ D(x) = \frac{1}{N} \sum_{i=1}^N [x_i - E(x)]^2 \\ E(x) = \frac{1}{N} \sum_{i=1}^N x_i \end{cases} \quad (7)$$

In addition, by taking the airplane image as an example, the horizontal, vertical, and diagonal correlation scatter diagrams of the original and encrypted images are as shown in Figs. 6 and 7, respectively. Figure 6a, e, and i are three components of the airplane image. Figure 6b, c, and d show the horizontal, vertical, and diagonal correlation values in the red channel. Figure 6f, g, and h depict the horizontal, vertical, and diagonal correlation values in the green channel. Analogously, the horizontal, vertical, and diagonal correlation values in the

blue channel are presented in Fig. 6j, k, and l. It can be observed that the correlation values in the original image are extremely high. By contrast, these correlation values in the encrypted components in Fig. 7b-d, f-h, and j-l are low, and adjacent pixels cover the corresponding whole plane. In addition, from Table 6 and Figs. 6 and 7, it can be observed that the proposed method is effective at not only encrypting different types of images to resist a statistical attack but also breaking the relationships between adjacent pixels.

### 3.6 Key sensitivity analysis

A key sensitivity analysis is provided in this section. According to the image encryption technology, a good encryption algorithm will lead to an unfeasible decryption from even a minuscule key change. In the following analysis, the sensitivity of the keys for logistic map parameter  $u$  and coarse-grained parameter  $C$  will be provided. Set a minuscule change of  $\delta = 10^{-15}$  for key  $u$ , and add a change of 1 for key  $C$ . Taking the Lake image as an example, the derived encryption and decryption images are as shown in Fig. 8.

Figure 8a shows the original Lake image, and Fig. 8b is an encrypted image with correct keys  $u$  and  $C$ . With the change in parameter  $u$ , the image in Fig. 8a is encrypted as shown in Fig. 8c, and the differences between Fig. 8b and c can be seen in Fig. 8d. It can be concluded that the tiny change in  $u$  will result in completely different encrypted images. Analogously, the same conclusion for key  $C$  can be derived in Fig. 8e and f. With correct keys, the image in Fig. 8b can be recovered as in Fig. 8g. However, when the correct keys and algorithm are used, the images in Fig. 8c and e will not be decrypted, as shown in Fig. 8h and i. In addition, the changed keys will not decrypt the image in Fig. 8b, as shown in Fig. 8j and k. From Fig. 8, it can be observed that the images will be recovered using only the exactly correct parameters, thereby demonstrating that our proposed algorithm is a robust algorithm with good key sensitivity.

In order to further test the key sensitivity and do contrast test with other algorithms, the MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) analysis are done in the Lena image as following Table. 7.

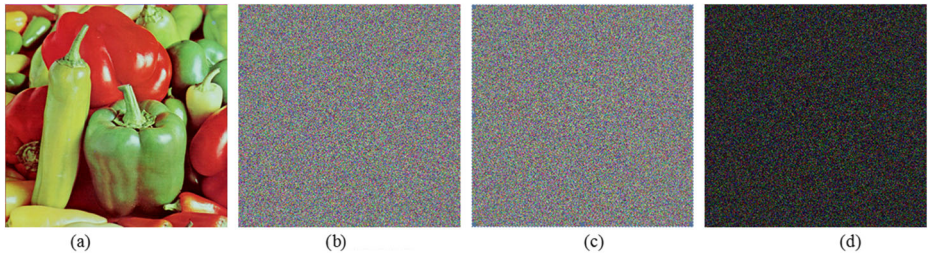
From Table 7, it can be found that MSE values are all very large and PSNR values are all smaller than 10 in different components. Compared with listed references, the proposed algorithm has larger MSE values and smaller PSNR values. It means the algorithm is very sensitive to keys and it is robust to resist to possible key attacks.

### 3.7 Plain-text sensitivity analysis

Similar to a key sensitivity analysis, the plain-text sensitivity analysis is another important index for testing the encryption algorithm. This means that a good algorithm should also be

**Table 7** MSE and PSNR of Lena image

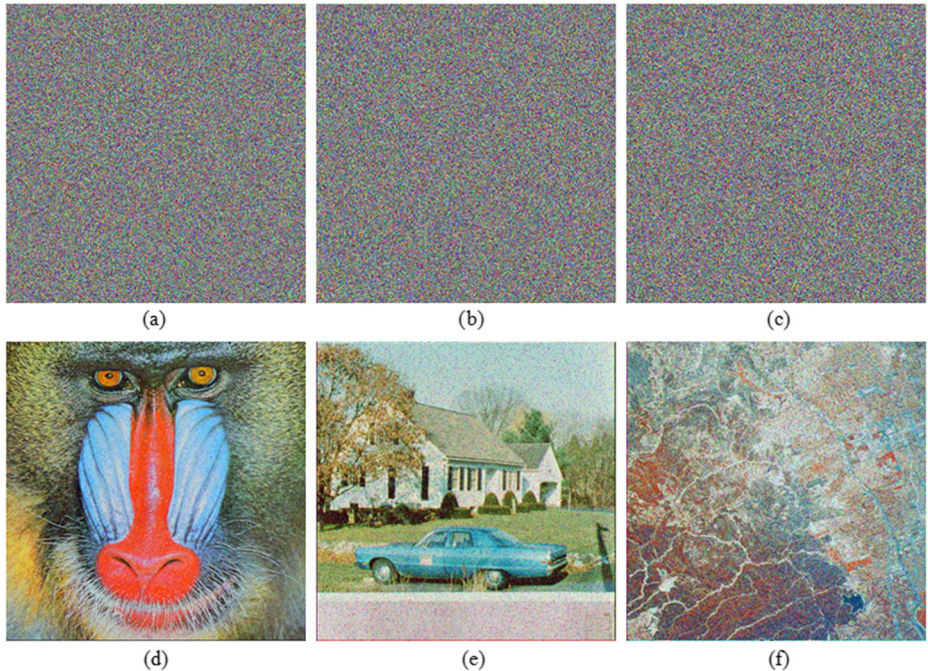
Analysis Test	Component	Our algorithm	[10]	[2]
MSE	R	10,561.7	10,464.9	7385.1
	G	9032.2	8800.42	8698.7
	B	7100.7	7119.13	9580.0
PSNR	R	7.8935	7.93344	9.1080
	G	8.5729	8.68577	8.7362
	B	9.6178	9.60653	8.3167



**Fig. 9** Plain-text sensitive analysis of the color Peppers image: **a** The original image of the Peppers, **b** Encrypted image of (a), **c** Encrypted image of (a) with one-bit change, **d** Difference of (b) and (c)

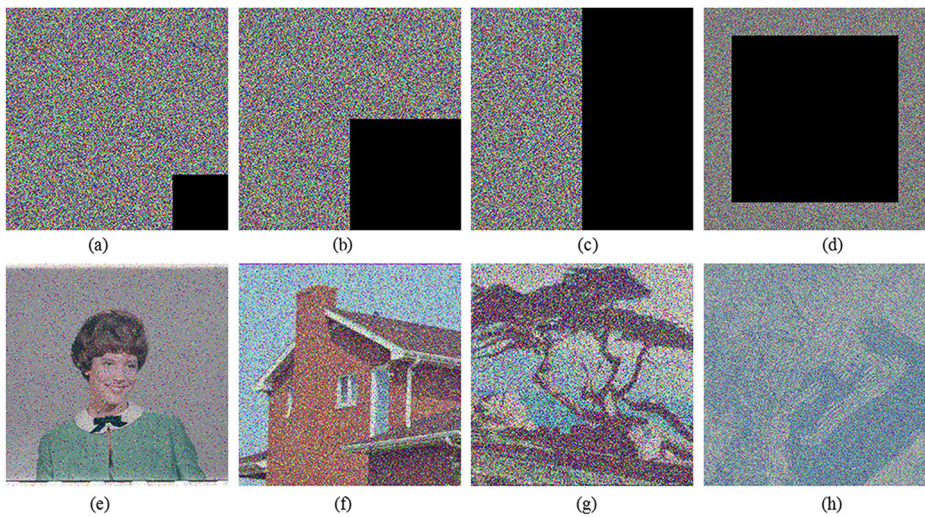
sensitive to tiny changes in the plain-text information. Taking the Peppers image as an example, by changing the pixel at position (10,10) by only 1 bit, the derived results are as shown in Fig. 9.

Figure 9a is the original image of the Peppers and Fig. 9b is the encrypted image with no change in the plain-text. With the tiny changed plain-text, the image Fig. 9a is encrypted as Fig. 9c and the difference between Fig. 9b and c can be seen in Fig. 9d. It can be concluded that our proposed algorithm is a robust algorithm with good plain-text sensitivity.



**Fig. 10** Encryption and decryption with salt-and-pepper noise: **a** Encrypted image with noise intensity 0.02, **b** Encrypted image with noise intensity 0.05, **c** Encrypted image with noise intensity 0.1, **d** Decrypted image of (a), **e** Decrypted image of (b), **f** Decrypted image of (c)





**Fig. 11** Encryption and decryption with cropping attack: images with following cropping proportions **a** 6.25%, **b** 25%, **c** 50%, **d** 56.25%. **e** Decrypted image of (a), **f** Decrypted image of (b), **g** Blue component of (c), **h** Decrypted image of (d)

### 3.8 Resistance to salt-and-pepper noise

There are different types of noise in practice, among which salt-and-pepper noise is a typical example and has a strong influence on the encrypted image. To test the resistance, three salt-and-pepper noises with different intensities were considered in color Baboon, Car, and Map images. The encrypted images with salt-and-pepper noise and the recovered images are shown in Fig. 10.

Figure 10a shows an encrypted Baboon image with a noise intensity of 0.02. Figure 10b presents an encrypted Car image with a noise intensity of 0.05. Figure 10c shows an encrypted Map image with a noise intensity of 0.1. Figure 10d–f present the corresponding decrypted images. It can be observed that the recovered images are all recognizable under different noise intensities, thereby indicating that the proposed method is robust to noise attacks.

### 3.9 Resistance to cropping attacks

The information in an image may not only be influenced by noise but also become lost during the storage procedure. To test resistance to information loss, we added cropping attacks to the encrypted images of Female, House, Tree, and Map2. In addition, different information loss percentages are considered. The experiments are shown in Fig. 11.

Figure 11a shows an encrypted Female image with 6.25% information loss. Figure 11b shows an encrypted House image with 25% information loss. Figure 11c shows an encrypted Tree image with 50% information loss. Figure 11d shows an encrypted Map2 image with 56.25% information loss. Finally, Fig. 11e–h present the corresponding decrypted images. It can be observed that the recovered images are all recognizable under the different information loss percentages, thereby indicating that the proposed method is robust to cropping attacks.

## 4 Conclusion

In this paper, a color image encryption algorithm based on coarse-grained chaotic keys is proposed. The coarse-grained method enables the derived chaotic series to be unfixed even with same initial value and chaotic parameter. In addition, both a discrete logistic map and a fractional chaotic system are utilized to generate chaotic signals for the coarse-grained process. During the encryption process, parallel bit-level scrambling, including a high bit row, column scrambling, and low bit global scrambling, are operated. In addition, a further cyclic shift and XOR diffusion are applied to the diffused bit streams. During the simulation, the key space, information entropy, correlation analysis, key sensitivity, plain-text sensitivity, and resistance to differential, noise, and cropping attacks are presented, thereby demonstrating the advantages of the proposed approach. Both theoretical analysis and simulation results indicate that the coarse-grained fractional chaotic encryption algorithm is effective and robust to various types of attacks.

**Acknowledgements** This research is supported by the National Natural Science Foundation of China (Nos: 61702356, 61672124, 61503375 and 61701070), the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund (No: MMJJ20170203).

## References

1. Antonini M, Barlaud M, Mathieu P, Daubechies I (1992) Image coding using wavelet transform. *IEEE Trans Image Process* 1(2):205–220
2. Arshad U, Khan M, Shaukat S, Amin M, Shah T (2020) An efficient image privacy scheme based on nonlinear chaotic system and linear canonical transformation. *Phys A: Stat Mech Appl* 546:123458
3. Caputo M (1967) Linear models of dissipation whose  $q$  is almost frequency independent-2. *Geophys J R Astron Soc* 13(5):529–539
4. Chang YF (2013) Chaos, fractal in biology, biothermodynamics and matrix representation on Hypercycle. *Neuroquantology* 11(4):527–536
5. Chen CS, Wang T, Kou YZ, Chen XC, Li X (2013) Improvement of trace-driven I-cache timing attack on the RSA algorithm. *J Syst Softw* 86(1):100–107
6. Coppersmith D (1994) The data encryption standard (DES) and its strength against attacks. *IBM J Res Dev* 38(3):243–250
7. Cramer JA, Booksh KS (2006) Chaos theory in chemistry and chemometrics: a review. *J Chemom* 20(11): 447–454
8. Diaconu AV, Ionescu V, Iana G, Lopez-Guede JM (2016) A new bit-level permutation image encryption algorithm. *Int Conf Commun* 411–416
9. Faragallah OS, Afifififi A (2017) Optical color image cryptosystem using chaotic baker mapping based-double random phase encoding. *Opt Quant Electron* 49:89
10. Haq TU, Shah T (2020)  $12 \times 12$  S-box design and its application to RGB image encryption. *Optik* 217: 164922
11. Huang XL (2012) Image encryption algorithm using chaotic Chebyshev generator. *Nonlinear Dyn* 67(4): 2411–2417
12. Huang X (2020) Nia M P, ding Q, research on image encryption based on Hyperchaotic system. *J Netw Intell* 5(1):10–22
13. Hussain I, Shah T (2013) Application of S-box and chaotic map for image encryption. *Math Comput Model* 57(9–10):2576–2579
14. Kawashima M (1985) Terminal care (1): Chaos brought about by the progress of modern medical technology. *Kangogaku Zasshi* 49(10):1092–1095
15. Kol'tsov NI, Fedotov VK (2018) Two-Dimensional Chaos in chemical reactions. *Russ J Phys Chem B* 12(3):590–592
16. Li JF, Xiang SY, Wang HN, Gong JK, Wen AJ (2018) A novel image encryption algorithm based on synchronized random bit generated in cascade-coupled chaotic semiconductor ring lasers. *Opt Lasers Eng* 102:170–180

17. Parvaz R, Zarebnia M (2018) A combination chaotic system and application in color image encryption. *Opt Laser Technol* 101:30–41
18. Pasquini C, Boato G, Bohme R (2019) Teaching digital signal processing with a challenge on image forensics. *IEEE Signal Process Mag* 36(2):101–109
19. Suryanto Y, Ramli K (2017) A new image encryption using color scrambling based on chaotic permutation multiple circular shrinking and expanding. *Multimed Tools Appl* 76:16831–16854
20. Teng L, Wang XY, Meng J (2018) A chaotic color image encryption using integrated bit-level permutation. *Multimed Tools Appl* 77(6):6883–6896
21. Trahanias PE, Venetsanopoulos AN (1993) Vector directional filters—a new class of multichannel image processing filters. *IEEE Trans Image Process* 2(4):528–534
22. Wang XY, Zhang H (2012) Chaotic synchronization of fractional-order spatiotemporal coupled Lorenz system. *Int J Mod Phys C* 23(10):1250067
23. Wang XT, Zhu HF (2019) A novel two-party key agreement protocol with the environment of wearable device using chaotic maps. *Data Sci Pattern Recognit* 3(2):12–23
24. Wang X, Zhao J, Liu H (2012) A new image encryption algorithm based on chaos. *Opt Commun* 285(5): 562–566
25. Wang ZF, Dong JJ, Zhen JQ, Zhu FZ (2019) Template protection based on chaotic map and DNA encoding for multimodal biometrics at feature level fusion. *J Inf Hiding Multimed Signal Process* 10(1):1–10
26. Wang XY, Zhao HY, Wang MX (2019) A new image encryption algorithm with nonlinear-diffusion based on multiple coupled map lattices. *Opt Laser Technol* 115:42–57
27. Weidenmuller HA, Mitchell GE (2009) Random matrices and Chaos in Nuclear Physics: nuclear structure. *Rev Mod Phys* 81(2):539–589
28. Wu TY, Fan XN, Wang KH, Pan JS, Chen CM, Wu JMT (2018) Security analysis and improvement of an image encryption scheme based on chaotic tent map. *J Inf Hiding Multimed Signal Process* 9(4):1050–1057
29. Yang JC, Wright J, Huang TS, Ma Y (2010) Image super-resolution via sparse representation. *IEEE Trans Image Process* 19(11):2861–2873
30. Yang S J, Ye X, Zhang S J (2017) A new infrared turbulent fuzzy image restoration algorithm based on Gaussian function parameter identification, International Conference on Image, Vision and Computing, 423–427
31. Yao W, Zhang X (2015) Zheng Z M, Qiu W J, a colour image encryption algorithm using 4-pixel Feistel structure and multiple chaotic systems. *Nonlinear Dyn* 81(1):151–168
32. Ye GD, Pan C, Huang XL, Mei QX (2018) An efficient pixel-level chaotic image encryption algorithm. *Nonlinear Dyn* 94(1):745–756
33. Zhang J, Zhang YT (2014) An image encryption algorithm based on balanced pixel and chaotic map. *Math Probl Eng* 216048
34. Zhang H, Wang XY, Wang SW, Guo K, Lin XH (2017) Application of coupled map lattice with parameter  $q$  in image encryption. *Opt Lasers Eng* 88:65–74

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