Complexity

Nonlinear Dynamical System-based Cybersecurity

Lead Guest Editor: Ahmed A. Abd El-Latif
Guest Editors: Christos Volos, Akram Belazi, and Padmapriya Praveenkumar

Nonlinear Dynamical System-based Cybersecurity

Complexity

Nonlinear Dynamical System-based Cybersecurity

Lead Guest Editor: Ahmed A. Abd El-Latif
Guest Editors: Christos Volos, Akram Belazi, and Padmapriya Praveenkumar

Contents

Hindawi
Complexity
Volume 2022, Article ID 5193618, 18 pages
https://doi.org/10.1155/2022/5193618

Research Article

Performance of the 2D Coupled Map Lattice Model and Its Application in Image Encryption

Zhuo Liu,[1,2] **Jin Yuan Liu**,[2,3] **Leo Yu Zhang**,[4] **Yong Zhao**,[1] **and Xiao Feng Gong**[5]

[1]*School of Mathematics and Big Data, Guizhou Education University, Guiyang 550018, China*
[2]*College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[3]*School of Intelligent Technology and Engineering, Chongqing University of Science and Technology, Chongqing 401331, China*
[4]*School of Information Technology, Deakin University, Victoria 3216, Australia*
[5]*Guizhou Science and Technology Information Center, Guiyang 550018, China*

Correspondence should be addressed to Leo Yu Zhang; leo.zhang@deakin.edu.au

Received 15 July 2021; Revised 15 February 2022; Accepted 1 April 2022; Published 11 May 2022

Academic Editor: Padmapriya Praveenkumar

The two-dimensional coupled map lattice (2D CML) model has been extensively employed as the basis component for designing various schemes in the cryptography system due to its complicated chaotic dynamic behavior. In this study, we analyze the chaotic characteristics of the 2D CML model, such as the Lyapunov exponent (LE), synchronization stability, bifurcation, and ergodicity. We then show that the chaotic sequences generated by the 2D CML model are random according to the NIST testing. Furthermore, we propose an image encryption scheme based on the 2D CML model and Singular Value Decomposition (SVD). In our scheme, the SVD method is used to reduce the image storage, and the Red, Green, and Blue channels of a color image will be encrypted through confusion and diffusion. The simulation results, as well as the results of the comparison with other schemes, demonstrate that our scheme possesses outstanding statistics, excellent encryption performance, and high security. It has great potential for ensuring the security of digital images in real applications.

1. Introduction

Chaos has become a fresh challenge in the cryptographic systems [1–7], because of its unique characteristics such as the sensitivity to the initial conditions and the unpredictability of trajectory. In subfields like the stream cipher [8], Hash function [9], and multimedia encryption [10], chaotic systems have been widely used as their basic components to construct cryptographic primitives.

The chaotic system commonly contains two categories. The first is a simple chaotic system, such as the Logistic map, the Tent map, and the Sine map. The chaos-based schemes based on a simple chaotic system have the highlight of being significantly more efficient. However, because of their simplistic structure, the chaotic dynamic behaviors are not sufficiently complicated, and some security vulnerabilities, such as being easy to predict and thus get attacked, exist in those schemes [11, 12]. The second is a higher-dimensional chaotic system, which has a significantly greater Lyapunov exponent (LE) and wider bifurcation interval than the simple one, and its chaotic characteristics are more complicated. As a result, the higher-dimensional one is generally regarded as more suitable for constructing the chaos-based schemes [8–10].

In the past decades, many researchers have committed to the chaos-based image encryption schemes with the aim of resisting attacks that make use of high pixel correlation and redundancy of digital images. According to the discrete output signal of Chen's chaotic system, a chaos-based image encryption algorithm has been presented [13]; the simulation results show that the scheme can withstand a brute-force attack. The spatiotemporal chaos was used to construct a new chaos-based encryption [14], which is both efficient and secure. A new color image encryption scheme using the combination of different 1D chaotic maps was introduced [15]; the experimental results demonstrate that the scheme

owns remarkable performance in noise and attacks. The enhanced Sine map was used to propose a unique image encryption approach in which row-by-row and column-by-column concepts were introduced [16], and the strategy is both efficient and effective. The scheme in [17] studied a novel chaos-based image encryption scheme based on the Lorenz chaotic system, and experimental results demonstrate the effectiveness and superiority of the algorithm. By imitating the jigsaw method, a chaos-based image encryption scheme was designed in [18], and the experiment and security analyses show that the scheme is both secure and efficient. A fast-reaching finite time synchronization approach for chaotic systems along with its application to medical image encryption is proposed in [19], which owns good robustness and a fast convergence rate. A new chaotic system with hyperbolic sinusoidal function is designed in [20], and a novel voice encryption algorithm based on the new system is proposed. The chaos-based satellite image encryption system is shown in [21], and it is secure, reliable, robust, and simple to implement.

For all the aforementioned image encryption schemes [13–21], higher-dimensional chaotic systems are employed as their core. However, for most employed chaotic systems, their LE values are either not sufficiently large or derived by simulations. That said, a theoretic analysis of the desirable characteristics for employing those models in cryptographic applications is still missing. Moreover, even if a desired higher-dimensional chaotic system is used, the above schemes fail to justify the usage of additional heuristic procedures to turn the chaotic sequences into random binary streams. Indeed, without addressing these shortcomings, cryptographic primitives based on higher-dimensional chaotic systems are also vulnerable to simple attacks [22].

To address the aforementioned shortcomings and to better balance efficiency and security, the 2D CML model, whose characteristics have been theoretically analyzed in [23], is used as the key component for constructing a novel image encryption scheme. We choose the piecewise Logistic map (PLM) as the local map, since it is more sophisticated than the Logistic map, and we then theoretically investigate the 2D CML system instantiated with PLM. In particular, for this specific system, its properties like LE, synchronization stability, bifurcation, and ergodicity are all thoroughly studied. When the parameters of the system are appropriately chosen, we show that the chaotic sequences can be directly extracted as random binary stream without any further processing, and the extracted stream passes the NIST randomness test suite. Powered by the theoretical studies, using the singular value decomposition (SVD) method, we reduce the storage of the original image, and the block of the combined image in Red, Green, and Blue can improve the running time of the scheme.

In a nutshell, this work makes the following contributions:

(i) When the PLM is used as the local map, the LE of the 2D CML model is proven to be larger, and its bifurcation and ergodicity become much wider. All these indicate that the 2D CML model has complex chaotic behavior, and it can be used as a good candidate to construct image encryption schemes.

(ii) The random binary stream can be extracted directly by using the chaotic sequences generated by the 2D CML model. In particular, we can obtain 32 bits from each node of the model, and the NIST test suite confirms that the extracted binary sequences have good randomness.

(iii) According to the SVD approach, the storage of the original image becomes smaller, the confusion in the block of the combined image in R, G, B can improve the running time of our scheme, and also the diffusion has been performed based on the chaotic sequences produced by the 2D CML model. The simulation experiments show that our scheme has good encryption performance.

The remaining parts of this work are organized as follows. Section 2 shows the preliminary knowledge, and the characteristics of the 2D CML model are analyzed in Section 3. In Section 4, the random binary sequences based on the 2D CML model are generated. Section 5 studies an image encryption scheme based on SVD and 2D CML chaotic sequences. The performance of the proposed image encryption scheme is evaluated in Section 6 and the last section draws the conclusion of this work.

2. Preliminaries

2.1. CML Model. The CML model proposed by Kaneko is a classic form of the spatiotemporal chaos model [24], and it is formulated as

$$x_{n+1}^{s} = (1-\varepsilon) f\left(x_n^s\right) + \frac{\varepsilon}{2}\left[f\left(x_n^{s-1}\right) + f\left(x_n^{s+1}\right)\right], \quad (1)$$

where $f(\cdot)$ denotes the local chaotic map; $s = 1, 2, \ldots, U$, with U being the size of the CML model. The periodic boundary condition of the CML model is $x_n^0 = x_n^{U+1}$.

To improve the complexity of CML, it is later extended into higher-dimensional spaces, for example, the two-dimensional one. In the 2D CML model, the local node is affected by the nearest four nodes simultaneously; that is,

$$x_{n+1}^{s,t} = (1-\varepsilon) f\left(x_n^{s,t}\right) + \frac{\varepsilon}{4}\left[f\left(x_n^{s-1,t}\right) + f\left(x_n^{s+1,t}\right) + f\left(x_n^{s,t+1}\right) + f\left(x_n^{s,t-1}\right)\right], \quad (2)$$

where $s = 1, 2, \ldots, R$ and $t = 1, 2, \ldots, L$ are the row and column indexes of the nodes, respectively. The periodic boundary conditions are $x_n^{R+1,t} = x_n^{0,t}$ and $x_n^{s,L+1} = x_n^{s,0}$. From equation (2), the value of the current node $x_{n+1}^{s,t}$ at the $(n+1)$-timestamp is determined by the local node $f(x_n^{s,t})$, the left node $f(x_n^{s-1,t})$, the right node $f(x_n^{s+1,t})$, the top node $f(x_n^{s,t-1})$, and the bottom node $f(x_n^{s,t+1})$, respectively.

According to [23], the LE values of 2D CML are given by

$$\mathrm{LE}s = \mathrm{LE}_f + \ln\left|1 - \varepsilon + \frac{\varepsilon}{2}\left(\cos\frac{2\pi r}{R} + \cos\frac{2\pi l}{L}\right)\right|, \quad (3)$$

where $r = 1, \ldots, R$, $l = 1, \ldots, L$, and LE_f is the LE value of the employed local chaotic map f $(\cdot)$. When $r = 1$ and $l = 1$, the LEs of 2D CML reach the maximum LE (MLE) LE_f. According to equation (3), we can easily get the following theorem.

Theorem 1. *The MLE of the 2D CML model is independent of the model size, but it is determined by the local chaotic map $f(\cdot)$.*

According to Theorem 1, the local chaotic map has special significance for the 2D CML model and directly decides the MLE value and chaotic characteristics of the model. Consequently, selecting a larger LE in the local map indicates more complexity of the model. As will be discussed later, we use the PLM with $\mu = 4$ and $N = 64$ as the local chaotic map because it has a larger LE.

2.2. The Piecewise Logistic Map. The PLM is the enhanced version of the well-known Logistic map [25], and it possesses much larger LE and more complex chaotic characteristics than the Logistic map. The PLM is defined as

$$x_{m+1} = \mathrm{PLM}(x_m) = \begin{cases} N^2\mu x_m\left(\frac{1}{N} - x_m\right), & 0 < x_m < \frac{1}{N}, \\ 1 - N^2\mu\left(x_m - \frac{1}{N}\right)\left(\frac{2}{N} - x_m\right), & \frac{1}{N} < x_m < \frac{2}{N}, \\ N^2\mu\left(x_m - \frac{1}{N}\right)\left(\frac{i}{N} - x_m\right), & \frac{1}{N} < x_m < \frac{i}{N}, \\ 1 - N^2\mu\left(x_m - \frac{i}{N}\right)\left(\frac{i+1}{N} - x_m\right), & \frac{i}{N} < x_m < \frac{i+1}{N}, \\ \cdots & \cdots \\ N^2\mu\left(x_m - \frac{N-2}{N}\right)\left(\frac{N-1}{N} - x_m\right), & \frac{N-2}{N} < x_m < \frac{N-1}{N}, \\ 1 - N^2\mu\left(x_m - \frac{N-1}{N}\right)(1 - x_m), & \frac{N-1}{N} < x_m < 1, \end{cases} \tag{4}$$

where $x_m \in (0, 1)$ is the state value, $\mu \in (0, 4]$ is the control parameter, and N is the segment number of PLM. When $N = 64$ and $\mu = 4$, its LE value is 4.574594, and hence the MLE of 2D CML is the same.

2.3. The Binary Format. When designing digital image encryption methods based on chaotic systems, the real-valued chaotic orbits need to be converted into binary to obtain pseudorandom sequences (i.e., 0s or 1s). We consider the fixed-point representation of chaotic orbits within the range $[0, 1]$ using Definition 1.

Definition 1. A floating number $D \in [0, 1]$ can be written into the binary format with M bits as follows:

$$D = 0.C^1(x)C^2(x)\cdots C^{M-1}(x)C^M(x), \tag{5}$$

where $C^M(x) \in \{0, 1\}$.

2.4. Singular Value Decomposition. SVD is an effective method for the factorization of an $M \times N\,(M \neq N)$ matrix, and it is commonly used in signal processing and image compression. The general form of SVD is given by

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\mathbf{T}}, \tag{6}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $M \times M$ and $N \times N$ matrices, respectively, and Σ represents the $M \times N$ singular value matrix, whose elements are all 0 except the SVD values on its diagonal.

3. Performance Analyses of the 2D CML Model

As discussed previously, the performance of the 2D CML model is critical for designing chaos-based cryptographic primitives. In the 2D CML model, according to equation (3), its performance is solely determined by the local chaotic map $f(\cdot)$. Therefore, selecting a local map $f(\cdot)$ with a large LE is essential, since it in turn enhances the overall complexity of the 2D CML model. With this consideration, we hereby choose the PLM with $N = 64$ and $\mu = 4$ as the local map.

3.1. The Lyapunov Exponent Analysis. LE is an index used to judge whether a dynamic system is chaotic or not, and a positive LE indicates chaos. Moreover, the larger the value of LE was, the more complex the chaotic system would be. The LE of a chaotic system $x_{n+1} = F(x_n)$ is defined as

$$\mathrm{LE} = \lim_{x \longrightarrow \infty} \frac{1}{n} \ln \left| \prod_{s=0}^{n} F'(x) \right|. \tag{7}$$

Taking the PLM as the local map, we plot the LE values of all 64 nodes $(L = R = 8)$ according to (3) in Figure 1. According to this figure, it can be seen that the LEs lie within the interval $[4, 6]$; all are positive and relatively large (compared to LE of the original Logistic map). This fact demonstrates that the 2D CML model has complex chaotic dynamic behaviors.

Moreover, by taking derivative of equation (3) with respect to ε, we can further have

$$\mathrm{LE}' = \frac{\cos 2\pi r/R + \cos 2\pi l/L - 2}{2 + \varepsilon(\cos 2\pi r/R + \cos 2\pi l/L - 2)}. \tag{8}$$

To select the coupling parameter ε with better chaotic property, we first consider the case where the denominator $2 + \varepsilon(\cos 2\pi r/R + \cos 2\pi l/L - 2)$ of equation (8) is 0. In this case, $r = l = 4$ and $\varepsilon = 0.5$, so $\varepsilon = 0.5$ should be avoided. We then investigate the value of LE' by enumerating all the possibilities of l and r. It turns out that when $\varepsilon \in (0, 0.5)$, $\mathrm{LE}' < 0$ regardless of the choices of l and r, and, depending on specific choices of l and r, LE' can be either positive and negative for $\varepsilon \in (0.5, 1)$. That said, the value of LE monotonically decreases for $\varepsilon \in (0, 0.5)$ and fluctuates for $\varepsilon \in (0.5, 1)$ and smaller ε achieves better chaotic property.

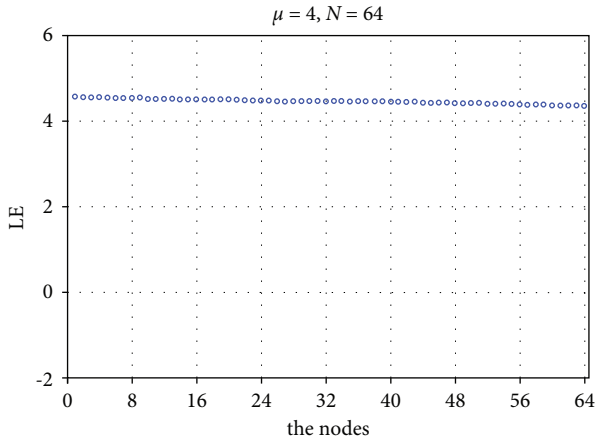


Figure 1: LE of the 8 × 8 2D CML model.

With this consideration and to maintain a certain level of coupling effect, we take the empirical value $\varepsilon = 0.1$ for 2D CML instantiated with PLM in the remainder of this paper.

3.2. The Synchronization Stability Analysis. The stability of periodic orbit and chaos synchronization of the 2D CML model are substantially more complicated [24] compared to its 1D counterpart. However, there is little theoretical study for its configuration. From the standpoint of cryptography applications, the parameter settings should ensure that the 2D CML model runs in a fully developed chaotic state. Thus, we present a theoretical investigation of the synchronization stability for the 2D CML model. Theoretically, for ordered LEs of the 2D CML, the second maximum LE value $\mathrm{LE}_2 > 0$ means that the system is in an asynchronous state, while $\mathrm{LE}_2 < 0$ means that it is synchronous.

To begin with, let $r = R$ and $l = L - 1$; according to equation (3), we can get LE_2 as

$$\mathrm{LE}_2 = \mathrm{LE}_f + \ln\left|1 - \varepsilon + \frac{\varepsilon}{2}\left(\cos 2\pi + \cos\frac{2\pi(L-1)}{L}\right)\right|. \quad (9)$$

Set $\mathrm{LE}_2 = 0$, and the critical value of L is

$$L_c = \left\lfloor \frac{2\pi}{\arccos\left(2e^{-\mathrm{LE}_f} - 2 + \varepsilon\right)/\varepsilon} \right\rfloor. \quad (10)$$

Here, L_c represents the minimum number of nodes that can ensure that the system is in an asynchronous state; that is, $L > L_c$ should be used to make $\mathrm{LE}_2 > 0$.

To verify the above-mentioned analysis, we take the Logistic map,

$$x_{n+1} = 4x_n(1 - x_n), \quad (11)$$

as the local chaotic map and set $R = L = 3$ and $\varepsilon = 0.9$ for the 2D CML model. For this specific 2D CML model, from equation (10), $L_c = 3$.

We randomly initialize the values of the 2D CML, which are denoted as $x_0^{s,t}, s = 1, 2, 3; t = 1, 2, 3$. Then, the 2D CML is iterated 3 times and 100 times and the values are denoted as $x_3^{s,t}$ and $x_{100}^{s,t}$, respectively. We plot $x_0^{s,t}$, $x_3^{s,t}$, and $x_{100}^{s,t}$ in Figure 2. From Figure 2(c), the state values of the nodes in the 2D CML model appear to be synchronized after 100 iterations, which confirms that the 2D CML model is not in a fully developed chaotic pattern. To make $\mathrm{LE}_2 > 0$, we set $L = 4 > L_c = 3$ for the 2D CML and keep all the other parameters unchanged. The simulation results are depicted in Figure 3. It is clear from this figure that no stable synchronous chaos can be observed in the states of the model. Thus, we can conclude that increasing the size of the 2D CML model is an effective way to guarantee that the 2D CML model is not in a synchronous pattern.

3.3. The Bifurcation Analysis. Bifurcation shows the sudden altering of the critical point when changing the parameters in a chaotic system. For the 2D CML model instantiated with the PLM, simulation results indicate that the bifurcations of all 64 nodes are almost the same. Taking the 1^{st} node as an example, we plot its bifurcation diagram in Figure 4. It is clear from this Figure 4 that changing μ significantly influences the bifurcation of the system. When $\mu \in (2, 4)$, the 2D CML model has well-established bifurcation performance. Specifically, the 2D CML model possesses the best bifurcation performance with $\mu = 4$.

3.4. The Ergodicity Analysis. For a chaotic system, ergodicity describes the randomness of statistical results in both time and space. If the states of the system cover a larger interval, the system is more complex. Here, with the parameter settings $\mu = 0.5, 1.0, 1.6, 2, 2.6, 3.0, 3.6$, and 4.0 for the 2D CML instantiated with the PLM, we plot the ergodicity of the model in Figures 5(a)–5(h). As can be seen, the 2D CML model covers the entire interval and has the best chaotic dynamic behavior when $\mu = 4$.

3.5. The Probability Density Distribution. PDD describes the distribution of chaotic state values in the phase space. We plot the PDD of the chaotic sequences generated by all the nodes in Figure 6 for the 2D CML instantiated with the PLM. According to Figure 6, it is clear that PDD of those sequences is uneven, with the peaks appearing in the intervals $[0.0, 0.2]$ and $[0.8, 1.0]$.

4. The Random Chaotic Sequences

According to the above-discussed theoretic analyses and simulation, apparently, when selecting the PLM with $\mu = 4, N = 64$ as the local map and setting $\varepsilon = 0.1$ for 2D CML, the model owns outstanding chaotic dynamic behaviors. Taking the 2D CML model as the key component, we derive random sequences through the following steps:

(i) Step 1: In the 2D CML model, set $R = L = 8$ and $\varepsilon = 0.1$, choose the PLM with $\mu = 4, N = 64$, iterate the model 1,000 times to avoid transition effect, and abandon these first 1,000 states.

(ii) Step 2: Continue to iterate the 2D CML model. For each iteration, a floating number $B \in (0, 1)$ is derived from each node, and there are totally 64

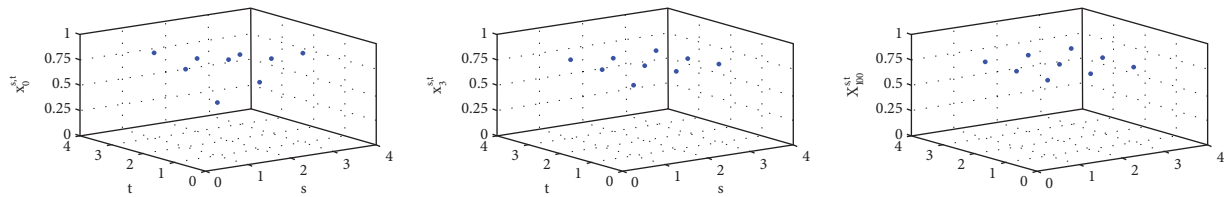


Figure 2: The values of the 3×3 2D CML model with the Logistic map: (a) $x_0^{s,t}$, (b) $x_3^{s,t}$, and (c) $x_{100}^{s,t}$.

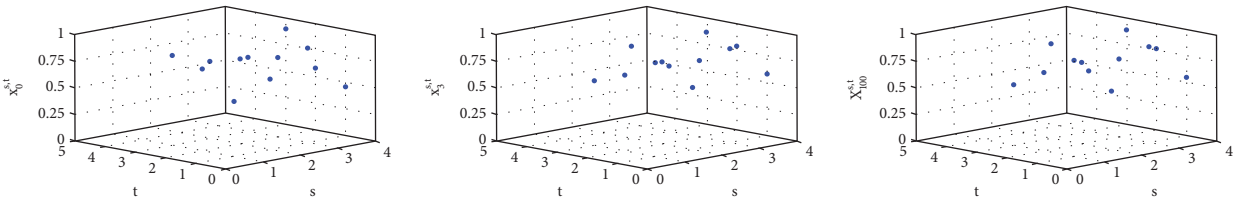


Figure 3: The values of the 3×4 2D CML model with the Logistic map: (a) $x_0^{s,t}$, (b) $x_3^{s,t}$, and (c) $x_{100}^{s,t}$.

floating numbers. Transform B into 64 binary bits according to Definition 1; that is,

$$B = 0.w_1 w_2 \cdots w_{63} w_{64}. \tag{12}$$

(iii) Step 3: The least significant 32 bits are the required binary bits; that is,

$$B' = w_{32} w_{33} \cdots w_{63} w_{64}. \tag{13}$$

For a single iteration of the 8×8 2D CML model, all those 64 nodes can directly generate $32 \times 64 = 2048$ bits. To further analyze the randomness of the binary stream, we use the NIST test suite and the key sensitivity analysis to demonstrate that the binary stream derived using the method above owns excellent randomness and key sensitivity performance.

4.1. Testing Results Analysis. The statistical test package launched by NIST is currently the most authoritative tool for testing the pseudorandom sequences, and it contains 15 subtests. For each test, there exists a p_{value} for measuring whether the sequences can pass the random testing successfully. If $p_{\text{value}} \geq \alpha$, it indicates pass. Otherwise, the sequences fail that test. We randomly initialize the 2D CML model according to the method in Section 4 and run the method 488, 888 times to have 1, 000 M bits. Set $\alpha = 0.01$ and split the 1, 000 M bits to 1, 000 groups of 1 M bit; the NIST test is then performed on these 1, 000 groups and the results are listed in Table 1. According to Table 1, it is clear that all the p_{value} are greater than 0.01, and the minimum pass rate and the maximum pass rate are 0.9841 and 0.9952, respectively. The testing results of p_{value} and pass rate show that the chaotic sequences produced by the 2D CML model possess good randomness.

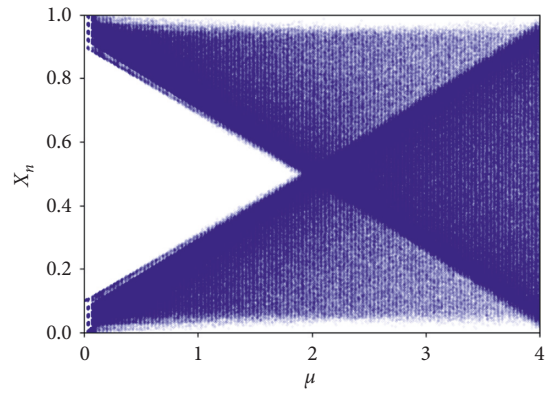


Figure 4: Bifurcation of the 1^{st} 1st node with the change in μ in the 8×8 2D CML model instantiated with PLM.

4.2. Sensitivity Analysis. Sensitivity means that a tiny change of the parameters will lead to huge changes in the output chaotic sequences. We set the parameters of the 2D CML model as the two following proximal cases:

Case I: $\varepsilon = 0.1$, $\mu = 4$, and $x_0 = 0.49903121525011673$;

Case II: $\varepsilon = 0.1$, $\mu = 4$, and $x_0 = 0.49903121625011673$; their outputted pseudorandom binary streams are collected, respectively. To verify the sensitivity, the steams are then used to mask the digital Lena image. The two versions of the masked image and their difference are shown in Figure 7.

Looking into the details of the difference image, the different rate of the encrypted images with Case I and Case II is 99.60%, and the histograms of the two masked images are almost uniform, as shown in Figures 7(f) and 7(g). Hence, the pseudorandom sequences derived from the method discussed in Section 4 own pretty good sensitivity.

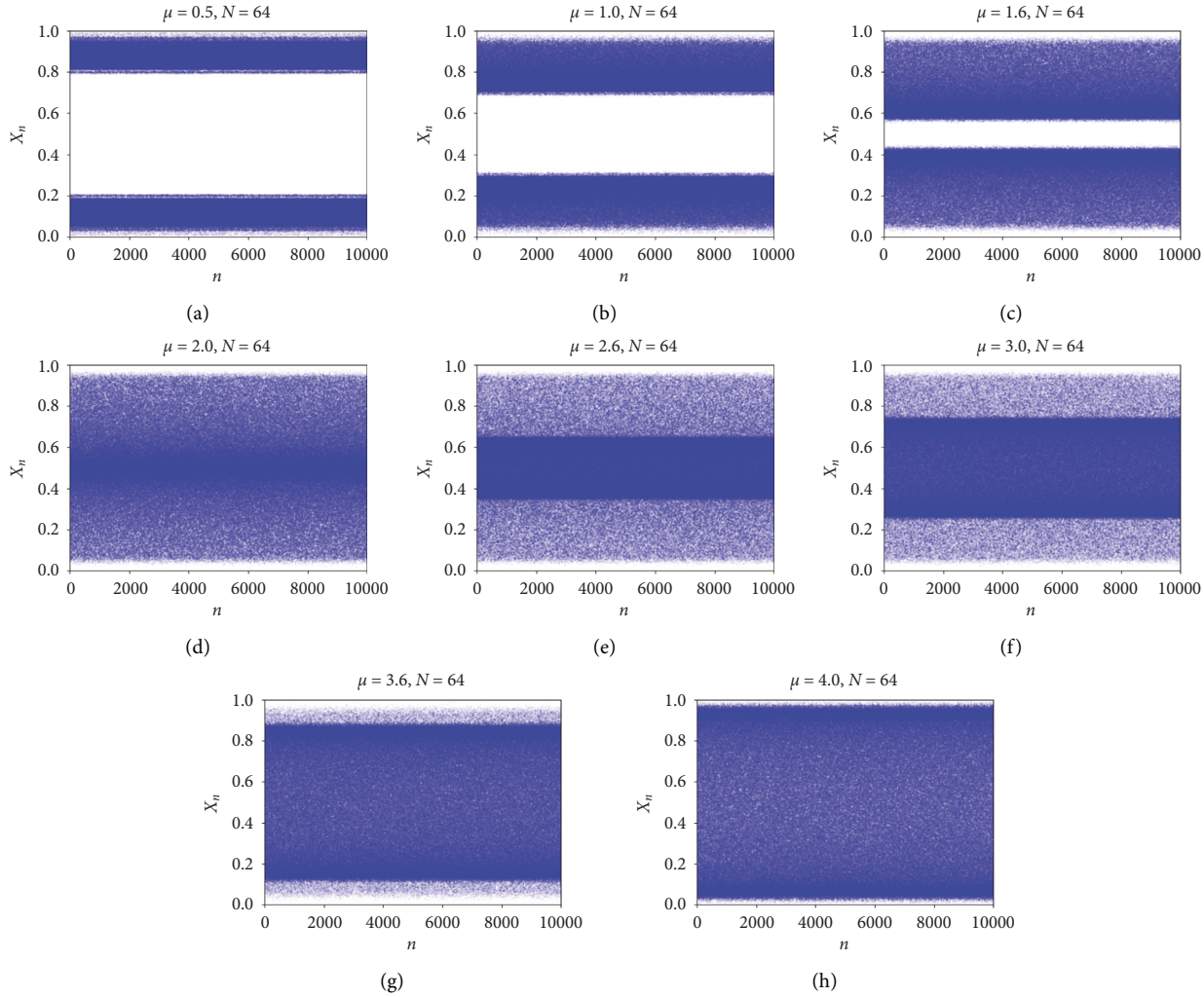


Figure 5: Ergodicity of the 8×8 2D CML model with different μ.

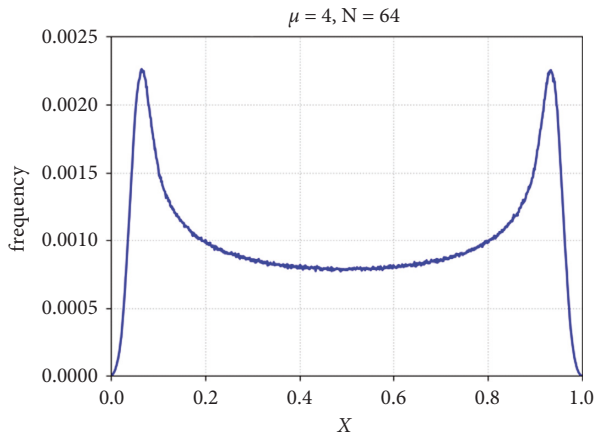


Figure 6: PDD of the 8×8 2D CML model.

5. The Proposed Encryption Scheme

Section 4 demonstrates that the chaotic sequences generated by the 2D CML model are random and they also own good sensitivity against the parameters. This section takes advantage of the chaotic sequences generated by the 2D CML model to design a digital image encryption scheme as an application. As depicted by Figure 8, the proposed image encryption scheme consists of three core components: SVD, confusion, and diffusion. The purpose of using SVD is to reduce storage and improve efficiency. For confusion, the cross-plane permutation in R, G, and B channels has been employed to comprehensively shuffle the pixel positions in the three-color planes via a single operation. The diffusion is performed based on the random chaotic sequences.

5.1. The Encryption Algorithm. The proposed image encryption algorithm, depicted by Figure 9, is elaborated as the four following steps, and also its pseudocode is presented as algorithm 1.

5.2. The Decryption Algorithm. The decryption is basically the inverse of the encryption process. In detail, the encrypted image **C** can be decrypted into the original image **P** according to the following steps. The pseudocode is shown in Algorithm 2.

Table 1: The test results of NIST 800-22.

No.	Test index	Pass number/failure number	Pass rate	P_{value}	Results
1	FT	992/08	0.9920	0.440975	Success
2	FBT	991/09	0.9910	0.233162	Success
3	CST (forward)	989/11	0.9890	0.397688	Success
	CST (reverse)	989/11	0.9890	0.408275	Success
4	RT	995/05	0.9950	0.217857	Success
5	LROBT	989/11	0.9890	0.682823	Success
6	BMRT	993/07	0.9930	0.755819	Success
7	DFTT	992/08	0.9920	0.560545	Success
8	NTMT*	990/10	0.9899	0.525430	Success
9	OTMT	992/08	0.9920	0.448424	Success
10	MUST	985/15	0.9850	0.149495	Success
11	AET	987/13	0.9870	0.883171	Success
		RET (the sample size = 629)			
	(1)	623/06	0.9905	0.744751	Success
	(2)	620/09	0.9857	0.980003	Success
	(3)	619/10	0.9841	0.705598	Success
12	(4)	626/03	0.9952	0.731821	Success
	(5)	625/04	0.9936	0.548839	Success
	(6)	621/08	0.9873	0.526040	Success
	(7)	622/07	0.9889	0.462960	Success
	(8)	622/07	0.9889	0.830070	Success
		REVT (the sample size = 629)			
	(1)	626/03	0.9952	0.692344	Success
	(2)	625/04	0.9936	0.261610	Success
	(3)	625/04	0.9936	0.418149	Success
	(4)	623/06	0.9905	0.290356	Success
	(5)	622/07	0.9889	0.089615	Success
	(6)	621/08	0.9873	0.854868	Success
	(7)	621/08	0.9873	0.882929	Success
	(8)	624/05	0.9921	0.299642	Success
13	(9)	621/08	0.9873	0.251135	Success
	(10)	621/08	0.9873	0.095926	Success
	(11)	624/05	0.9921	0.906025	Success
	(12)	624/05	0.9921	0.131195	Success
	(13)	626/03	0.9952	0.435787	Success
	(14)	623/06	0.9905	0.018417	Success
	(15)	620/09	0.9857	0.516370	Success
	(16)	621/08	0.9873	0.077315	Success
	(17)	620/09	0.9857	0.722038	Success
	(18)	621/08	0.9873	0.194881	Success
14	ST1	994/06	0.9940	0.397688	Success
	ST2	989/11	0.9890	0.344048	Success
15	LCT	990/10	0.9900	0.166260	Success

6. Experimental Analysis

To further analyze the characteristics of the proposed encryption algorithm, the following simulations are performed.

6.1. The Encryption and Decryption Image. For the plain Lena (512×512) and Chocolate (256×256) images, use SVD to decompose the images with the rate $p = 0.3$; the results are shown in Figures 10(b) and 10(d), respectively. From visual inspection, these two images are almost the same as the plain counterparts, shown in Figures 10(a) and 10(c).

Figures 10(e)–10(h) further depict the confusion result from equation (14). Moreover, the encrypted images and the recovered images are shown in Figures 10(i)–10(l). According to Figures 10(i)–10(l), the encrypted images are noisy, and the decrypted images in Figures 10(j) and 10(l) are the same as the original Lena and Chocolate images in Figures 10(a) and 10(d).

6.2. The Statistics Results. The histogram reflects the distribution of the image's pixel value; the more uniform the histogram of the encrypted image is, the better the scheme is. We plot the histogram results of the original images (Lena and Chocolate) and the encrypted images in Figures 11 and 12, respectively. According to Figures 11(a)–11(c), the histogram results of the original image are highly uneven. However, the histograms of the encrypted images in R, G,

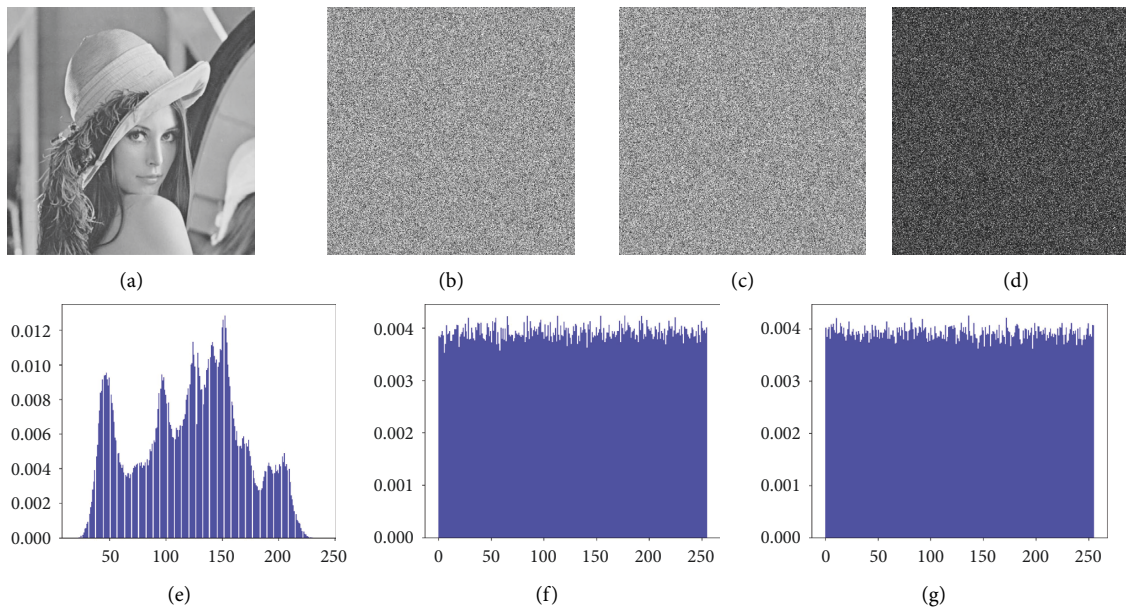


Figure 7: The results of Lena image and the encrypted Lena image. (a) The plain Lena image; (b) the encrypted image with case (i); (c) the encrypted image with case II; (d) the different image with case I and case II; (e) histogram of the plain Lena image; (f) histogram of the encrypted image with case (i); (g) histogram of the encrypted image with case II.

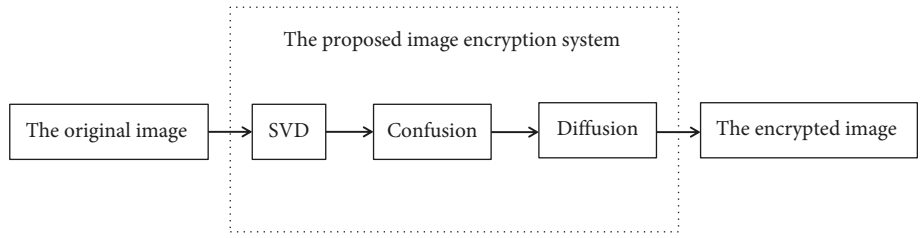


Figure 8: The core parts of our proposed image encryption scheme.

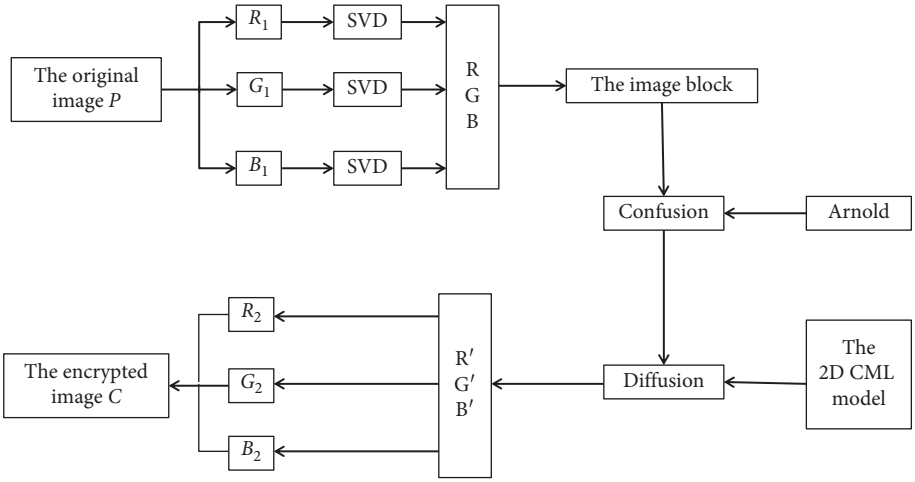


Figure 9: The proposed image encryption scheme.

and B channels are uniform in Figures 11(d)–11(f), and the histogram results of the Chocolate image are similar to those of the Lena image.

According to the approach depicted in [26], the uniformity of histogram can be assessed via the χ^2 test. In this test, the significance value is set as 0.05; if the resultant $P-\text{value} < 0.05$, the decision is 1 (rejecting the hypothesis); if the resultant $P-\text{value} > 0.05$, the decision is 0 (accepting the hypothesis). The values of the χ^2 test for the histogram results of the Lena image and the Chocolate image shown in

Input: The original image $\mathbf{P}$ with size $N \times N$
Output: The cipher image $\mathbf{C}$
(1) Use SVD to decompose $\mathbf{P}$ and get the inverse-transformed image
(2) Divide the inverse-transformed image into $\mathbf{R}_1$, $\mathbf{G}_1$ and $\mathbf{B}_1$ to get the new matrix $\mathbf{P}'$
(3) Divide $\mathbf{P}'$ into small blocks with size 3×1
(4) **while** time $\leq$ count1 **do**
(5) $$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \% N$$
(6) **end**
(7)
(8) **while** time $\leq$ count2 **do**
(9) $\mathbf{P}'''(t) = \mathbf{P}''(t) \oplus \mathbf{H}(t) \oplus \mathbf{P}'''(t-1)$,
(10) **end**
(11) Divide the sequence $\mathbf{P}'''$ into three 2D matrices $\mathbf{R}_2$, $\mathbf{G}_2$, $\mathbf{B}_2$ with size $N \times N$ to form the R, G, B channel of the cipher image $\mathbf{C}$;

(i) Step 1: For an original image $\mathbf{P}$ with size $N \times N$, use SVD to decompose $\mathbf{P}$ and keep $p = 0.3$ of the singular values. Then, separate the inverse-transformed image into $\mathbf{R}_1$, $\mathbf{G}_1$, and $\mathbf{B}_1$ according to its color channels. Stack the three matrices $\mathbf{R}_1$, $\mathbf{G}_1$, and $\mathbf{B}_1$ to get a new matrix $\mathbf{P}'$ with size $3N \times N$.
(ii) Step 2: Divide matrix $\mathbf{P}'$ into small blocks with size 3×1, and in total there will be $N \times N$ blocks. Use the following equation:
$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod (N),$$
with $a = 3$ and $b = 4$ to confuse the $N \times N$ blocks of matrix $\mathbf{P}'$ for a few times (count1). The resultant block-shuffled matrix is denoted as $\mathbf{P}''$.
(iii) Step 3: Stack $\mathbf{P}''$ row by row to get a sequence of length $3N \times N$ and generate a chaotic sequence $\mathbf{H}$ of length $3N \times N$ with the method in Section 4, and then diffuse $\mathbf{P}''$ for some times (count2) by the following equation:
$\mathbf{P}'''(t) = \mathbf{P}''(t) \oplus \mathbf{H}(t) \oplus \mathbf{P}'''(t-1)$,
where $t \in 1, 2, \ldots 3N \times N$ and $\mathbf{P}'''(0) = 69$.
(iv) Step 4: Divide sequence $\mathbf{P}'''$ into three 2D matrices $\mathbf{R}_2$, $\mathbf{G}_2$, and $\mathbf{B}_2$ with size $N \times N$ to form the R, G, and B channel of the cipher image $\mathbf{C}$.

Algorithm 1: The proposed image encryption algorithm.

Input: The cipher image $\mathbf{C}$
Output: The original image $\mathbf{P}$ with size $N \times N$
(1) The encrypted image $\mathbf{C}$ with size $N \times N$ is
(2) divided $\mathbf{C}$ into $\mathbf{R}_2$, $\mathbf{G}_2$, $\mathbf{B}_2$; Combine those three components and reshape it to a sequence $\mathbf{P}'''$ of length $3N \times N$;
(3) **while** time $\leq$ count1 **do**
(4) $$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} \% N$$
(5) **end**
(6) **while** time $\leq$ count2 **do**
(7) $\mathbf{P}'(t) = \mathbf{P}''(t) \oplus \mathbf{H}(t) \oplus \mathbf{P}'(t-1)$
(8) **end**
(9) Recover the original image $\mathbf{P}$ from $\mathbf{P}'$ according to SVD.

(i) Step 1: The encrypted image $\mathbf{C}$ with size $N \times N$ is divided $\mathbf{C}$into $\mathbf{R}_2$, $\mathbf{G}_2$, and $\mathbf{B}_2$; then combine those three components and reshape it to a sequence $\mathbf{P}'''$ of length $3N \times N$.
(ii) Step 2: $\mathbf{P}'''$ is then reshaped to a matrix with size $3N \times N$, and it will be further divided into blocks of size 3×1. All $N \times N$ blocks of $\mathbf{P}'''$ will be shuffled by using the following equation for the same number of times used for encryption:
$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} \bmod (N),$$
where $a = 3, b = 4$. The result is denoted as $\mathbf{P}''$.
(iii) Step 3: Use the chaotic sequences $\mathbf{H}$ to diffuse sequence $\mathbf{P}''$ to get $\mathbf{P}'$; that is,$\mathbf{P}'(t) = \mathbf{P}''(t) \oplus \mathbf{H}(t) \oplus \mathbf{P}'(t-1)$,
where $t \in 1, 2, \ldots, 3N \times N$.
(iv) Step 4: Recover the original image $\mathbf{P}$ from $\mathbf{P}'$ according to SVD.

Algorithm 2: The image decryption algorithm.

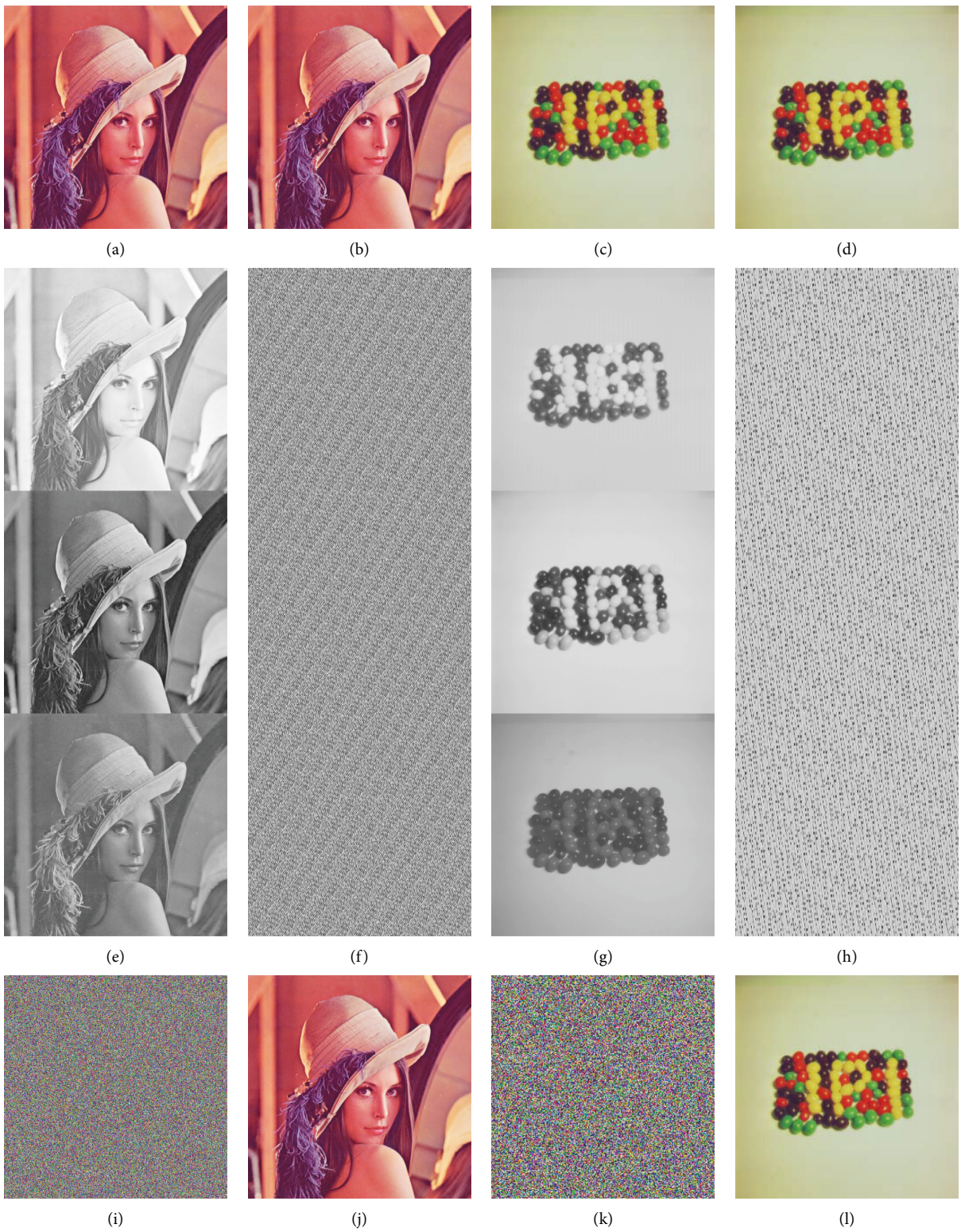


Figure 10: The original image and the SVD image.

Figures 11 and 12 are listed in Table 2. It can be seen from this table that all those P values (i.e., 0.9005, 0.7919, 0.6577, 0.5449, 0.2246, and 0.2069) are greater than the significance value 0.05 for both the encrypted Lena and Chocolate images, thus validating the uniformity of the histograms. So, it is evident that the redundancy of plain images is completely concealed, which confirms the failure of statistical attack.

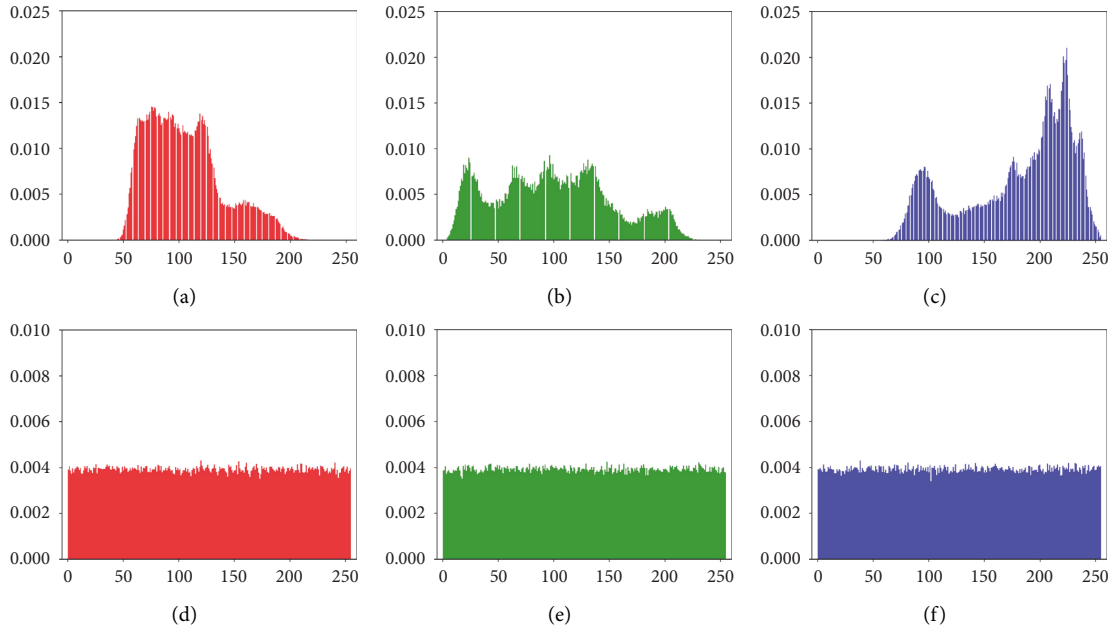


Figure 11: The histogram of the Lena image and the encrypted Lena image. (a) Histogram of the Lena image in R; (b) histogram of the Lena image in G; (c) histogram of the Lena image in B; (d) histogram of the encrypted Lena image in R; (e) histogram of the encrypted Lena image in G; (f) histogram of the encrypted Lena image in B.

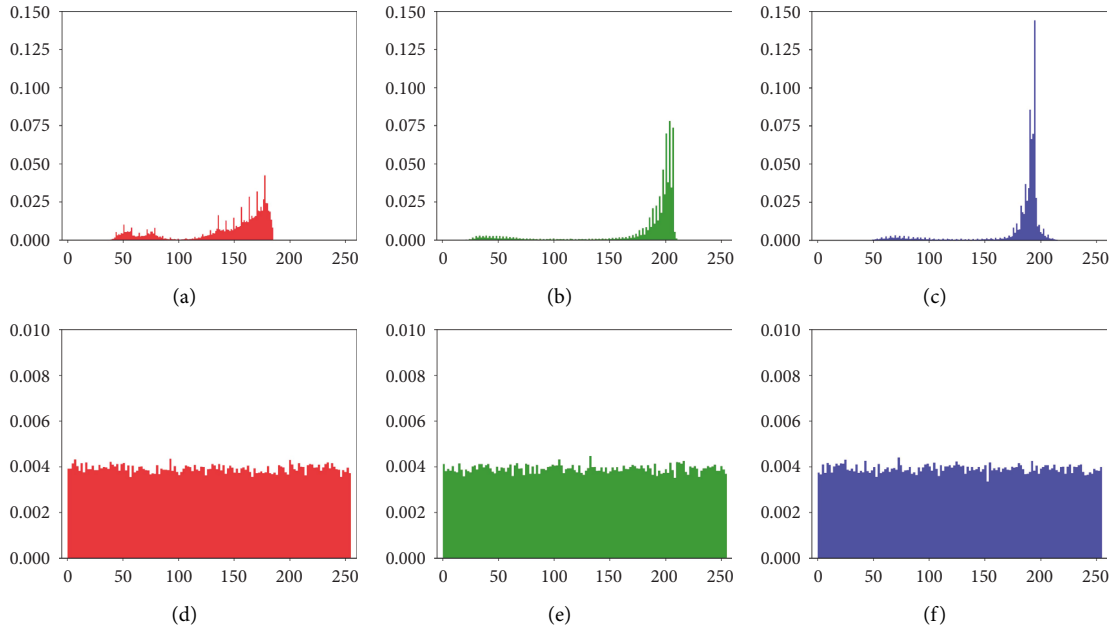


Figure 12: The histogram results of Chocolate image and the encrypted Chocolate image. (a) Histogram of the Chocolate image in R; (b) histogram of the Chocolate image in G; (c) histogram of the Chocolate image in B; (d) histogram of the encrypted Chocolate image in R; (e) histogram of the encrypted Chocolate image in G; (f) histogram of the encrypted Chocolate image in B.

The correlation coefficient is commonly used to measure the independence of horizontal (H), vertical (V), and diagonal (D) adjacent pixels. It is defined by

$$\operatorname{cov}(x, y) = E\{(x - E(x))(y - E(y))\}, \tag{14}$$

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{15}$$

where x and y are the adjacent pixel values and $E(x) = \sum_{i=1}^{P} x_i/P$ and $D(x) = \sum_{i=1}^{P} (x_i - E(x))^2/P$ with P being the number of the pixel pairs.

We use 2,000 pairs for each of the H, V, and D directions and present the correlation values in Figures 13 and 14. According to Figures 13(a)–13(i), the correlation coefficients in the H, V, and D directions of R, G, and B channels are concentrated. However, the correlation coefficients of the

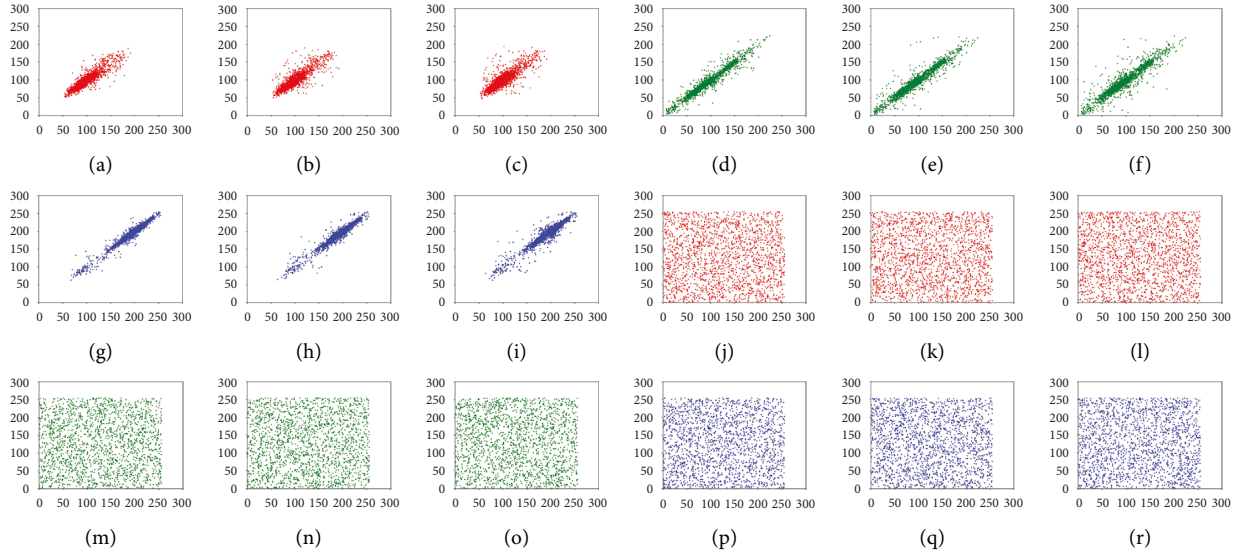


FIGURE 13: The correlation results of the Lena image and the encrypted Lena image. (a) (H, V, D) correlation of the Lena image in R; (b) (H, V, D) correlation of the Lena image in G; (c) (H, V, D) correlation of the Lena image in B; (d) (H, V, D) correlation of the encrypted Lena image in R; (e) (H, V, D) correlation of the encrypted Lena image in G; (f) (H, V, D) correlation of the encrypted Lena image in B.

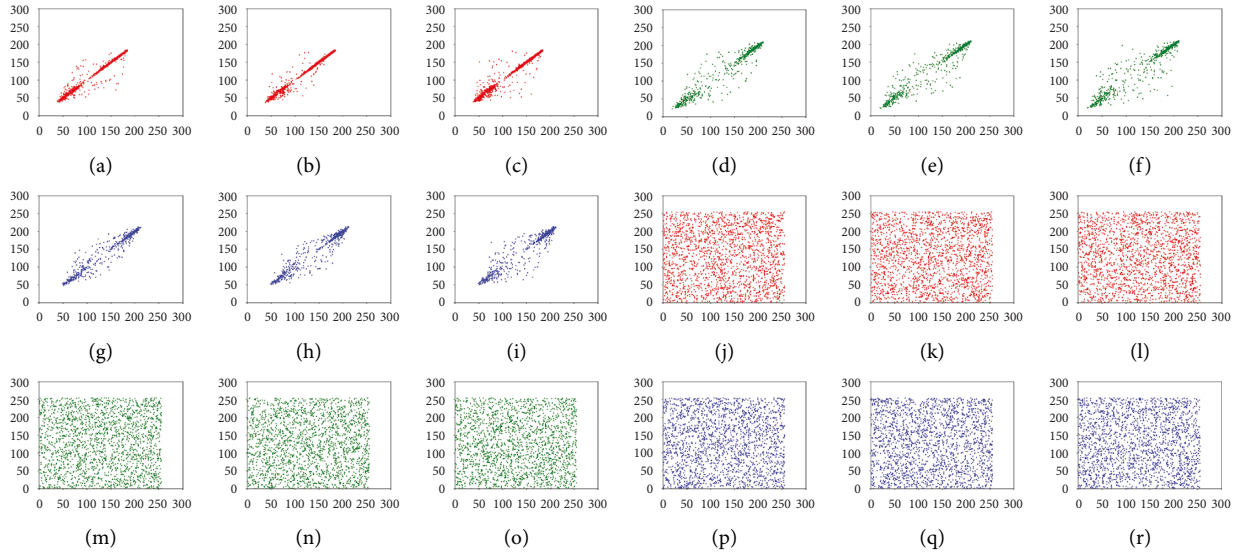


FIGURE 14: The correlation results of the Chocolate image and the encrypted Chocolate image. (a) (H, V, D) correlation of the Chocolate image in R; (b) (H, V, D) correlation of the Chocolate image in G; (c) (H, V, D) correlation of the Chocolate image in B; (d) (H, V, D) correlation of the encrypted Chocolate image in R; (e) (H,V, D) correlation of the encrypted Chocolate image in G; (f) (H, V, D) correlation of the encrypted Chocolate image in B.

encrypted image become uniform, as shown in Figures 13(j)–13(r). Furthermore, we calculate the correlation coefficients of 2, 000 pairs of adjacent pixels in the H, V, and D directions according to equations (18) and (19). The correlation coefficient results are listed in Table 3. It can be seen from this table that the correlation coefficients are all close to 0, which indicates that the pixels of the encrypted image are almost independent of each other.

6.3. Shannon Entropy Analysis. The Shannon entropy reflects the average information contained in an image. It is defined as

$$IE = \sum_{I=0}^{n} p(X_I)\log_2^{p(X_I)}, \tag{16}$$

where X_I is the grayscale value of the image and $p(X_I)$ is the rate of the grayscale value X_I. In the encrypted image, the ideal entropy of a grayscale pixel is 8.0. The global Shannon entropy values of the original image and the encrypted image are calculated via equation (20) and listed in Table 4. According to this table, the values of global Shannon entropy of the encrypted images are quite near 8.0.

To overcome the weaknesses of the global Shannon entropy, such as inaccuracy, inconsistency, and low

Table 2: Histogram uniformity assessment based on the chi-square test.

Image	Histogram of the encrypted Lena image			Histogram of the encrypted Chocolate image		
	R	G	B	R	G	B
P values	0.2246	0.6577	0.2069	0.9005	0.7919	0.5449
Decision (H = 0 or 1)	0; accepted	0; accepted	0; accepted	0; accepted	0; accepted	0; accepted

Table 3: The correlation coefficients of the original image and the encrypted image.

Image	Channel	Original image			Encrypted image		
		H	V	D	H	V	D
Lena	R	0.8946	0.9247	0.8636	−0.0061	0.0042	−0.0007
	G	0.9562	0.9714	0.9307	−0.0040	−0.0003	−0.0045
	B	0.9727	0.9826	0.9515	−0.0018	−0.0013	−0.0032
Chocolate	R	0.9889	0.9851	0.9794	−0.0029	0.0095	−0.0022
	G	0.9768	0.9823	0.9647	0.0021	−0.0077	−0.0008
	B	0.9713	0.9780	0.9573	0.0024	−0.0049	0.0047

efficiency, we use the local Shannon entropy proposed in [27] to measure the encrypted image. For this purpose, we select some nonoverlapping image blocks in the encrypted image and compute the local Shannon entropy value of each block and further calculate the mean of those Shannon entropy values via the following equation:

$$H_{k,T_B}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k}, \tag{17}$$

where k is the number of the randomly selected nonoverlapping image blocks, its minimum number is 30, and T_B is the block size of the nonoverlapping image block.

In our testing, the parameter k is set as 40. Moreover, the block size T_B is 4096 (64×64). The local Shannon entropy values of the encrypted Lena and Chocolate images are presented in Table 5. As can be seen from the table, the mean values of the encrypted Lena image's local Shannon entropy in RGB channel are 7.997297, 7.997129, and 7.997207, respectively, and those of the encrypted Chocolate image's local Shannon entropy are 7.973989, 7.973779, and 7.975590. Both are close to the ideal value of 7.984977322 for 8-bit grayscale images with $64 \times 64 \times 3$ in [26, 28]. To summarize, both global and local Shannon entropy values are very close to the ideal value, which demonstrates the high randomness of the encrypted images.

Table 4: Global Shannon entropy of the original image and the encrypted image.

Image	Channel	Original image	Encrypted image
Lena	R	6.9684	7.9992
	G	7.5940	7.9999
	B	7.2531	7.9992
Chocolate	R	6.5464	7.9975
	G	5.6947	7.9974
	B	5.2626	7.9972

Table 5: Local Shannon entropy of the encrypted image.

Image	Channel	Local Shannon entropy
Lena	R	7.997297
	G	7.997129
	B	7.997207
Chocolate	R	7.973989
	G	7.973779
	B	7.975590

6.4. Differential Attack Analysis. The differential attack is an attack method in which the attacker slightly modifies the plaintext and compares the difference of the ciphertexts generated before and after the modification. The number of pixels change rate (NPCR) and the unified average changing intensity (UACI) are two important indicators to judge whether the encryption scheme can resist the differential attack. Those two indexes are defined as

$$\begin{aligned} \mathrm{NPCR}(P_1, P_2) &= \frac{1}{M \times N} \sum_{a=1}^{M} \sum_{b=1}^{N} \mathrm{Sign}(\mathrm{D}(a,b)), \\ \mathrm{UACI}(P_1, P_2) &= \frac{1}{M \times N} \left(\sum_{a=1}^{M} \sum_{b=1}^{N} \frac{D(a,b)}{255} \right), \end{aligned} \tag{18}$$

Table 6: Theoretical NPCR critical values for different image size.

Size	Theoretical NPCR critical values		
	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.001$
512×512	$N^*_{0.05} = 0.995893$	$N^*_{0.01} = 0.995810$	$N^*_{0.001} = 0.995717$
256×256	$N^*_{0.05} = 0.995693$	$N^*_{0.01} = 0.995527$	$N^*_{0.001} = 0.995341$

Table 7: Theoretical UACI critical values for different image size.

Size	Theoretical UACI critical values		
	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.001$
512×512	$u^{*-}_{0.05} = 0.333730$ $u^{*+}_{0.05} = 0.335541$	$u^{*-}_{0.01} = 0.333445$ $u^{*+}_{0.01} = 0.335826$	$u^{*-}_{0.001} = 0.333115$ $u^{*+}_{0.001} = 0.336156$
256×256	$u^{*-}_{0.05} = 0.332824$ $u^{*+}_{0.05} = 0.336447$	$u^{*-}_{0.01} = 0.332255$ $u^{*+}_{0.01} = 0.337016$	$u^{*-}_{0.001} = 0.331594$ $u^{*+}_{0.001} = 0.337677$

Table 8: The NPCR and UACI values of the encrypted image.

Image	Channel	Chang bit	NPCR	UACI
Lena	R	First pixel	0.9960	0.3347
		Last pixel	0.9960	0.3341
	G	First pixel	0.9959	0.3342
		Last pixel	0.9961	0.3351
	B	First pixel	0.9961	0.3347
		Last pixel	0.9960	0.3347
Chocolate	R	First pixel	0.9962	0.3347
		Last pixel	0.9959	0.3348
	G	First pixel	0.9961	0.3346
		Last pixel	0.9959	0.3346
	B	First pixel	0.9959	0.3343
		Last pixel	0.9962	0.3342

where P_1 and P_2 are two encrypted images, $\mathrm{D}(i, j) = |P_1(a, b) - P_2(a, b)|$, and

$$\mathrm{Sign}(\mathrm{D}(a, b)) = \begin{cases} 1, & P_1(a, b) = P_2(a, b), \\ 0, & \text{else.} \end{cases} \tag{19}$$

According to the method in [29], the theoretical NPCR critical values for different sizes with respect to the significance levels $\alpha = 0.05$, $\alpha = 0.01$, and $\alpha = 0.001$ are shown in Table 6; $N^*_{0.05}$, $N^*_{0.01}$, and $N^*_{0.001}$ are the critical values of NPCR to reject the null hypothesis regarding the associated α-level of significance. If the NPCR test values are above N^*_α, and they are random-like with the significance level α. As for theoretical UACI critical values listed in Table 7, when the UACI test values for the encrypted image lie in the interval $[u^{*-}_\alpha, u^{*+}_\alpha]$, the encrypted image passes the UACI test successfully.

We generate the cipher images by modifying the first pixel and last pixel of the same plain image. The resultant NPCR and UACI values are listed in Table 8. It is clear from this table that all the NPCR and UACI values meet the criteria for accepting the null hypothesis with respect to the significance levels (i.e., $\alpha = 0.05$, $\alpha = 0.01$, and $\alpha = 0.001$). In other words, the encrypted images pass the NPCR and UACI tests successfully, and our scheme is resistant to differential attacks.

Table 9: The differences of the encrypted image with Case 1 and Case 2.

Image	Channel	Case 1-Case 2
Lena	R	99.6166%
	G	99.6143%
	B	99.6253%
Chocolate	R	99.6170%
	G	99.5910%
	B	99.6200%

6.5. Key Security Analysis

6.5.1. Key Space Analysis. The larger key space indicates a better security of the encryption algorithm. As for today's computation power, the key space over 2^{128} is secured and infeasible, which can resist the brute-force attacks effectively. In our scheme, the keys are the initial values of 64 nodes of the 2D CML model. It is well known that the floating-point arithmetic defined by IEEE 754 has a precision of 10^{-15}. Therefore, each node has the 10^{15} possibilities; the key space of all 64 nodes is then

$$10^{15} \times \cdots \times 10^{15} = 10^{960}. \tag{20}$$

Clearly, 10^{960} is much larger than 2^{128}. Thus, the key space of our scheme is large enough to resist the brute-force attacks.

6.5.2. Key Sensitivity Analysis. Key sensitivity means a tiny change of the secret key causing huge changes of the encrypted results. To verify the key sensitivity of our scheme, we use two proximal secret keys as Case 1 and Case 2 to test our design:

Case 1: Keep the original initial conditions unchanged

Case 2: The initial condition of one node is changed by a magnitude of 0.00001 and all others remain unchanged

We then list the differences of the two cipher images in Table 9. From this table, the rate of different pixels between two cipher images is larger than 99.59%. It indicates that the proposed scheme possesses good key sensitivity.

Table 10: The correlation coefficient results of our scheme and others.

Image	Original image			Encrypted image		
	H	V	D	H	V	D
Lena in ours	0.8946	0.9247	0.8636	−0.0061	0.0042	−0.0007
Lena in [30]	0.8946	0.9247	0.8636	0.0016	0.0002	0.0038
Lena in [31]	0.8946	0.9247	0.8636	0.0003	0.0040	0.0013
Lena in [32]	0.8946	0.9247	0.8636	0.0013	0.0034	0.0072
Lena in [33]	0.8946	0.9247	0.8636	−0.0031	0.0025	−0.0001
Lena in [34]	0.8946	0.9247	0.8636	0.0046	−0.0028	0.0014
Lena in [35]	0.8946	0.9247	0.8636	0.0005	−0.0070	0.0006
Lena in [36]	0.8946	0.9247	0.8636	−0.0047	0.0028	−0.0043
Lena in [37]	0.9902	0.9908	0.9794	0.0013	0.0047	0.0020

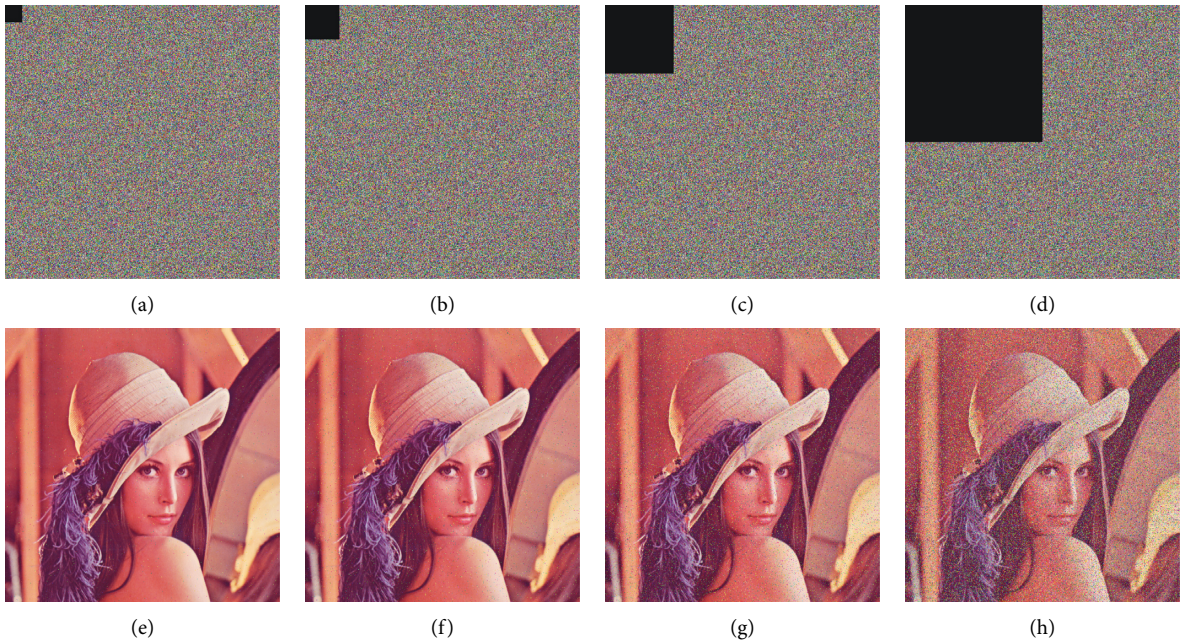


Figure 15: Occlusion attack analysis results of the cipher Lena image. ((a)–(d)) The cipher Lena image with 1/256, 1/64, 1/16, and 1/4 occlusion; ((e)–(h)) decryption results for the images above.

6.6. Comparison Analysis. To further assess the performance of the proposed scheme, quite a few recent studies [30–38] in the same literature are included for comparison. The comparison results of correlation coefficient, NPCR, UACI, and information entropy are listed in Tables 10–12.

Table 10 shows the correlation coefficient values; we can obverse that the correlation coefficient values are quite smaller, almost equal to 0. Meanwhile, Table 11 presents the NPCR and UACI values of our scheme and other schemes; the values of our scheme are closer to the ideal values (NPCR = 0.996094 and UACI = 0.334636) than those of the schemes in [32, 33, 35–37]. Finally, Table 12 describes the information entropy results of our scheme and other schemes; the information entropy results of the R, G, and B channel in our scheme are 7.9992, 7.9999, and 7.9992, respectively, and the average value is 7.999433, which means that they are almost near the ideal value of 8.0. In conclusion, the proposed method performs at least similar to, if not always better than, the others.

6.7. Resistance to Occlusion Attacks. When transmitting the encrypted images, network congestion or malicious destruction may lead to data loss. Occlusion attack is commonly unitized to measure the capacity of recovering the original image from the encrypted image with data loss.

Figures 15(a)–15(d) and Figures 16(a)–16(d) show different encrypted versions of the Lena and Chocolate images with 1/256, 1/64, 1/16, and 1/4 occlusion, respectively, and Figures 15(e)–15(h) and Figures 16(e)–16(h) show the corresponding recovered images of the different occluded cipher images. It is clear that the recovered images are still recognizable even when 25% of the encrypted data are lost.

6.8. Runtime Analysis. The runtime of an encryption scheme is an important factor in practical applications. We implement our scheme with C language on a personal computer equipped with Intel(R) Core(TM) i7-10710U CPU @ 1.10 GHz, 1.61 GHz. The runtimes of our scheme and the literature schemes are given in Table 13. As can be

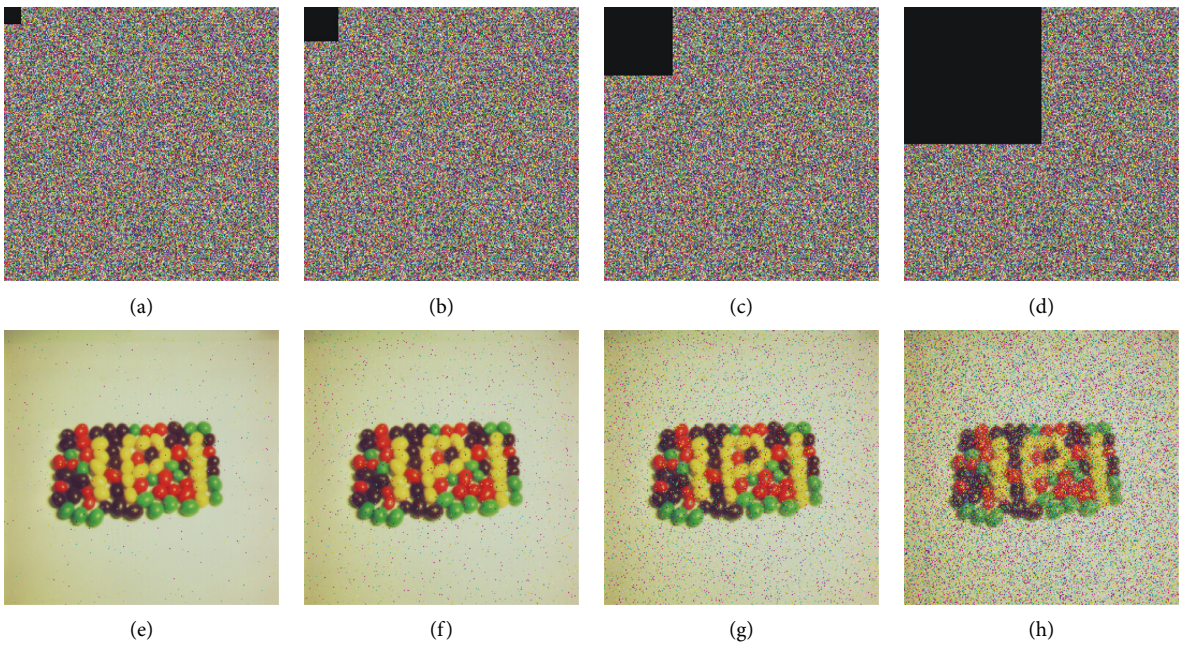


Figure 16: Occlusion attack analysis results of the cipher Chocolate image. ((a)–(d)) The cipher Chocolate image with 1/256, 1/64, 1/16, and 1/4 occlusion; ((e)–(h)) decryption results for the images above.

Table 11: The NPCR and UACI results of our scheme and others.

Image	NPCR				UACI			
	R	G	B	Average	R	G	B	Average
Lena in ours	0.9960	0.9959	0.9961	0.996000	0.3347	0.3342	0.3346	0.334500
Lena in [30]	0.9961	0.9963	0.9961	0.996167	0.3347	0.3345	0.3348	0.334667
Lena in [31]	0.9961	0.9961	0.9961	0.996100	0.3346	0.3346	0.3347	0.334633
Lena in [32]	0.9969	0.9969	0.9969	0.996900	0.3333	0.3333	0.3333	0.333300
Lena in [33]	0.9967	0.9969	0.9969	0.996833	0.3352	0.3353	0.3354	0.335300
Lena in [34]	0.9961	0.9961	0.9961	0.996100	0.3349	0.3349	0.3347	0.334833
Lena in [35]	0.9973	0.9968	0.9970	0.997033	0.3346	0.3345	0.3346	0.334567
Lena in [36]	0.9967	0.9964	0.9965	0.996533	0.3351	0.3350	0.3349	0.335000
Lena in [37]	0.9964	0.9965	0.9964	0.996433	0.3343	0.3347	0.3348	0.334600
Lena in [38]	0.9962	0.9961	0.9962	0.996167	0.3342	0.3343	0.3346	0.334367

Table 12: The information entropy results of our scheme and others.

Image	Encrypted image			Average
	R	G	B	
Lena in ours	7.9992	7.9999	7.9992	7.999433
Lena in [30]	7.9985	7.9985	7.9986	7.998533
Lena in [31]	7.9994	7.9994	7.9993	7.999366
Lena in [32]	7.9997	7.9997	7.9996	7.999666
Lena in [33]	7.9976	7.9972	7.9972	7.997333
Lena in [34]	7.9994	7.9993	7.9992	7.999300
Lena in [35]	7.9972	7.9973	7.9971	7.997200
Lena in [36]	7.9973	7.9965	7.9969	7.996900
Lena in [37]	7.9994	7.9993	7.9994	7.999366
Lena in [38]	7.9917	7.9912	7.9917	7.991533

seen from the table, the encryption times of the Lena and Chocolate images in our scheme for the R channel are 0.369s and 0.097s, respectively, which indicates that it is more efficient than the schemes in [39–44] but inferior to the scheme in [39]. This is because the scheme in [39] has a paralleled architecture and we take the task of designing a paralleled implementation of our design as the future work.

Table 13: Encryption runtime of our scheme and others.

Image	Encrypted Lena image Time (s)	Encrypted Chocolate image Time (s)
Ours	0.369	0.097
[39]	0.079	0.020
[40]	1.260	0.460
[41]	2.290	0.710
[42]	1.590	0.350
[43]	15.140	4.540
[44]	11.420	3.480

7. Conclusion

In this paper, according to theoretical analyses in LE and synchronization stability of the 2D CML model and also the simulation analyses in bifurcation, ergodicity, and PDD, we thoroughly demonstrate that the 2D CML model has good chaotic properties. Moreover, binary sequences can be directly and effectively generated by the 2D CML model, and passing the NIST test suite confirms that the generated sequences possess desired properties for encryption. Relying on this observation, we put forward an image encryption algorithm through confusion and diffusion. Further simulation analyses show that the proposed image encryption scheme possesses good encryption characteristics.

For future work, the study of the characteristics of the higher-dimensional CML system and its applications will be considered.

Data Availability

All data used during the study appear in the submitted article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work described in this paper was supported by Grants from the National Natural Science Foundation of China (no. 61572089), the Science and Technology Foundation Project of Guizhou Province (QianKeHeJiChu[2020]1Y422, QianKeHeJiChu-ZK[2022]YiBan329, QianKeHeJiChu [2019]1425, and QianKeHeJiChu-ZK[2022]YiBan331), the key project research achievements of Guizhou Education University in 2020 (2020ZD006, 2020ZD008), and Guizhou Education Department Youth Science and Technology Talent Growth Project (QianJiaoHe-KY-Zi[2021]239).

References

[1] M. A. Midoun, X. Wang, and M. Z. Talhaoui, "A sensitive dynamic mutual encryption system based on a new 1D chaotic map," *Optics and Lasers in Engineering*, vol. 139, Article ID 106485, 2021.

[2] C. E. C. Souza, D. P. B. Chaves, and C. Pimentel, "One-dimensional pseudo-chaotic sequences based on the discrete arnold's cat map over $\mathbb{Z}_3m$," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 491–495, 2021.

[3] S. Kumari and R. Chugh, "A novel four-step feedback procedure for rapid control of chaotic behavior of the logistic map and unstable traffic on the road," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 12, Article ID 123115, 2020.

[4] R. A. Elmanfaloty and E. Abou-Bakr, "An image encryption scheme using a 1D chaotic double section skew tent map," *Complexity*, vol. 2020, Article ID 7647421, 18 pages, 2020.

[5] G. Ye, C. Pan, Y. Dong, Y. Shi, and X. Huang, "Image encryption and hiding algorithm based on compressive sensing and random numbers insertion," *Signal Processing*, vol. 172, Article ID 107563, 2020.

[6] J. Feng, L. T. Yang, R. Zhang, W. Qiang, and J. Chen, "Privacy preserving high-order Bi-lanczos in cloud-fog computing for industrial applications," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[7] G. Ye, C. Pan, Y. Dong, K. Jiao, and X. Huang, "A novel multi-image visually meaningful encryption algorithm based on compressive sensing and Schur decomposition," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, 2021.

[8] Z. Liu, Y. Wang, Y. Zhao, and L. Y. Zhang, "A stream cipher algorithm based on 2D coupled map lattice and partitioned cellular automata," *Nonlinear Dynamics*, vol. 101, no. 2, pp. 1383–1396, 2020.

[9] J. S. Teh, M. Alawida, and J. J. Ho, "Unkeyed hash function based on chaotic sponge construction and fixed-point arithmetic," *Nonlinear Dynamics*, vol. 100, no. 1, pp. 713–729, 2020.

[10] A. Sahasrabuddhe and D. S. Laiphrakpam, "Multiple images encryption based on 3D scrambling and hyper-chaotic system," *Information Sciences*, vol. 550, pp. 252–267, 2021.

[11] R. Logeshwari and L. Rama Parvathy, "Generating logistic chaotic sequence using geometric pattern to decompose and recombine the pixel values," *Multimedia Tools and Applications*, vol. 79, no. 31-32, Article ID 22375, 2020.

[12] L. Wang and H. Cheng, "Pseudo-random number generator based on logistic chaotic system," *Entropy*, vol. 21, no. 10, p. 960, 2019.

[13] Z.-H. Guan, F. Huang, and W. Guan, "Chaos-based image encryption algorithm," *Physics Letters A*, vol. 346, no. 1–3, pp. 153–157, 2005.

[14] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 514–522, 2011.

[15] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Processing*, vol. 138, pp. 129–137, 2017.

[16] A. Mansouri and X. Wang, "A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme," *Information Sciences*, vol. 520, pp. 46–62, 2020.

[17] S. Xiao, Z. Yu, and Y. Deng, "Design and analysis of a novel chaos-based image encryption algorithm via switch control mechanism," *Security and Communication Networks*, vol. 2020, Article ID 7913061, 12 pages, 2020.

[18] Z. Li, C. Peng, W. Tan, and L. Li, "An effective chaos-based image encryption scheme using imitating jigsaw method," *Complexity*, vol. 2021, Article ID 8824915, 18 pages, 2021.

[19] B. Vaseghi, S. Mobayen, S. S. Hashemi, and A. Fekih, "Fast reaching finite time synchronization approach for chaotic systems with application in medical image encryption," *IEEE Access*, vol. 9, Article ID 25911, 2021.

[20] S. Mobayen, C. Volos, Ü. Çavuşoğlu, and S. Kaçar, "A simple chaotic flow with hyperbolic sinusoidal function and its application to voice encryption," *Symmetry*, vol. 12, no. 2047, pp. 2047–2118, 2020.

[21] B. Vaseghi, S. S. Hashemi, S. Mobayen, and A. Fekih, "Finite time chaos synchronization in time-delay channel and its application to satellite image encryption in OFDM communication systems," *IEEE Access*, vol. 9, Article ID 21332, 2021.

[22] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 305–313, 2018.

[23] Y. Wang, Z. Liu, L. Y. Zhang, F. Pareschi, G. Setti, and G. Chen, "From chaos to pseudorandomness: a case study on the 2-D coupled map lattice," *IEEE Transactions on Cybernetics*, pp. 1–11, 2021.

[24] K. Kaneko, "Pattern dynamics in spatiotemporal chaos: pattern selection, diffusion of defect and pattern competition intermittency," *Physica D: Nonlinear Phenomena*, vol. 34, no. 1-2, 1989.

[25] Y. Wang, Z. Liu, J. Ma, and H. He, "A pseudorandom number generator based on piecewise logistic map," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2373–2391, 2016.

[26] D. Ravichandran, P. Praveenkumar, J. B. Balaguru Rayappan, and R. Amirtharajan, "Chaos based crossover and mutation for securing DICOM image," *Computers in Biology and Medicine*, vol. 72, pp. 170–184, 2016.

[27] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, pp. 323–342, 2013.

[28] E. Yavuz, "A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme," *Optics & Laser Technology*, vol. 114, pp. 224–239, 2019.

[29] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[30] Z. Liu, Y. Wang, and L. Y. Zhang, "A novel compressive image encryption with an improved 2D coupled map lattice model," *Security and Communication Networks*, vol. 6, pp. 1–21, 2021.

[31] L. Huang, S. Cai, M. Xiao, and X. Xiong, "A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion," *Entropy*, vol. 20, no. 7, p. 535, 2018.

[32] X.-J. Tong, M. Zhang, Z. Wang, Y. Liu, H. Xu, and J. Ma, "A fast encryption algorithm of color image based on four-dimensional chaotic system," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 219–234, 2015.

[33] M. Mollaeefar, A. Sharif, and M. Nazari, "A novel encryption scheme for colored image based on high level chaotic maps," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 607–629, 2017.

[34] S. Cai, L. Huang, X. Chen, and X. Xiong, "A symmetric plaintext-related color image encryption system based on bit permutation," *Entropy*, vol. 20, no. 4, p. 282, 2018.

[35] X. Wu, B. Zhu, and Y. Hu, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, 2017.

[36] A. U. Rehman and X. Liao, "A novel robust dual diffusion/confusion encryption technique for color image based on Chaos, DNA and SHA-2, DNA and SHA-2," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2105–2133, 2019.

[37] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Information Sciences*, vol. 546, pp. 1063–1083, 2021.

[38] Y.-Q. Zhang, Y. He, P. Li, and X.-Y. Wang, "A new color image encryption scheme based on 2DNLCML system and genetic operations," *Optics and Lasers in Engineering*, vol. 128, no. 3, Article ID 106040, 2020.

[39] E. Yavuz, "A new parallel processing architecture for accelerating image encryption based on chaos," *Journal of Information Security and Applications*, vol. 63, Article ID 103056, 2021.

[40] X. Chai, Z. Gan, and M. Zhang, "A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion," *Multimedia Tools and Applications*, vol. 76, no. 14, Article ID 15561, 2017.

[41] Z. Eslami and A. Bakhshandeh, "An improvement over an image encryption method based on total shuffling," *Optics Communications*, vol. 286, pp. 51–55, 2013.

[42] X. Huang, "Image encryption algorithm using chaotic Chebyshev generator," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2411–2417, 2012.

[43] X. Wang and D. Xu, "A novel image encryption scheme based on Brownian motion and PWLCM chaotic system," *Nonlinear Dynamics*, vol. 75, no. 1-2, pp. 345–353, 2014.

[44] Y. Zhou, W. Cao, and C. L. Philip Chen, "Image encryption using binary bitplane," *Signal Processing*, vol. 100, pp. 197–207, 2014.

Hindawi
Complexity
Volume 2022, Article ID 3676265, 20 pages
https://doi.org/10.1155/2022/3676265

Research Article

Fast and Robust Image Encryption Scheme Based on Quantum Logistic Map and Hyperchaotic System

Nehal Abd El-Salam Mohamed,[1] **Aliaa Youssif**,[2] **and Hala Abdel-Galil El-Sayed**[3]

[1]*College of Information Technology, Misr University for Science & Technology (MUST), 6th of October City 77, Egypt*
[2]*College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Smart Village 12577, Egypt*
[3]*College of Computers and Artificial Intelligence, Helwan University, Ain Helwan (Helwan University Building), Helwan 11795, Egypt*

Correspondence should be addressed to Nehal Abd El-Salam Mohamed; nehal.mohamed@must.edu.eg

Received 7 October 2021; Revised 9 February 2022; Accepted 14 February 2022; Published 29 March 2022

Academic Editor: Ahmed A. Abd El-Latif

Topic of quantum chaos has begun to draw increasing attention in recent years. So, to ensure the security of digital image, an image encryption algorithm based on combining a hyperchaotic system and quantum 3D logistic map is proposed. This algorithm is applied in four stages. Initially, the key generator builds upon the foundation of mean for any row or column of the edges of the plain image. Its output value is used to yield initial conditions and parameters of the proposed image encryption scheme. Next, it diffuses the plain image by the random sequences generated by 3D hyperchaotic system, and the diffusion process is realized by implementing XOR operation. Then, the diffused image and chaotic sequences are produced by the 3D quantum chaotic logistic map, expressed as a quantum superposition state using density matrix which is a representation of the state of a quantum system, and finally the resulting quantum image is then confused and diffused simultaneously by a unitary matrix generated by logistic chaos using XNOR operation to obtain the final cipher image. Because of the dependence on the plain image, the algorithm can frustrate the chosen-plaintext and known-plaintext attacks. Simulation results and theoretical analysis verify that the presented scheme has high safety performance, a good encryption effect, and a large key space. The method can effectively resist exhaustive, statistical, and differential attacks. Moreover, the encryption time of the proposed method is satisfactory, and the method can be efficiently used in practice for the secure transmission of image information.

1. Introduction

In today's era [1–6], with the fast development of electronic technology and the scale of the communication network, a lot happens over a time of one minute. Along with this rapid development of Internet and multimedia, usage of digital media has increased tremendously in past decades. In this period of digital data technology, today, we are in the sphere of digitally advanced era, where most of the private data and secure digital information is being exchanged by the help of electronic media such as television, smartphones, personal computers, tablets, facsimiles, satellites, and so forth to all corners of the world over just one minute to facilitate the daily needs of people where digital information is being applied in all the fields in the society.

Images originated in some scenarios such as any social media servers, business, personal privacy, healthcare or military systems, organizations, banks, and other private sectors contain private information which is placed and maintained in very big databases, since it can be transmitted, shared, and stored on the Internet, so if this information is stolen or an unauthorized person accesses it, this may cause a serious damage and serious consequences to any organization [7–9].

With the widespread application of a digital image, providing digital image information security in the

transmission channel has become an increasingly serious issue to be urgently solved because the data can be intercepted, cracked, or destroyed [10, 11]. Hence, the security of the important and valuable image information has become a hot recent topic of the field of information security.

Image encryption [5, 12, 13] is one of the possible effective solutions used to protect these images from this threat where it is extensively recognized as a useful technique for secure transmission and its objective is to accomplish privacy and integrity of data. It converts images into noise-like encrypted images with key by disrupting pixel positions or changing pixel values and decryption will reveal the original message or information by utilizing same key utilized for encryption.

To satisfy the emerging demand, a lot of useful image encryption algorithms based on optical transformation, DNA sequence operations, wave motion, Brownian motion, cellular automata, compressive sensing, and chaotic system [6, 14] were developed in literature to secure these digital images.

Since the chaos theory was first proposed by Lorenz, many chaotic phenomena were found in many fields, such as physics, astronomy, chemistry, biology, and medicine. In 1998, Fridrich firstly proposed a chaos-based image encryption algorithm composed of two stages: permutation and diffusion. After that, many scholars have designed numerous efficient algorithms for chaotic systems and chaos-based image encryption to be applied for the secure communications [3, 4, 8, 11].

Chaotic systems [2, 4] have many noteworthy features which satisfy the requirements of image encryption, such as random-like behaviors, high sensitivity to initial conditions and control parameters where the wrong initial condition will lead to nonchaotic behavior, nonperiodicity, and ergodicity, and low cost in the computer operation system and microprocessor [8–10]. Therefore, these systems can be rapidly applied to cryptographic systems which achieve superior performance with respect to the trade-offs between the security and efficiency. However, the appearance of quantum computing brought a great challenge to classic encryption methods [15, 16].

Additionally, with the advancement in technology in the modern era of computer world, brute-force attack [4, 6] will be quite easily performed in quantum computers which are based on quantum information theory. This vulnerability gives potential danger to idealized security required at national security and protected innovation level. To beat this threat, it is necessary to study novel and safer cryptosystem to meet the current safety requirements in image encryption, and, therefore, quantum encryption can be applied in the image encryption process as it gives us a secure encryption method.

Quantum computation [7, 17, 18] has shown great potential for improving information processing speed and enhancing communication security. Combining quantum computing and image encryption is a secure and effective approach to design the encryption algorithms. The essence task of quantum image encryption is to store the images into quantum computers, and then quantum encryption techniques can be exploited to process these images. Due to the promising prospect of quantum image encryption, more and more researchers devoted their attention to developing quantum image representation models and designing image encryption algorithms.

For example, Li et al. [3] proposed an efficient chaos-based image encryption scheme, which uses the imitating jigsaw method containing revolving and shifting operations and shows good performance in both security and speed. Liu et al. [7] proposed a quantum image encryption algorithm based on bit-plane permutation and sine logistic map which has good performance in the aspect of security and the computational complexity is superior to its classical counterpart. Dong et al. [9] proposed a self-adaptive image encryption algorithm based on the quantum logistic map, which can achieve secure communications and frustrate the chosen-plaintext and known-plaintext attacks. In [15], an innovative quantum color image encryption method focused on the Lucas series-based substitution box is suggested to enhance the competence of encryption. This cryptosystem has more excellent key space and significant confidentiality. In [19], an image encryption algorithm based on 3D DNA level permutation and substitution scheme is proposed, where the proposed encryption scheme has large key space and high key sensitivity and may resist some typical attacks, and it may effectively secure the secret image information. El-Latif et al. [20] presented a new method for constructing substitution boxes (S-boxes) based on cascaded quantum-inspired quantum walks and chaos inducement, which will offer gains in many cryptographic applications where the performance of the proposed S-box scheme is investigated via established S-box evaluation criterion and outcomes suggest that the constructed S-box has significant qualities for viable applications information security. In [21], a new method for the encryption by utilizing quantum chaotic maps and continuous chaotic dynamical systems is designed which contributes to achieving the security of data with the minimum time of encryption. Sridevi and Philominathan [22] presented a quantum encryption technique which is built by adopting Haar Integer Wavelet Transform (HIWT), RC6 (Rivest Cipher) block cipher, and DNA (deoxyribonucleic acid) sequences. In addition, a Unified Chaotic Logistic Tent Map (ULTM) has been employed in the permutation phase to produce the pseudorandom sequence for shuffling the RGB planes of the quantum represented source image in spatial and transform domains. This cryptosystem has confirmed the significant immune level of the quantum cryptosystem. In [23], an enhanced quantum scheme is proposed for generalized novel enhanced quantum image representation which has good visual effects and high security. Wen et al. [24] proposed an image cryptosystem adopting a quantum chaotic map and the certain security-enhanced mechanisms where the cryptosystem has excellent performance and can resist various cryptographic attacks. Moreover, the feasibility and effectiveness of the image cryptosystem are verified on the Internet of Things secure communication experimental platform. It proves that the proposed image cryptosystem is a preferred and promising secure communication technology solution.

After conducting a detailed analysis of the breaking methods, it was found that some chaos-based image encryption schemes have security vulnerabilities, which are as follows: (1) key dependence and fixed key; (2) one cycle of permutation-diffusion architecture; (3) low-dimensional systems used for image encryption; (4) single chaotic system still used for encryption operation, which leads to inability to resist brute-force attack; and (5) low sensitivity to all the chaotic secret keys.

To overcome these security shortcomings and design secure and effective image encryption, an image encryption algorithm based on integrating a hyperchaotic system and quantum 3D logistic map is presented in this paper. The essence goals of the proposed scheme are listed as follows:

(i) First, it can fight against the chosen-/known-plaintext attacks due to the use of symmetric key image cryptosystem based on original image.

(ii) Second, the generated key cryptosystem based on the plain image is used to determine the number of cycles of composite chaotic algorithms.

(iii) Third, multidimensional chaotic maps like hyperchaos and 3D quantum logistic map are used which have more chaotic attractors, so the high-dimensional chaotic system has stronger randomness, better confidentiality, greater amount of information, and higher communication efficiency, providing sufficiently large key space and having high security.

(iv) Fourth, two different chaotic systems (quantum logistic map and hyperchaotic Chen's system) are combined, which have the advantage of excellent random sequence to expand the key space, enhance the performance of resisting brute-force attack, and achieve better encryption effect and high level of security.

(v) Fifth, high sensitivity with respect to all secret keys is achieved, which leads to creating a completely different cipher image when applied to the same plain image whenever flipping one bit in a key.

Based on the above literature, it is evident that, for generating excellent encryption effects and producing a highly secure encryption scheme, it is needed to design a combination of hyper- and multidimensional chaotic systems through density matrix which describes the quantum state of a system.

2. Preliminary Knowledge

2.1. Chen's Hyperchaotic System. In order to improve the security and efficiency performance, many image encryption methods based on three-dimensional chaotic systems, hyperchaos, and even spatiotemporal chaos have been presented in recent years [25].

In 1963, Lorenz [26] found the first chaotic attractor in a three-dimensional autonomous system:

$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = cx - xz - y, \\ \dot{z} = xy - bz, \end{cases} \tag{1}$$

where a, b, and c are constant parameters of the system. Typically, when $a = 10$, $b = 8/3$, and $c = 28$, the system is in a chaotic state.

In 1999, Chen [27] discovered another chaotic system with more complex dynamic behaviors than Lorenz system when studying chaotic feedback control. Chen's hyperchaotic system is defined as follows:

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1), \\ \dot{x}_2 = -x_1 x_3 + d x_1 + c x_2 - x_4, \\ \dot{x}_3 = x_1 x_2 - b x_3, \\ \dot{x}_4 = x_1 + k, \end{cases} \tag{2}$$

where a, b, c, d, and k are the system parameters. In this system, when the values of the parameters $(a, b, c, d, k) = (36, 3, 28, -16, -0.7 < k < 0.7)$, the system is hyperchaotic in a very wide parameter range in this case and has many more interesting complex dynamical behaviors than those of Lorenz system. The hyperchaos attractors of this system are shown in Figure 1, while the corresponding bifurcation diagram of state x with respect to k is given in Figure 2.

Its Lyapunov exponents are $\lambda_1 = 1.552$, $\lambda_2 = 0.023, \lambda_3 = 0, \lambda_4 = -12.573$; Lyapunov exponents for this system are depicted in Figure 3. As the hyperchaos has four positive Lyapunov exponents, the prediction time of a hyperchaotic system is shorter than that of a chaotic system [28]; as a result, it is safer than chaos in security algorithm.

2.2. 3D Quantum Logistic Chaotic Map. Quantum chaotic systems are the quantized of classical chaotic system, such as quantum logistic map which [1, 29, 30] is constructed by the classical logistic system and that is a perfect example of complex chaotic maps which arises from nonlinear dynamical equations. Classical chaotic maps have a small range for key space as they suffer from low control parameters which in turn lead to a limited chaotic range, whereas the chaotic maps with higher dimensional as the used one in the proposed scheme can be lead to increase the key space range, have excessive complexity, high degree of randomness, and high sensitivity to initial conditions and control parameters. Therefore, quantum logistic system is suitable as seed system in encryption algorithm.

Based on the classical logistic map and the effect of quantum correlations on a dissipative system [31], the proposed quantum logistic map was applied to image encryption, which can be defined as follows:

$$\begin{aligned} x_{n+1} &= r\left(x_n - |x_n|^2\right) - r y_n, \\ y_{n+1} &= -y_n e^{-2\beta} + e^{-\beta} r\left[(2 - x_n - x_n^*) y_n - x_n z_n^* - x_n^* z_n\right], \\ z_{n+1} &= -z_n e^{-2\beta} + e^{-\beta} r\left[2(1 - x_n^*) z_n - 2 x_n y_n - x_n\right], \end{aligned} \tag{3}$$

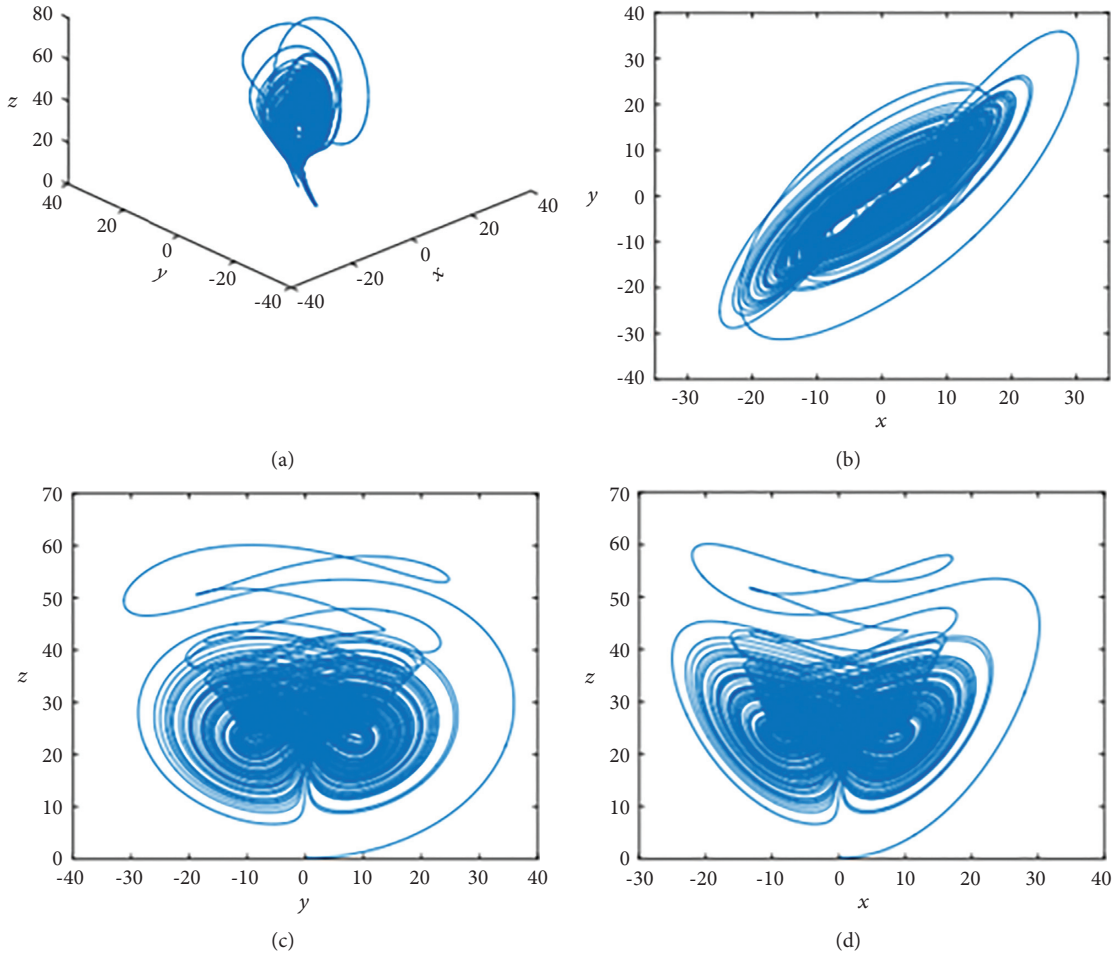


Figure 1: $x - yx - z$Figure 1Hyperchaos attractors of Chen's chaotic system: (a) Distribution in the direction of x-y-z. (b) Plane graph of x-y. (c) Plane graph of y-z. (d) Plane graph of x-z.

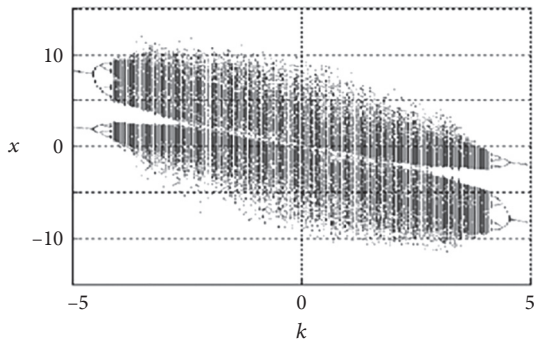


Figure 2: Bifurcation diagram of Chen's chaotic system with parameter k.

where β is dissipation parameter and γ represents control parameter. However, the initial conditions (x_n, y_n, z_n) are set as real numbers to meet the requirement of communication. Figure 3 shows the phase diagram of quantum logistic map, and its bifurcation diagram is displayed in Figure 4.

3. The Proposed Image Encryption and Decryption Scheme

3.1. Image Encryption Process. This section presents the details of the design of the proposed method based on the adopted fundamental Fridrich's permutation-diffusion model,

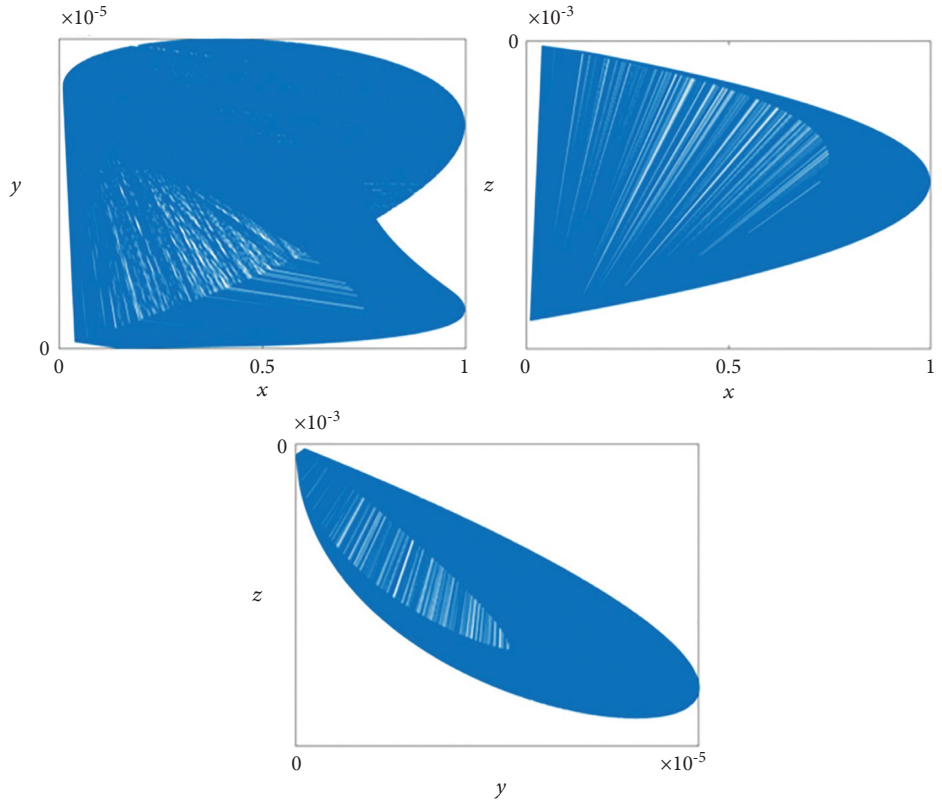


Figure 3: Phase diagram of quantum logistic map.

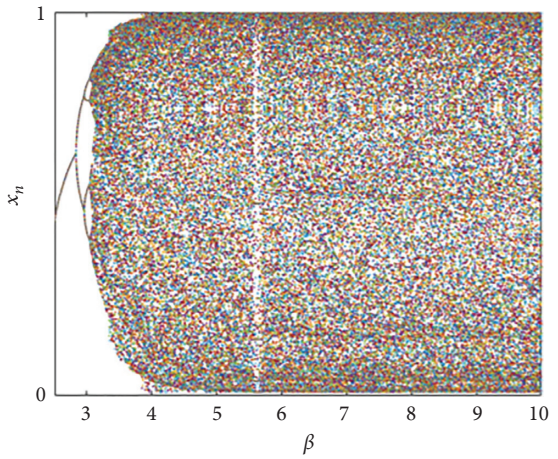


Figure 4: Bifurcation diagram of quantum logistic map.

hyperchaotic system, and a 3D quantum logistic mapping. The proposed algorithm is designed in the context of sensitive information of digital color and gray images. Consider a color image I with size $W \times H$, where W and H represent the image's rows and columns, respectively. The R, G, and B components of I are denoted as R, G, and B, respectively. The proposed framework consists of four main phases, and the details of these phases are presented as follows:

(i) The first phase is key extraction from a plain image through computing the mean of any of the four edges of the plain image and then utilizing that mean to make a number of iterations for both Chen's hyperchaotic system and the quantum logistic map in order to modify the initial seeds and control parameters for them.

(ii) Iterate continuously Chen's hyperchaotic system $W \times H$ times to generate a random sequence of integers E_I whose values range from [0...255], where the length of sequence E_I, that is, n, will be equal to the number of pixels in the image. Then split it into three chaotic sequences E_i^R, E_i^G, E_i^B which are computed using the following equations:

$$E_i^R(i,j) = \text{unit8}\Big(\text{round}\Big(\text{mod}\Big((abs(x) - \text{floor}(abs(x))) * 10^{(14)}, 256\Big)\Big)\Big),$$
$$E_i^G(i,j) = \text{unit8}\Big(\text{round}\Big(\text{mod}\Big((abs(y) - \text{floor}(abs(y))) * 10^{(14)}, 256\Big)\Big)\Big), \quad (4)$$
$$E_i^B(i,j) = \text{unit8}\Big(\text{round}\Big(\text{mod}\Big((abs(z) - \text{floor}(abs(z))) * 10^{(14)}, 256\Big)\Big)\Big),$$

where $i = 1, 2, \ldots, W \times H$.

(iii) Diffuse three components of the plain image by the random sequences generated by 3D hyperchaotic system to obtain their corresponding cipher sequences. The diffusion process is performed by implementing XOR operation as follows:

$$C_i^R = P_i^R \oplus E_i^R,$$
$$C_i^G = P_i^G \oplus E_i^G, \quad (5)$$
$$C_i^B = P_i^B \oplus E_i^B.$$

(iv) Perform quantum logistic map to produce a chaotic sequence Q_i; after that separate it into three channels Q_i^R, Q_i^G, Q_i^B which can be calculated as follows:

$$Q_i^R = \text{mod}\left(\text{floor}\left(\varepsilon_1 * x_{i+1} + \varepsilon_2\right), 256\right), \quad i = 1, 2, \ldots, W \times H,$$
$$Q_i^G = \text{mod}\left(\text{floor}\left(\varepsilon_1 * y_{i+1} + \varepsilon_2\right), 256\right), \quad i = 1, 2, \ldots, W \times H, \quad (6)$$
$$Q_i^B = \text{mod}\left(\text{floor}\left(\varepsilon_1 * z_{i+1} + \varepsilon_2\right), 256\right), \quad i = 1, 2, \ldots, W \times H,$$

where $(\varepsilon_1, \varepsilon_2)$ are two large prime numbers and $(x_{i+1}, y_{i+1}, \text{and}\, z_{i+1})$ are random sequences which are generated by 3D quantum logistic map (3).

(v) Generate density matrix H using the following equations:

$$H_{11} = (p + (1-p)) * \left(\cos\left(\frac{a}{2}\right)^2\right),$$
$$H_{12} = (1-p) * \left(\sin\left(\frac{a}{2}\right) * * \cos\left(\frac{a}{2}\right)\right),$$
$$H_{21} = (1-p) * \left(\sin\left(\frac{a}{2}\right) * \cos\left(\frac{a}{2}\right)\right), \quad (7)$$
$$H_{22} = (1-p) * \left(\sin\left(\frac{a}{2}\right)^2\right),$$

where p is probability and a is angle.

(vi) The diffused layers (C_i^R, C_i^G, C_i^B) and chaotic sequences are produced by the 3D chaotic logistic map (Q_i^R, Q_i^G, Q_i^B) which are expressed as a quantum superposition state, using the XNOR function as follows:

$$\text{SC}_i^R = \overline{C_i^R \oplus H},$$
$$\text{SC}_i^G = \overline{C_i^G \oplus H},$$
$$\text{SC}_i^B = \overline{C_i^B \oplus H},$$
$$\text{SQ}_i^B = \overline{Q_i^B \oplus H}, \quad (8)$$
$$\text{SQ}_i^B = \overline{Q_i^B \oplus H},$$
$$\text{SQ}_i^B = \overline{Q_i^B \oplus H},$$

where operator $\overline{\oplus}$ denotes bitwise exclusive NOR.

(vii) Finally, the final cipher channels $\text{FC}_i^R, \text{FC}_i^G, \text{FC}_i^B$ are obtained by applying XNOR function on both a unitary matrix generated by logistic chaos $(\text{SQ}_i^R, \text{SQ}_i^G, \text{SQ}_i^B)$ and the diffused components $(\text{SC}_i^R, \text{SC}_i^G, \text{SC}_i^B)$ generated density matrix to confuse and diffuse pixels simultaneously, which can be expressed as follows:

$$\text{FC}_i^R = \overline{\text{SC}_i^R \oplus \text{SQ}_i^R},$$
$$\text{FC}_i^G = \overline{\text{SC}_i^G \oplus \text{SQ}_i^G}, \quad (9)$$
$$\text{FC}_i^B = \overline{\text{SC}_i^B \oplus \text{SQ}_i^B}.$$

(viii) Combine (FC_i^R, FC_i^G, and FC_i^B) into a chaotic matrix FC_i with transpose rows and columns of the border of the image to get the final cipher image $C_{W\times H}$.

The sketch of the proposed encryption scheme is exhibited in Figure 5 with a succinct explanation of each phase presented herewith while the specific implementation process of the proposed image encryption scheme is presented in Algorithm 1.

3.2. Decryption Method. The architecture of the proposed decryption algorithm is shown in Figure 6, which is applied on a cipher image to produce a plain image.

4. Experimental Results and Numerical Analysis

Due to the absence of a practical and functional quantum computer, the experimental results are performed with MATLAB R2017b platform on a classical computer to verify the security and effectiveness of the proposed quantum image encryption algorithm. The operation system used is Windows 10 Professional operating system with the specific configuration being i7-8550U applied as the central processing unit (CPU) and the random-access memory (RAM) adopted is 8 GB.

For simulation, the control parameters and initial values of Chen's hyperchaotic system, given in (2), are set as $a = 36$, $b = 3$, $c = 28$, and $d = -16$, and $x_0 = 0.3$, $y_0 = -0.4$, $z_0 = 1.2$, and $q = 1$, we carry out the encryption scheme. The keys for this proposed cryptosystem include the iteration times of Chen's hyperchaotic system and quantum logistic chaotic map M, where the discarded number M is set according to the mean of plain image. For color images, the encryption key is the same in RGB channels.

To demonstrate the practical benefits of the proposed image encryption scheme, a number of experiments were performed based on the USC-SIPI (the University of Southern California Signal and Image Processing Institute) Image Database [32]. This database is divided into four groups of images: Textures (64 images), Aerials (38 images), Miscellaneous (39 images), and Sequences (69 images). Each group contains images of various sizes $m \times m$, $m = 256$, 512, 1024. Different sample images (gray and color) are chosen as test images from the USC-SIPI "Miscellaneous" dataset and the simulation results of these encryption and decryption images are presented in Figure 7, where the plain images of "Aerial," "Boat," "Male," "Airplane," "Lena," and "Baboon" are shown in Figures 7(a)–7(f), their corresponding cipher images are shown in Figures 7(g)–7(l), and the recovery images from decryption process with correct secret keys are shown in Figures 7(m)–7(r) which are identical to the original images, and their detailed information is listed in Table 1.

As illustrated in Figures 7(g)–7(l) that the proposed encryption scheme can encrypt different size images, besides that it destroys the obvious pattern of the plain image and makes the ciphered image display a space filling with a noise-like pattern which makes the ciphered image seem random to the intruder. Therefore, the proposed encryption algorithm has good encryption and decryption effect; it can attain the image data security and appearance security. The quantitative performance of the newly resulted image encryption algorithm could be measured through different evaluation parameters, including statistical, differential, sensitivity, and key space metrics. Each of these measures is discussed in detail in the accompanying subsections.

4.1. Key Space Analysis. The key space of a cryptosystem is the very important factor on security when brute-force attack is happening. For high-security cryptosystem, it should be highly sensitive to a tiny change in the cryptographic keys and the key space is suggested to be much larger than 2^{100} to resist exhaustive attack effectively [33–36]. Moreover, the keys should be easy to establish and exchange for practical communication. The key space is the total number of different keys that can be used in the encryption/decryption procedure. According to the algorithm structure, the secret key format should consist of the following: (1) The parameters of Chen's hyperchaotic system are a, b, c, d, and k and each of the original variables (x_1, x_2, x_3, x_4) has 2 decimal places; there exist 10^2 possible values for each value. This contributes to 6 possible guesses of value. This applies to (a, b, c, d, k, x_1, x_2, x_3, x_4) as well. Thus, there are 2^{54} possible values of (a, b, c, d, k, x_1, x_2, x_3, x_4). (2) The initial values of hyperchaotic system (x_1, x_2, x_3) are obtained by iterating system; each has 14 decimal places with the range between 0 and 1, and there exist 10^{14} possible values for each value. This contributes to $2^{46.5}$ possible guesses of value. This applies to (x_1, x_2, x_3) as well. Thus, there are $2^{139.5}$ possible values of (x_1, x_2, x_3). (3) Parameters β and r are used in the quantum logistic chaotic map, where β consists of 4 decimal places; there exist 10^4 possible values for each value. This contributes to 2^{12} possible guesses of its value and r consists of 2 decimal places, and there exist 10^2 possible values for each value. This contributes to 2^6 possible guesses of its value. Thus, there are 2^{18} possible values of β and r. (4) Each initial value of quantum map consists of 12 decimal places with the range between 0 and 1; there exist 10^{12} possible values for each value. This contributes to 2^{40} possible guesses of value. This applies to (x_0, y_0, z_0) as well. Thus, there are 2^{120} possible values of x_0, y_0, and z_0. (5) Two large prime numbers are of 8 decimal places with the range between 0 and 1; there exist 10^8 possible values for each value. This contributes to 2^{26} possible guesses of value. This applies to $(\varepsilon_1, \varepsilon_2)$ as well. Thus, there are 2^{52} possible values of $(\varepsilon_1, \varepsilon_2)$. (6) Density matrix has probability p and an angle a, where p has only one decimal place with the range between 0 and 1, and a has 2 decimal places; thus there exist 10^1 possible values for p; this contributes to 2^3 possible guesses of value, whereas there exist 10^2 possible values for a; this contributes to 2^6 possible guesses of value. This contributes to 2^{10} possible guesses of value. Thus, there are 2^9 possible values of (p, a).

Consequently, the overall key space of the proposed image encryption scheme is

$$\begin{aligned}\text{TOTAL KEY SPACE} &= 2^{54} * 2^{18} * 2^{139.5} * 2^{120} * 2^{52} * 2^{9} \\ &= 2^{54+139.5+120+52+20} \\ &= 2^{392.5}.\end{aligned} \tag{10}$$

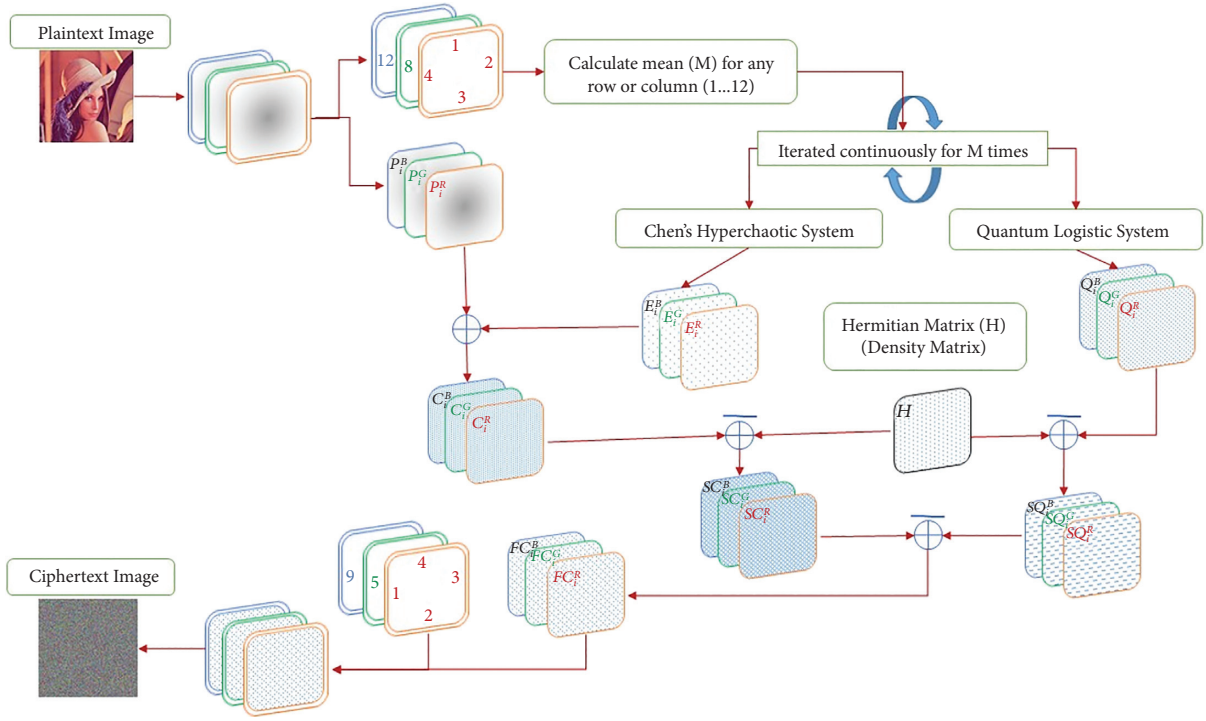


Figure 5: Block diagram of the proposed image encryption algorithm.

Input: Plain Image P of size $W \times H$, initial conditions and control parameters for hyperchaotic system (3D Chen's system), and seeds for the chaotic generator.
Output: Cipher Image C of size $W \times H$
Step 1: Plain image P is resized to a dimension of $((W-2) \times (H-2))$ pixels and is stored as P_2, and compute the mean M of any of the edges of the plain image P.
Step 2: Iterate both Chen's hyperchaotic system (equation (2)) and quantum logistic map (equation (3)) M times according to the computed mean M.
Step 3: Generate three chaotic sequences E_i^R, E_i^G, E_i^B by using a hyperchaotic system with given parameters and initial state values as secret keys.
Step 4: Separate each of the color pixel $P_i \in P_2$ of the resized image P_2 into its three grayscale components of P_i^R, P_i^G, P_i^B, then apply XOR function between three components P_i^R, P_i^G, P_i^B of the resized image P_2 and three chaotic sequences E_i^R, E_i^G, E_i^B produced by chen's hyperchaotic system. The result is considered as diffused R, G, and B components, which are C_i^R, C_i^G, C_i^B.
Step 5: Quantum logistic map is initiated and utilized to generate a chaotic keystream sequence Q_i, after that split it into R, G, and B components Q_i^R, Q_i^G, Q_i^B.
Step 6: Generate Density matrix which is described as Hermitian matrix $H_{W-2\times H-2}$.
Step 7: Employ Density matrix on the diffusion components (C_i^R, C_i^G, C_i^B), as well as the output of quantum logistic map (Q_i^R, Q_i^G, Q_i^B) using XNOR function to put each of them in a superposition environment.
Step 8: The three components of the cipher image FC_i^R, FC_i^G, FC_i^B are generated by XNORing the output of applying density matrix on the diffused components (SC_i^R, SC_i^G, SC_i^B), and quantum logistic map (SQ_i^R, SQ_i^G, SQ_i^B).
Step 9: Take transpose of the edges of the plain image P in order to increase the randomness within the plain image by shuffling the pixels.
Step 10: Recombine the cipher image FC with the shuffled edges of the plain image P to obtain the final cipher image C.

Algorithm 1: Image encryption method.

As a result, the key space is reasonably large enough for the cryptosystem to withstand exhaustive attacks and even quantum computer attacks. Table 2 shows the key space comparison of similar recent algorithms. Obviously, the proposed encryption algorithm has larger key compared to the existing works [4, 15, 24, 35, 37], which is sufficiently large to resist all presently known brute-force attacks.

4.2. Key Sensitivity Analysis. To resist violent attacks, a password system should be highly sensitive. Hence, key sensitivity [37–40] is an important index to measure the strength of encryption algorithm. The key sensitivity of an image cryptosystem can be evaluated in two aspects: First, the cipher image will be completely different when encrypting the same plain image with slightly different keys,

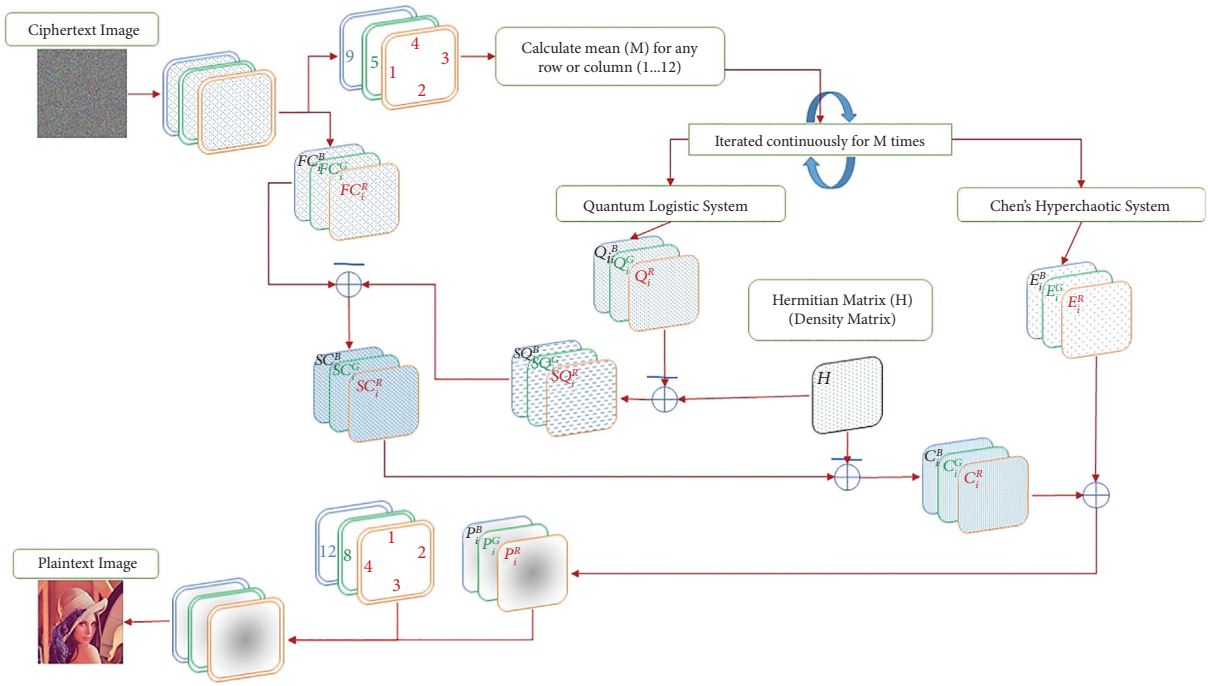


Figure 6: Block diagram of the proposed image decryption algorithm.

Input: Cipher Image C of size $W \times H$
Output: Decrypted Image P of size $W \times H$
Steps: Inverse steps of image encryption routine are carried out in the reverse order using the same encryption keys.

Algorithm 2: Image decryption method.

which is measured by the change rate t of the cipher image. Second, a small change in the decryption key makes a huge difference to the result, and the original image will not be decrypted correctly, indicating that the algorithm has a high sensitivity. The Lena color image with size 512×512 is utilized to verify the sensitivity of the suggested image encryption scheme. During the test process, one of the keys has undergone a tiny change, while other keys were kept untouched. Suppose that K_1 and K_2 are the two keys that are slightly different from each other, which gives encrypted outputs of E_1 and E_2, respectively, where K_1 is the correct key and K_2 is the wrong one. In the proposed cryptosystem, the control parameters of Chen's hyperchaos system are set as $a = 36$, $b = 3$, $c = 28$, and $d = -16$, and the initial values of the system are $x_0 = 0.3$, $y_0 = -0.4$, $z_0 = 1.2$, and $q = 1$, which are denoted as K_1, to obtain encrypted image E_1. Another encrypted image E_2 is generated with a tiny change in only x_0 ($x_0 = 0.4$, $y_0 = -0.4$, $z_0 = 1.2$, and $q = 1$), which are denoted as K_2. As shown in Figures 8(b) and 8(c), the image encrypted using K_1 is completely different from the image encrypted using K_2. From the result, as shown in Figures 8(e) and 8(f), it is clear that decryption of the encrypted image is possible only when we use the same key. Therefore, it can be seen that only a subtle difference in the secret key can have a huge effect which guarantees the security against brute-force attacks and known plain-text attacks.

4.3. Statistical Attack Analysis. To verify the security performance of the proposed algorithm, the statistical analyses including histogram, correlation, and entropy analysis are demonstrated in this subsection.

4.3.1. Histogram Statistical Analysis. Histogram statistical analysis is a kind of statistical attack, and the histogram can characterize the image. It has been widely used in image retrieval, classification, and other fields [41–47]. Image histogram is probability density function of discrete gray level, plotted by gray level on horizontal axis and the corresponding frequency on the vertical. The more uniform the histogram distribution for the encrypted image, the stronger the ability of antistatistical analysis. Therefore, the elimination of correlation among pixels was necessary, and pixels of the encrypted image had to be distributed evenly to prevent the opponent from extracting any useful information from the fluctuating histogram. In addition, comparing cipher image histogram with the original image histogram, there is a significant difference.

We have analyzed the histograms of two original images as well as their encryptions using the proposed approach. The histogram of the original grayscale image of "Boat" with dimensions 512×512 pixels and the histogram of its cipher image are shown in Figure 9, while Figure 10 illustrates the

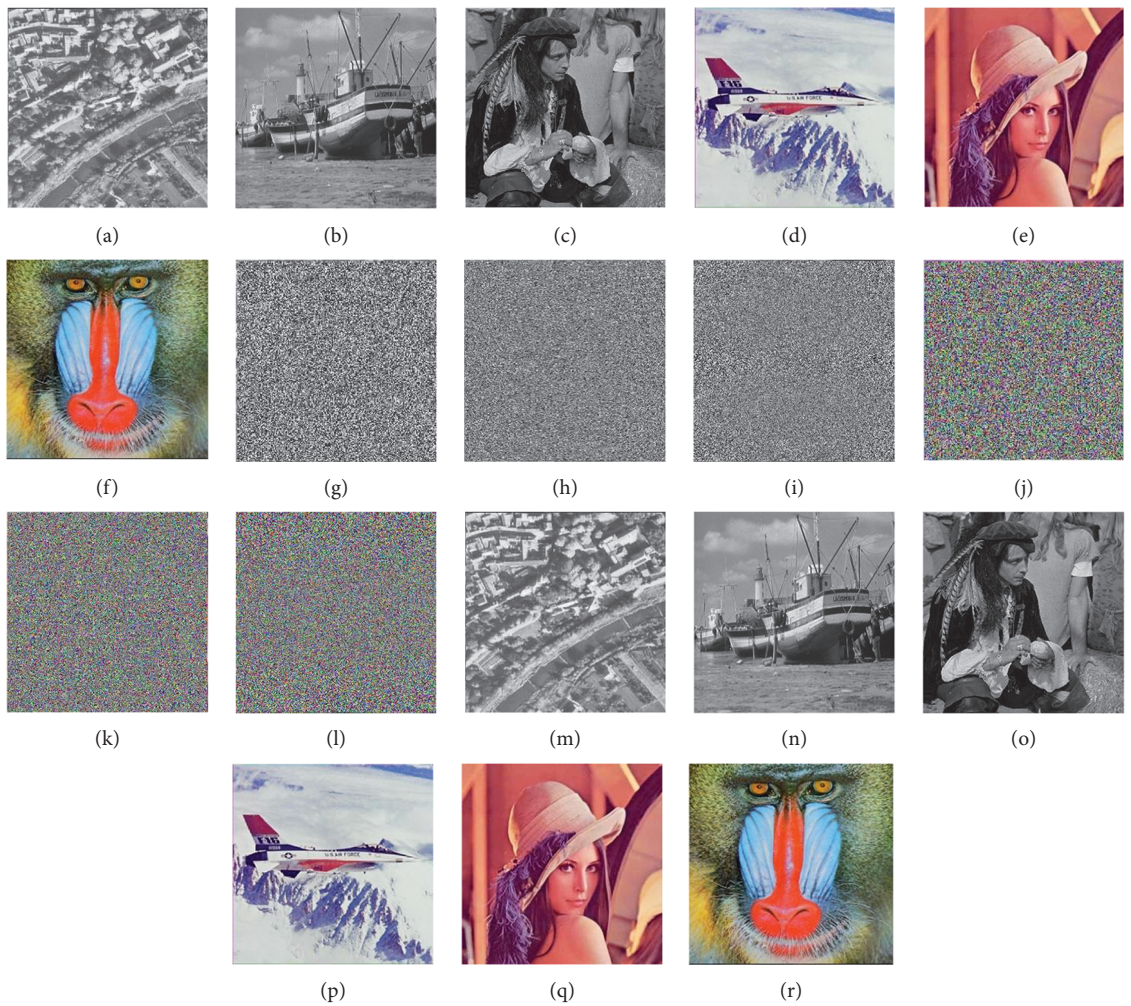


Figure 7: Encryption and decryption results: ((a)–(f)) plain images of "Aerial," "Boat," "Male," "Airplane," "Lena," and "Baboon"; ((g)–(l)) the corresponding encrypted images; and ((m)–(r)) decrypted images.

Table 1: Selected test images.

Image	Aerial	Boat	Male	Airplane	Lena	Baboon
Size	256×256	512×512	1024×1024	256×256	512×512	1024×1024
Type	Grayscale	Grayscale	Grayscale	Color	Color	Color

Table 2: Key space comparative analysis.

Encryption scheme	Key space
Ref. [4]	2^{256}
Ref. [15]	2^{125}
Ref. [24]	$10^{15\times3} + 2^{256}$
Ref. [35]	2^{186}
Ref. [37]	2^{364}
Proposed algorithm	$2^{392.5}$

histograms of the R, G, and B channels of the color plain image "Lena" alongside its encrypted counterparts with the size 512×512, respectively.

Clearly, it can be seen from Figures 9 and 10 that the histograms of the original images have obvious peaks, and the gray value and RGB component histogram of cipher images are very uniform and flat distribution, which indicates that the attack based on histogram analysis is difficult as attackers cannot use a statistical attack to obtain any useful information by analyzing the histogram of the encrypted image. Thus, the proposed scheme is strong enough to withstand statistical attacks.

Consequently, it is concluded that the proposed image encryption scheme can achieve good performance and meet the requirements of image encryption.

Furthermore, for quantity analyses of the image histogram, a metric called variance of the histogram (var) is measured to evaluate and guarantee the uniformity of pixels values of the encrypted images. The higher the uniformity of ciphered images, the lower the value of variances of

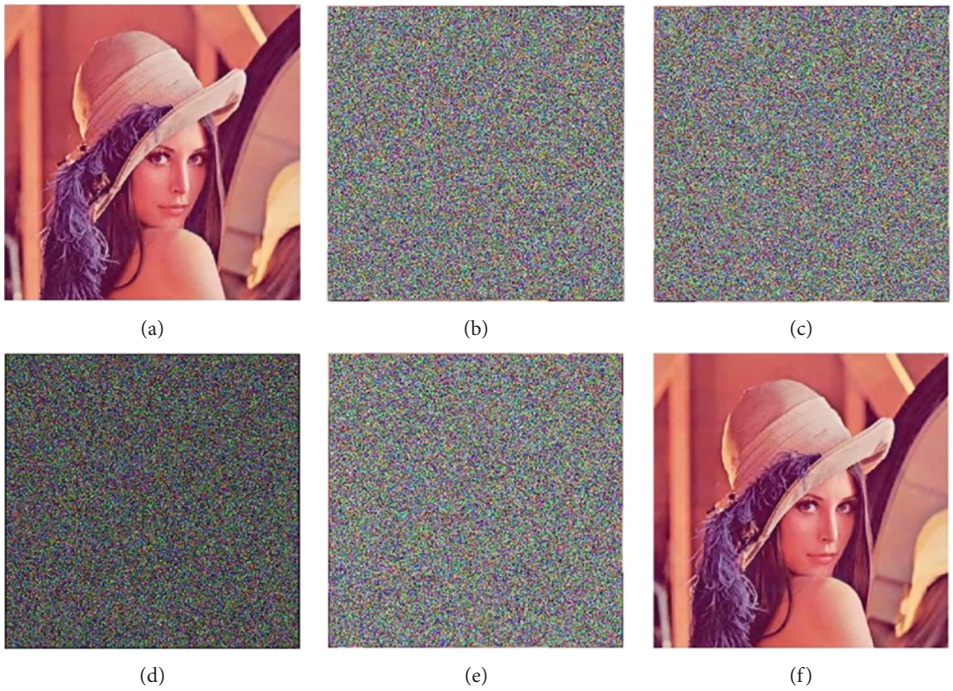


Figure 8: Key sensitivity analysis. (a) Plain image, (b) correctly encrypted image (E_1), (c) incorrectly encrypted image (E_2), (d) difference of E_1 and E_2, (e) incorrectly decrypted image, and (f) correctly decrypted image.

histogram [48]. The variance of histogram can be computed as follows:

$$\operatorname{var}(H) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}\left(h_i - h_j\right)^2, \tag{11}$$

where $H = \{h_1, h_2, \ldots, h_{256}\}$ is a one-dimensional array of the histogram values; h_i and h_j are considered as the numbers of pixels where gray values are equal to i and j, respectively. Tables 3 and 4 display the values of histogram variance for the experimented grayscale and color images, respectively, and illustrate that the variance of images after encryption is greatly reduced when compared with the variance of those images prior to encryption.

The simulation results indicate that the difference in variance value shows that the histogram depends on the plain image; in addition, the proposed algorithm can strongly withstand statistical analysis attack as it is efficient to prevent attackers from obtaining any useful statistical information to decrypt the cipher image.

4.3.2. Correlation Coefficient Analysis. It is known that some algorithm was broken by using correlation analysis between the adjacent pixels. So, correlation coefficient analysis [49–52] is performed to evaluate the statistical relationship between image pixels, and its value is in the range of [−1, 1]. This type of analysis visually shows the distribution between the neighborhood pixels of both the original and encrypted images.

Due to the intrinsic features of the digital image [53], there is a strong correlation between the adjacent pixels, namely, the gray value of one pixel of the plaintext image is very close to the gray value of the surrounding pixels. Therefore, attackers could try to infer adjacent pixel values based on probability theory. Conversely, in order to resist the statistical attack and achieve better security of the encrypted image, an excellent image encryption algorithm should be able to break high correlation between adjacent pixels of the plain image and produce a very small correlation value near the optimal value of zero.

Normally, three different types of correlation are performed to ensure the strength of the encrypted image: the horizontal, the vertical, and the diagonal correlation [54]. To evaluate the proposed encryption scheme, 3000 pairs of adjacent pixels are selected randomly in the three different adjacent directions in both original and encrypted images of the different sample images to calculate the correlation coefficient. Then, the correlation coefficient r_{xy} of each pair, defined in (12), is calculated as follows:

$$\begin{cases} r_{xy} = \dfrac{\operatorname{cov}(x,y)}{\sqrt{D(x)} \times \sqrt{D(y)}}, \\ \operatorname{cov}(x,y) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ D(x) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}\left(x_i - E(x)\right)^2, \\ E(x) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} x_i, \end{cases} \tag{12}$$

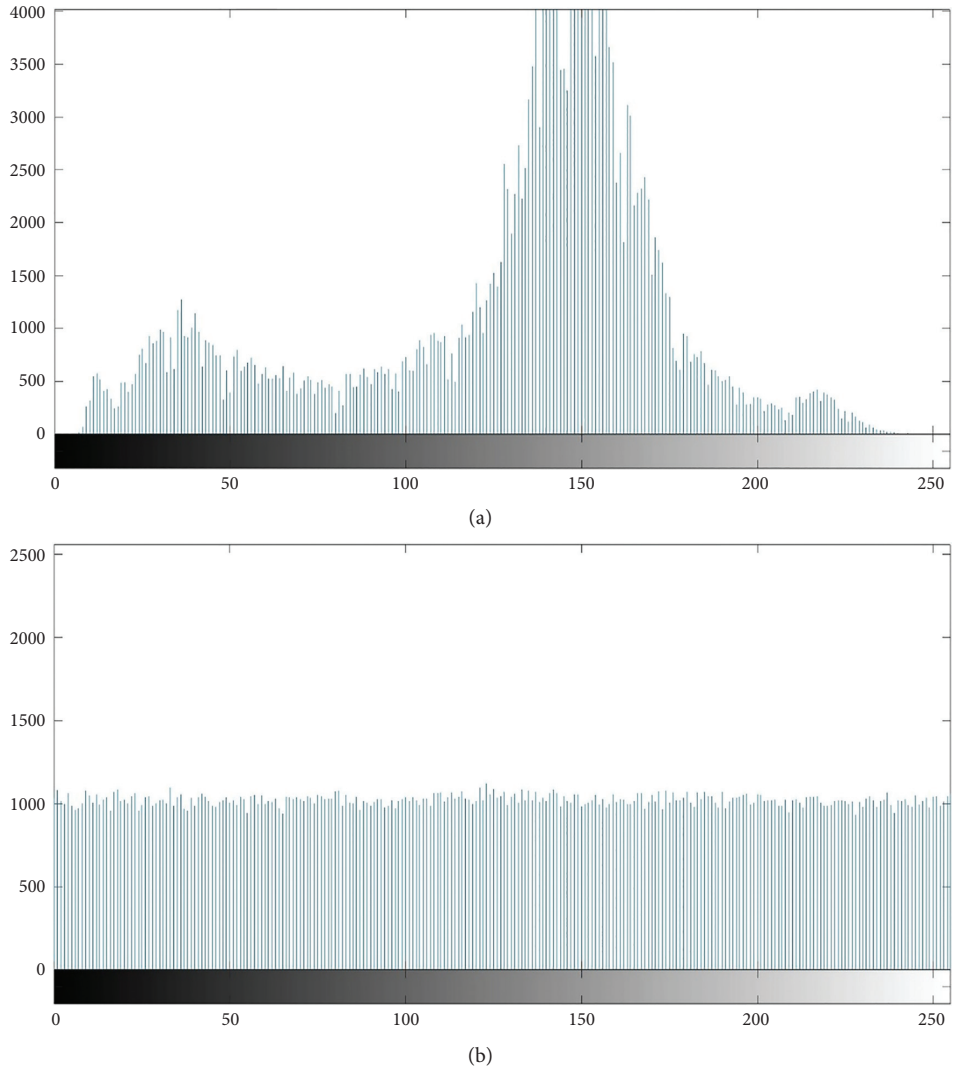


Figure 9: Histogram distribution analysis of plain and encrypted grayscale image. (a) Plain grayscale "Boat" image; (b) encrypted grayscale "Boat" image.

where x_i and y_i are the grayscale pixel values of the ith pair of the selected adjacent pixels in the tested image, N is the total number of the randomized chosen samples, $\mathrm{cov}(x, y)$ is the covariance of x and y, and $E(x)$ and $D(x)$ represent the mean value and the variance of vector x, respectively.

Figures 11 and 12 show the correlation distribution between neighborhood pixels in the three directions of the grayscale "Boat" image and color "Lena" image with size 512×512 before and after image encryption. It is obvious that the correlation of adjacent pixel pairs of the plain image is distributed intensively, but those of encrypted image are scattered randomly which looks very uniform, and the correlation is greatly reduced.

Numerically, Table 5 demonstrates values of correlation coefficient parameter for the proposed technique in different test images with diverse sizes. According to the quantitative results, it can be concluded that the correlation degrees between adjacent pixels in the plain images are close to 1, while those of the encrypted images are very small and are close to 0, which means that the plain image has strong relationships, but weakness exists in the encrypted image. Therefore, these results show that the proposed image encryption scheme has a good performance in fighting against attacks based on statistical properties of the images.

4.3.3. Information Entropy Analysis. Information entropy [55–57] is the most important criterion to evaluate the efficiency of an image encryption algorithm. In information theory, the entropy parameter is considered as the standard to test randomness. For a digital image, information entropy (IE) is one of the outstanding criteria that is usually utilized to evaluate the degree of disorder or randomness of each gray value in the encrypted image and measure the amount of information hidden in an image. The color-level distribution values in an image can also be determined via entropy analysis. Ideally, in the case of 8-bit grayscale image, a robust encryption scheme has an entropy value of 8; otherwise, it

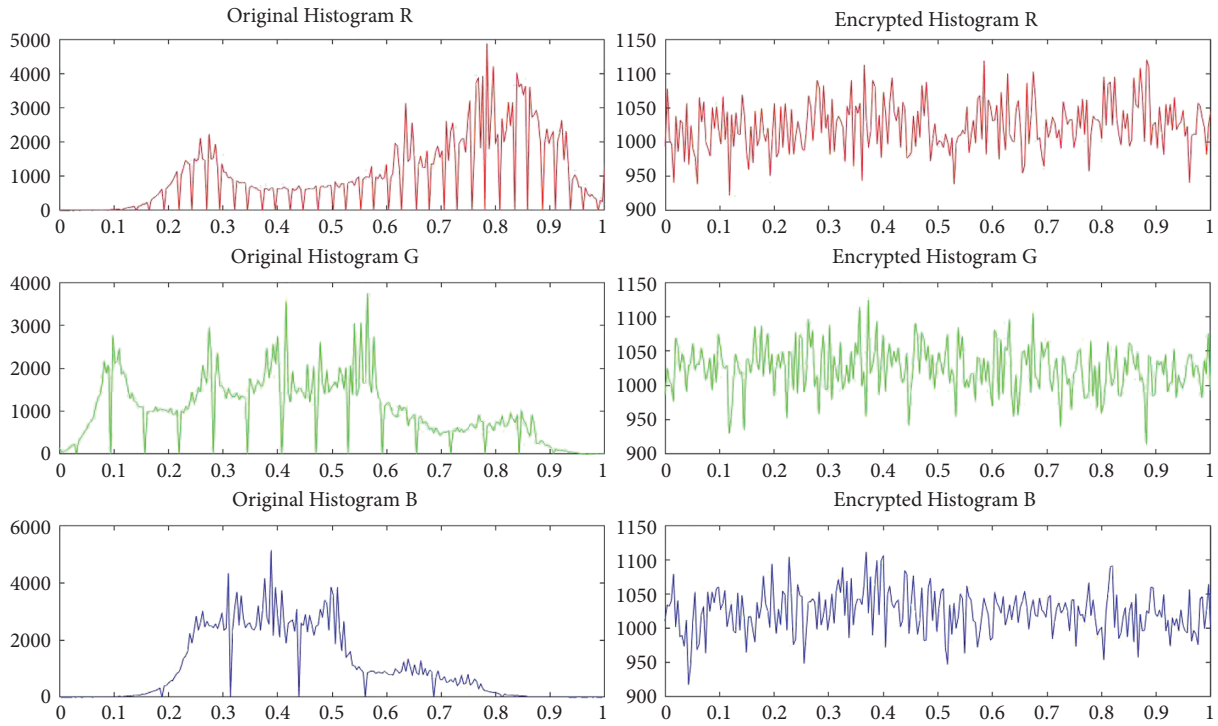


Figure 10: RGB color histogram of original "Lena" image and its corresponding cipher image.

Table 3: Variance of histogram for encrypted grayscale images.

Image name	Variance value	
	Plain image	Cipher image
Aerial	51062	780.8235
Boat	1541901.8039	9791.7098
Male	11393958.6980	138136.8627

Table 4: Variance of histogram for encrypted color images.

Image name	Variance value					
	Plain image			Cipher image		
	R	G	B	R	G	B
Airplane	165621.8980	163801.6941	274155.3333	783.1607	765.4039	796.7137
Lena	1021383.0980	457505.9372	1382757.2627	8929.4431	8930.2823	9570.1019
Baboon	6346579.1843	10106060.6980	5938608.9490	133133.6470	135706.9490	136423.2078

causes a plausibility of consistency which undermines its security. The closer the value is to 8, the greater the uncertainty is and the stronger the randomness of image is, which leads to better-secured encryption where the less visual information can be obtained from the image. The most famous entropy formula is Shannon's entropy equation, calculated in terms of the probability of each available data value, which can be defined as follows:

$$\mathrm{IE} = -\sum_{i=0}^{255} P(i)\log_2 P(i), \tag{13}$$

where $P(i)$ denotes the probability of occurrence of gray level i in an image, that is, the proportion of the number of pixels with gray value i to all pixels in an image. Besides, to verify the randomness, local Shannon entropy should be applied. It can be calculated by the following operations: ① divide the image into noninterlocked K blocks containing a certain fixed number of pixels; ② compute Shannon entropy $\mathrm{IE}(K_i)$ using the former equation (12); ③ calculate the sample mean of global Shannon entropy over all these K image blocks as local Shannon entropy.

Table 6 presents the simulation results of information entropy and local Shannon entropy values, where $K = 16$, on some standard original images and their respective encrypted images, which were encrypted by the proposed image encryption algorithm. It can be seen that the results reveal that the entropy values of each cipher image are very close to the ideal value of 8, while the information entropy of

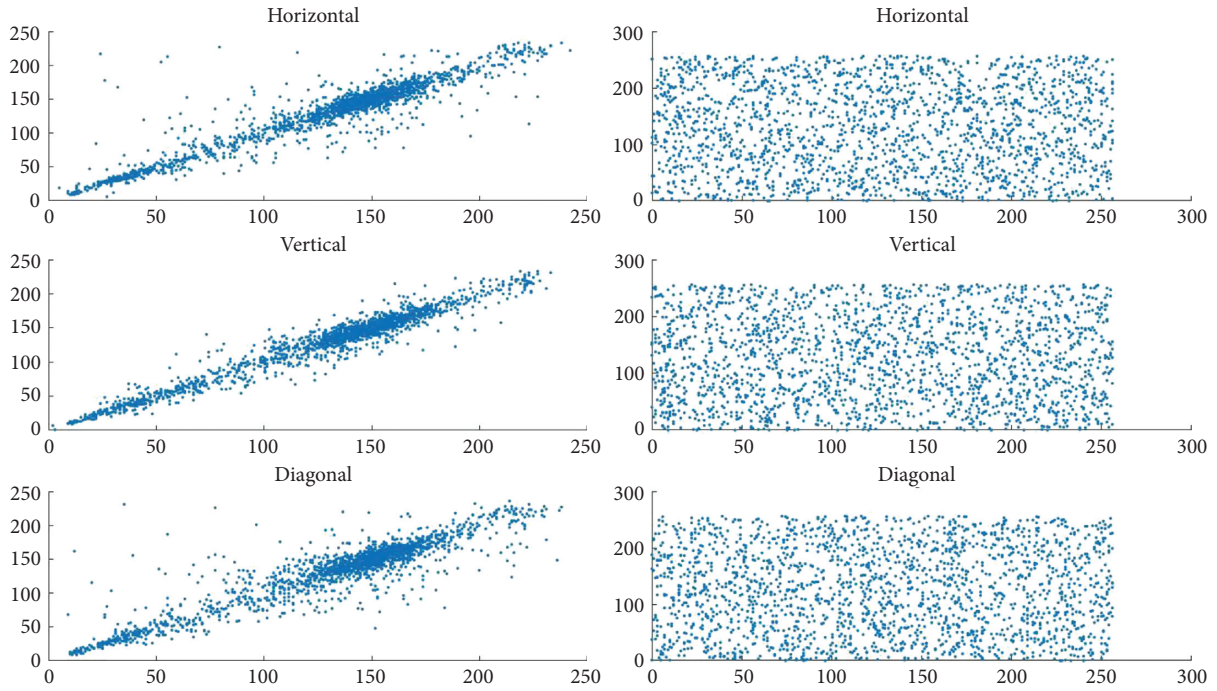


FIGURE 11: Adjacent pixel correlation test for plain grayscale "Boat" image (a) and the corresponding encrypted image (b) for horizontal, vertical, and diagonal directions.

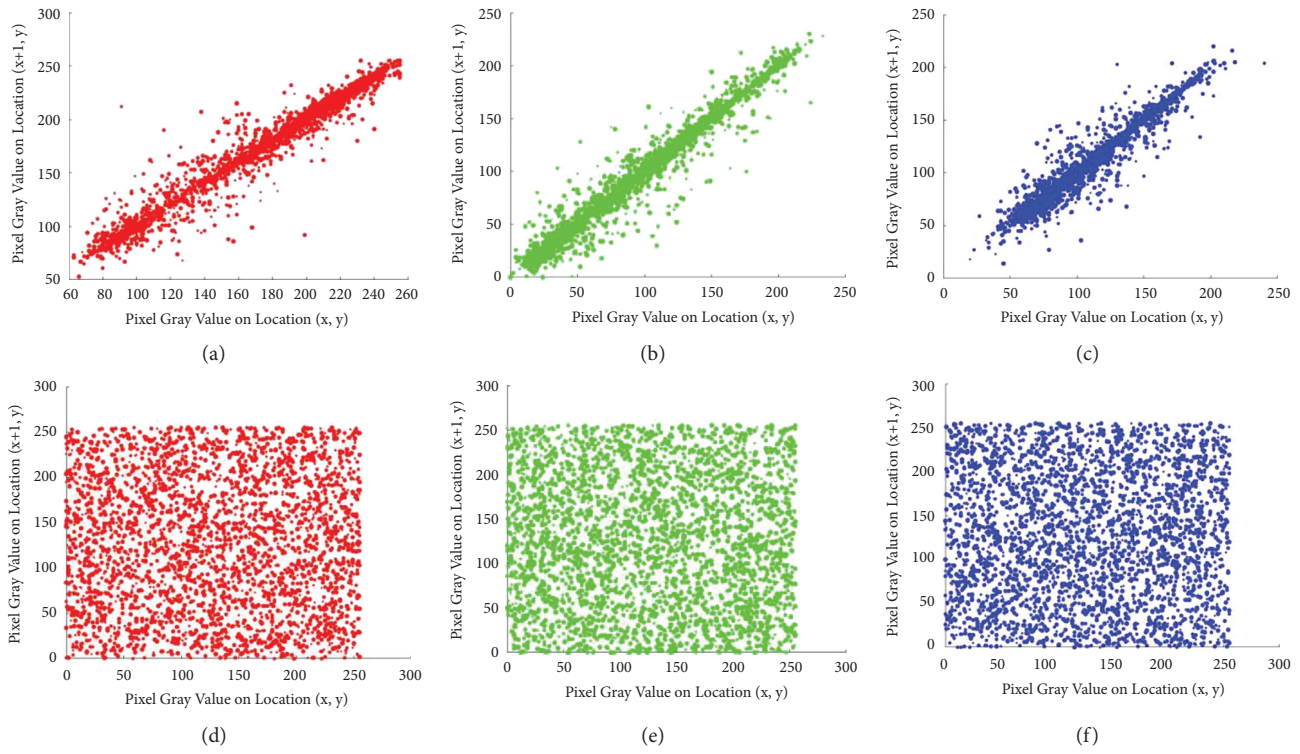


FIGURE 12: Correlation distribution of color "Lena" image: (a)–(c) show RGB layers of plain image; (d)–(f) show RGB layers of cipher image for vertical, horizontal, and diagonal directions, respectively.

each plain image is much less than the ideal one. This result makes obtaining image information by analyzing this information difficult for attackers. This indicates that the encrypted images have a good randomness. As a conclusion, the proposed scheme is safe against the perspective of information entropy attack.

Table 5: Correlation coefficients results.

Image name	Plain image			Cipher image		
	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal
Aerial	0.8602	0.9050	0.8213	0.0062	−0.0017	0.0031
Boat	0.9713	0.9381	0.9222	−0.0012	0.0007	0.0004
Male	0.9813	0.9774	0.9671	0.0001	0.0002	0.0023
Airplane	0.9174	0.9314	0.8643	0.0061	0.0109	0.0012
Lena	0.9902	0.9804	0.9695	0.0040	−0.0003	−0.0012
Baboon	0.9765	0.9877	0.9671	0.0023	0.0012	0.0001

Table 7 presents the values of information entropy of the proposed scheme as compared with the values which resulted from other recent schemes. It can be seen that the information entropy of the different cipher images is very close to 8 bits and the proposed algorithm has greater superiority or in the same range.

4.4. Differential Attack Analysis. Differential attacks are another effective and commonly used cryptanalysis technique. A differential attack is attempted to learn the key and figure out the encryption scheme by tracing differences. An Assailant may make a trivial change in the plain image, encrypt two plain images, and then carry out cryptanalysis by tracing the meaningful relationship between two cipher images. According to the principles of cryptography, the encryption algorithm should be sufficiently sensitive to the changes of plaintext image or secrete key in order to keep high security, such that a minor change in the plaintext image or the initial key parameters causes a significant change in the ciphertext image [59–64]; then differential analysis may become useless. The high sensitivity of the system shows that the generated algorithm is sturdy against any probable attack, since it would indicate no meaningful relationship between the plain image and the cipher image. In this test, the number of pixels changing rate (NPCR) and unified average changing intensity (UACI) become two widely used security analyses in the image encryption community for differential attacks. The tests signify the chance of occurrence of the attack and its sensitivity towards the source image by changing the value.

Considering C_1 and C_2 as the two cipher images obtained from encrypting two one-pixel different images with $M \times N$ size or encrypting same plain image with two secret keys of only 1-bit difference, introduce a bipolar array, D, with the sizes similar to images C_1 and C_2 as follows:

$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j), \\ 1, & C_1(i,j) \neq C_2(i,j). \end{cases} \tag{14}$$

The NPCR reflects the change rate of the gray value of different pixels at the same position between two corresponding encrypted images which are obtained by two original images with one-bit difference. In other words, NPCR helps us to understand the effect of change of single pixel over an image, while the UACI reflects the average change of the gray value within the two paired cipher images (C_1 and C_2). Then, the formula used to calculate UACI and NPCR is shown in the two following equations:

$$\text{NPCR} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\%, \tag{15}$$

$$\text{UACI} = \frac{1}{M \times N}\left(\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left|C_1(i,j) - C_2(i,j)\right|}{255}\right) \times 100\%. \tag{16}$$

Taking the images that are listed in Table 1 as examples and experimenting on them for 100 times, the theoretical ideal values of the NPCR and UACI for a gray image are 99.6094% and 33.4635%, respectively. Table 8 lists the test results of NPCR and UACI of grayscale encrypted images, whereas the theoretical values of NPCR and UACI for different color images in three channels are shown in Table 9. It can be observed from the former tables that the proposed image encryption algorithm can achieve better performances against differential attacks, since the values of NPCR and UACI are close to their theoretical values. Thus, the system has guaranteed that the designed system is applicable for real-time communication.

4.5. Known-Plaintext Attack and Chosen-Plaintext Attack Analysis. A cryptosystem is supposed to be secure if it resists all known types of cryptographic attacks. In cryptanalysis, the fundamental assumption enunciated by Kerckhoffs's principle is that encryption and decryption algorithms are known or transparent in a cryptosystem [65–68]. Therefore, the security of the cryptosystem depends on the key rather than the encryption algorithm itself. In the cryptanalysis, there are four traditional cryptanalysis attacks: (1) ciphertext-only attack, (2) known-plaintext attack, (3) chosen-plaintext attack, and (4) chosen-ciphertext attack. Among these attacks, chosen-plaintext attack is the most threatening attack. Therefore, it is claimed that the cryptosystem can resist the other three types of attacks if it can resist the chosen-plaintext attack. In order to assess the resistance of encryption algorithms against the main attacks, two tests are generally used, namely, the known-plaintext attack (KPA) and chosen-plaintext attack (CPA). In known-plaintext attack and chosen-plaintext attack, the attackers usually choose special plaintext and make minor changes to observe the changes of ciphertext. Or they choose some plaintext with linear relationship to observe the characteristics of ciphertext. By using this method, they can obtain secret key. By using this method, they can obtain secret key.

Table 6: Results of information entropy and local Shannon entropy.

Image name	Dimension	Information entropy		Local Shannon entropy	
		Plain image	Cipher image	Plain image	Cipher image
Aerial	256×256	3.3556	7.9970	3.2893	7.9556
Boat	512×512	3.3153	7.9993	3.1037	7.9880
Male	1024×1024	3.3540	7.9998	3.2127	7.9971
Airplane	$256 \times 256 \times 3$	6.6906	7.9987	6.1280	7.9834
Lena	$512 \times 512 \times 3$	7.7495	7.9997	7.3136	7.9959
Baboon	$1024 \times 1024 \times 3$	7.7208	7.9999	7.4281	7.9990

Table 7: Information entropy comparison.

Image name	Dimension	Information entropy				
		Proposed Scheme	Ref. [6]	Ref. [39]	Ref. [58]	Ref. [59]
Aerial	256×256	7.9970	—	—	7.9024	—
Boat	512×512	7.9993	—	7.9993	—	—
Male	1024×1024	7.9998	—	7.9998	—	—
Airplane	$512 \times 512 \times 3$	7.9997	—	—	—	7.9994
Lena	$512 \times 512 \times 3$	7.9997	7.9988	—	—	7.9994
Baboon	$512 \times 512 \times 3$	7.9997	—	—	—	7.9993

Table 8: NPCR and UACI results for cipher grayscale images.

Image name	Image size	NPCR (%)	UACI (%)
Aerial	256×256	99.6458	33.5243
Boat	512×512	99.6517	33.4416
Male	1024×1024	99.6300	33.4864

Table 9: NPCR and UACI scores on color images of different sizes.

Image name	Dimension	NPCR (%)			UACI (%)		
		Red	Green	Blue	Red	Green	Blue
Airplane	$256 \times 256 \times 3$	99.6892	99.6380	99.64113	33.3572	33.5681	33.6369
Lena	$512 \times 512 \times 3$	99.6394	99.6417	99.6574	33.4980	33.4593	33.5460
Baboon	$1024 \times 1024 \times 3$	99.6241	99.6168	99.6238	33.4607	33.4702	33.4707

In the presented encryption scheme, the mean (M) value of the plaintext image is computed to generate the number of preiterations, which is related chaotic sequences generation, and the initial value of diffusion process. In other words, the generated random sequences are related to the plaintext, and the chaotic systems are sensitive to the initial value. Consequently, the keystream used in the proposed algorithm has a high connection with the plain image, which means that a small change in the plaintext image produces a completely different key, as detailed in the "Key Sensitivity Analysis" section. That means the attacker cannot extract any useful information by encrypting certain selected images because the encrypted image is only relevant to the selected image, which implies the excellent performance in withstanding the known-plaintext attack and chosen-plaintext attack.

Besides, to test the ability of defending this kind of attack, both plain images with "pure white" and "pure black" images, their encrypted images, and the corresponding histograms are derived, which are shown in Figure 13. From the results, it can be seen that the pixels in the cipher image are uniformly distributed with random noise, and the attacker cannot decrypt other cipher images by using the same keys. By observing the resulting encrypted images, we can find that it is impossible to extract any information from the encrypted images. Therefore, the proposed encryption scheme is sufficiently robust to resist all forms of potential attacks.

4.6. Time Complexity Analysis. Apart from security analysis of the image encryption scheme, performance analysis is also an important aspect to evaluate the encryption/decryption time and time complexity of the algorithm [69, 70]. A good encryption algorithm needs to have a fast encryption time and low computation complexity.

The encryption/decryption time can be calculated manually where it is mainly analyzed into six parts as follows: (a) mean for any row column of R, G, and B channels, so its complexity is $O(1)$; (b) the cyclic process N times for Chen's hyperchaotic system quantum logistic chaotic map, so it has complexity of $O(n)$; (c) the generation of three chaotic sequences (E^R, E^G, E^G), which are produced by Chen's hyperchaotic system with length $M \times N$ and hence

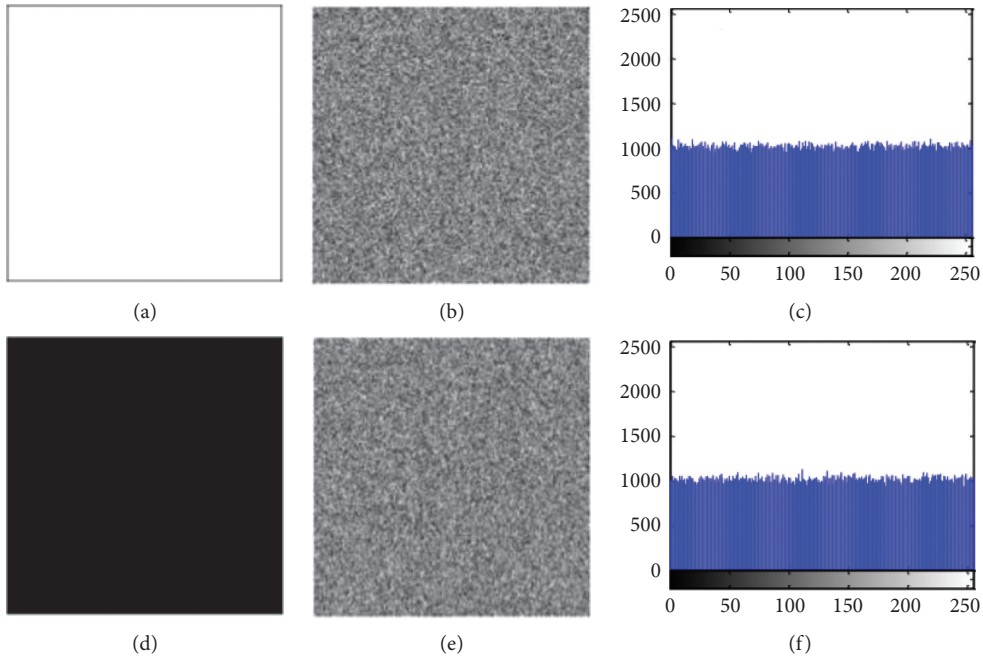


Figure 13: Simulation result of cryptanalysis tests: (a) all white image; (b) cipher image of (a); (c) histogram of (b); (d) all black image; (e) cipher image of (d); (f) histogram of (e).

Table 10: Running time analysis.

Image name	Image size	Encryption/decryption time (s)
Grayscale Aerial	256×256	0.4844
Color Airplane	256×256	0.8125
Grayscale Boat	512×512	1.4375
Color Lena	512×512	3.2969
Grayscale Male	1024×1024	5.7500
Color Baboon	1024×1024	13.6250

Table 11: Speed performance analysis (seconds).

Encryption scheme	Encryption time (s)	Processor speed	RAM	Platform
Ref. [71]	3.45	3 GHz	4 GB	Python 3.6
Ref. [22]	13.90	—	—	MATLAB R2016b
Ref. [72]	1.67	—	—	MATLAB
Ref. [73]	9.36	3.60 GHz	32 GB	MATLAB R2019b
Proposed algorithm	1.11	1.80 GHz	8 GB	MATLAB R2017b

the time cost is $(3 \times M \times N)$; (c) XOR operation having time complexity of $O(1)$; (d) time cost of chaotic map sequences and the generation of random matric being $O(n^2) = \max\{O(1), O(n^2)\}$; (e) the computational cost of density matrix being $O(1)$; and (f) the computational cost of XNOR operation being $O(1)$. From the above analyses, the total time cost of the proposed scheme is $O(n^2)$, so that the time consumption of proposed scheme hinges on t representing the number of code loops.

It can be calculated by using the in-built operations of the software used for implementation. Here, the elapsed time was measured by the tic and toc functions of MATLAB. The running speed of the proposed encryption scheme for a number of standard images with diverse sizes $(M \times M)$ is presented in Table 10. As a result, the proposed scheme reflects the efficiency to be used in practical cases.

Taking the 256×256 "Lena" image as an example, comparative analyses of the execution time among different encryption algorithms are illustrated in Table 11. It is observed that the proposed algorithm runs faster than the referenced algorithms [71–73]. In addition, it has less computational complexity.

5. Conclusion

Complex nonlinearity was preserved by choosing suitable chaotic maps. By choosing a high-dimensional chaotic system, the key space is increased. This study employed a

chaotic quantum logistic map, combining with both confusion and diffusion operations, to propose a new symmetric image encryption algorithm. This algorithm is based on Chen's hyperchaotic system to diffuse image pixels. Among them, the keystreams extracted are different for the same secret key associated with the plain image, which are true random numbers generated from noise arrays. Thus, the presented approach can achieve high resistance to the known-plaintext attack and chosen-plaintext attack as well as high level of sensitivity where the randomness of the random sequence displayed better behavior. At last, to confuse the relationship between original and encrypted images, the transpose process is applied to rows and columns of image. Through the results of extensive experiments and corresponding security analysis, it can be found that the salient features of the proposed symmetric image encryption algorithm can be summarized as follows: (a) large enough key space to resist brute-force attacks, (b) high level of security and being quite worthy of being called a good security system, (c) less computational complexity, and (d) being suitable for applications like wireless communications due to its fast implementation. An actual implementation of different kinds of operations in the scrambling stage to increase the security without affecting drastically the processing time is concerned and more detailed analysis on the chaotic or hyperchaotic dynamical systems deserves further investigation in the near future.

Data Availability

The data that support the findings of this study are openly available in [USC-SIPI Image Database] at [http://sipi.usc.edu/ database/], reference number [32].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] B. Sinha, S. Kumar, and C. Pradhan, "Comparative analysis of color image encryption using 3D chaotic maps," in *Proceedings of the International Conference on Communication and Signal Processing (ICCSP)*, pp. 332–335, Melmaruvathur, India, April 2016.

[2] J. Xu, P. Li, F. Yang, and H. Yan, "High intensity image encryption scheme based on quantum logistic chaotic map and complex hyperchaotic system," *IEEE Access*, vol. 7, pp. 167904–167918, 2019.

[3] Z. Li, C. Peng, W. Tan, and L. Li, "An effective chaos-based image encryption scheme using imitating jigsaw method," *Complexity*, vol. 2021, Article ID 88249115, 18 pages, 2021.

[4] X. Wang, N. Guan, H. Zhao, S. Wang, and Y. Zhang, "A new image encryption scheme based on coupling map lattices with mixed multi-chaos," *Scientific Reports*, vol. 10, no. 1, 2020.

[5] R. K. Singh, B. Kumar, D. K. Shaw, and D. A. Khan, "Level by level image compression-encryption algorithm based on quantum chaos map," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 7, pp. 844–851, 2021.

[6] M. Khan and H. M. Waseem, "A novel image encryption scheme based on quantum dynamical spinning and rotations," *PLoS ONE*, vol. 13, no. 11, Article ID e0206460, 2018.

[7] X. Liu, D. Xiao, and C. Liu, "Quantum image encryption algorithm based on bit-plane permutation and sine logistic map," *Quantum Information Processing*, vol. 19, no. 8, pp. 1–23, 2020.

[8] M. Ge and R. Ye, "A novel image encryption scheme based on 3D bit matrix and chaotic map with Markov properties," *Egyptian Informatics Journal*, vol. 20, no. 1, pp. 45–54, 2019.

[9] Y. Dong, X. Huang, Q. Mei, and Y. Gan, "Self-adaptive image encryption algorithm based on quantum logistic map," *Security and Communication Networks*, vol. 2021, Article ID 66749448, 12 pages, 2021.

[10] H. Liu, B. Zhao, and L. Huang, "Quantum image encryption scheme using arnold transform and S-box scrambling," *Entropy*, vol. 21, no. 4, p. 343, 2019.

[11] Y. Luo, R. Zhou, J. Liu, Y. Cao, and X. Ding, "A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map," *Nonlinear Dynamics*, vol. 93, no. 3, pp. 1165–1181, 2018.

[12] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, "A new algorithm for digital image encryption based on chaos theory," *Entropy*, vol. 23, no. 3, p. 341, 2021.

[13] Z. Tang, Y. Yang, S. Xu, C. Yu, and X. Zhang, "Image encryption with double spiral scans and chaotic maps," *Security and Communication Networks*, vol. 2019, Article ID 8694678, 15 pages, 2019.

[14] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Processing: Image Communication*, vol. 52, pp. 6–19, 2017.

[15] K. K. Butt, G. Li, F. Masood, and S. Khan, "A digital image confidentiality scheme based on pseudo-quantum chaos and Lucas sequence," *Entropy*, vol. 22, no. 11, p. 1276, 2020.

[16] Y. Liu, B. Zhou, Z. Li, J. Deng, and Z. Cai, "An image encryption method based on quantum fourier transformation," *International Journal of Intelligence Science*, vol. 8, no. 3, pp. 75–87, 2018.

[17] X. Liu, D. Xiao, and C. Liu, "Double quantum image encryption based on arnold transform and qubit random rotation," *Entropy*, vol. 20, no. 11, pp. 1–16, 2018.

[18] N. Zhou, X. Yan, H. Liang, X. Tao, and G. Li, "Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system," *Quantum Information Processing*, vol. 17, no. 12, pp. 1–36, 2018.

[19] C. Zhu, Z. Gan, Y. Lu, and X. Chai, "An image encryption algorithm based on 3-D DNA level permutation and substitution scheme," *Multimedia Tools and Applications International Journal*, vol. 17, no. 12, pp. 7227–7258, 2019.

[20] A. A. El-Latif, B. A. El-Atty, M. Amin, and A. M. Iliyasu, "Quantum-inspired cascaded discrete-time quantum walks with induced chaotic dynamics and cryptographic applications," *Scientific Reports*, vol. 10, p. 1, 2020.

[21] A. Alghafis, N. Munir, M. Khan, and I. Hussain, "An encryption scheme based on discrete quantum map and continuous chaotic system," *International Journal of Theoretical Physics*, vol. 59, no. 4, pp. 1227–1240, 2020.

[22] R. Sridevi and P. Philominathan, "Quantum colour image encryption algorithm based on DNA and unified logistic tent map," *Information Sciences Letters*, vol. 9, no. 3, pp. 219–231, 2020.

[23] W.-W. Hu, R.-G. Zhou, S. Jiang, X. Liu, and J. Luo, "Quantum image encryption algorithm based on generalized Arnold transform and Logistic map," *CCF Transactions on High Performance Computing*, vol. 2, no. 3, pp. 228–253, 2020.

[24] H. Wen, C. Zhang, P. Chen et al., "A quantum chaotic image cryptosystem and its application in IoT secure communication," *IEEE Access*, vol. 9, pp. 20481–20492, 2021.

[25] X. Wu, Y. Li, and J. Kurths, "A new color image encryption scheme using CML and a fractional-order chaotic system," *PLoS ONE*, vol. 10, no. 3, Article ID e0119660, 2015.

[26] J. Lü and G. Chen, "A new chaotic attractor coined," *International Journal of Bifurcation and Chaos*, vol. 12, no. 3, pp. 659–661, 2002.

[27] R. Zhang, L. Yu, D. Jiang et al., "A novel plaintext-related color image encryption scheme based on cellular neural network and chen's chaotic system," *Symmetry*, vol. 13, no. 3, p. 393, 2021.

[28] A. Z. Mahmoud, *On some new approaches for multimedia content encryption*, Ph.D. dissertation, Dept. Comp. Science, Menoufia Univ., Al Minufya, Egypt, 2015.

[29] X. Liu, D. Xiao, and Y. Xiang, "Quantum image encryption using intra and inter bit permutation based on logistic map," *IEEE Access*, vol. 7, pp. 6937–6946, 2019.

[30] G. Ye, K. Jiao, C. Pan, and X. Huang, "An effective framework for chaotic image encryption based on 3D logistic map," *Security and Communication Networks*, vol. 2018, no. 11, 11 pages, Article ID 8402578, 2018.

[31] Y. He, Y.-Q. Zhang, X. He, and X.-Y. Wang, "A new image encryption algorithm based on the OF-LSTMS and chaotic sequences," *Scientific Reports*, vol. 11, no. 1, pp. 1–22, 2021.

[32] SIPIUSC, "The USC-SIPI image database," 2021, http://sipi.usc.edu/database/.

[33] Y. Zhou, C. Li, W. Li, H. Li, W. Feng, and K. Qian, "Image encryption algorithm with circle index table scrambling and partition diffusion," *Nonlinear Dynamics*, vol. 103, no. 2, pp. 2043–2061, 2021.

[34] X. Yan, X. Wang, and Y. Xian, "Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10949–10983, 2021.

[35] M. Liu and G. Ye, "A new DNA coding and hyperchaotic system based asymmetric image encryption algorithm," *Mathematical Biosciences and Engineering*, vol. 18, no. 4, pp. 3887–3906, 2021.

[36] C. Fu, J.-j. Chen, H. Zou, W.-h. Meng, Y.-f. Zhan, and Y.-w. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Optics Express*, vol. 20, no. 3, pp. 2363–2378, 2012.

[37] Y. Wan, S. Gu, and B. Du, "A new image encryption algorithm based on composite chaos and hyperchaos combined with DNA coding," *Entropy*, vol. 22, no. 2, pp. 1–19, 2020.

[38] K. Jiao, G. Ye, Y. Dong, X. Huang, and J. He, "Image encryption scheme based on a generalized arnold map and RSA algorithm," *Security and Communication Networks*, vol. 2020, pp. 1–14, 2020.

[39] J. Ge, "ALCencryption: A secure and efficient algorithm for medical image encryption," *Computer Modeling in Engineering and Sciences*, vol. 125, no. 3, pp. 1083–1100, 2020.

[40] S. Zhu, C. Zhu, and W. Wang, "A new image encryption algorithm based on chaos and secure hash SHA-256," *Entropy*, vol. 20, no. 9, p. 716, 2018.

[41] X. Zhang, L. Wang, Y. Niu, G. Cui, and S. Geng, "Image encryption algorithm based on the H-fractal and dynamic self-invertible matrix," *Computational Intelligence and Neuroscience*, vol. 2019, no. 12, 12 pages, Article ID 9524080, 2019.

[42] C. Li, F. Zhao, C. Liu, L. Lei, and J. Zhang, "A hyperchaotic color image encryption algorithm and security analysis," *Security and Communication Networks*, vol. 2019, Article ID 8132547, 8 pages, 2019.

[43] H. Fan, K. Zhou, E. Zhang, W. Wen, and M. Li, "Subdata image encryption scheme based on compressive sensing and vector quantization," *Neural Computing & Applications*, vol. 32, no. 16, pp. 12771–12787, 2020.

[44] X. Xue, H. Jin, D. Zhou, and C. Zhou, "Medical image protection algorithm based on deoxyribonucleic acid chain of dynamic length," *Frontiers in Genetics*, vol. 12, pp. 1–18, Article ID 654663, 2021.

[45] S. Zhou, P. He, and N. Kasabov, "A dynamic DNA color image encryption method based on SHA-512," *Entropy*, vol. 22, no. 10, p. 1091, 2020.

[46] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "A novel image encryption scheme using the composite discrete chaotic system," *Entropy*, vol. 18, no. 8, p. 276, 2016.

[47] F. Yang, J. Mou, J. Liu, C. Ma, and H. Yan, "Characteristic analysis of the fractional-order hyperchaotic complex system and its image encryption application," *Signal Processing*, vol. 169, pp. 1–19, Article ID 107373, 2020.

[48] N. Tsafack, A. M. Iliyasu, N. J. De Dieu et al., "A memristive RLC oscillator dynamics applied to image encryption," *Journal of Information Security and Applications*, vol. 61, Article ID 102944, 2021.

[49] Z. Deng and S. Zhong, "A digital image encryption algorithm based on chaotic mapping," *Journal of Algorithms & Computational Technology*, vol. 13, pp. 1–11, 2019.

[50] J. Zeng and C. Wang, "A novel hyperchaotic image encryption system based on particle swarm optimization algorithm and cellular automata," *Security and Communication Networks*, vol. 2021, Article ID 6675565, 15 pages, 2021.

[51] L. Ding and Q. Ding, "A novel image encryption scheme based on 2D fractional chaotic map, DWT and 4D hyperchaos," *Electronics*, vol. 9, no. 8, p. 1280, 2020.

[52] I. Yasser, M. A. Mohamed, A. S. Samra, and F. Khalifa, "A chaotic-based encryption/decryption framework for secure multimedia communications," *Entropy*, vol. 22, no. 11, p. 1253, 2020.

[53] H. Liu, B. Zhao, J. Zou, L. Huang, and Y. Liu, "A lightweight image encryption algorithm based on message passing and chaotic map," *Security and Communication Networks*, vol. 2020, pp. 1–12, 2020.

[54] N. Sanam, A. Ali, T. Shah, and G. Farooq, "Non-associative algebra redesigning block cipher with color image encryption," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 1–21, 2021.

[55] F. Naz, I. A. Shoukat, R. Ashraf, U. Iqbal, and A. Rauf, "An ASCII based effective and multi-operation image encryption method," *Multimedia Tools and Applications*, vol. 79, no. 31-32, pp. 22107–22129, 2020.

[56] D. W. Ahmed, T. M. Jawad, and L. M. Jawad, "An effective color image encryption scheme based on double piecewise linear chaotic map method and RC4 algorithm," *Journal of Engineering Science & Technology*, vol. 16, no. 2, pp. 1319–1341, 2021.

[57] S. Zhu and C. Zhu, "Security analysis and improvement of an image encryption cryptosystem based on bit plane extraction and multi chaos," *Entropy*, vol. 23, no. 5, p. 505, 2021.

[58] Y. Chen, C. Tang, and Z. Yi, "A novel image encryption scheme based on PWLCM and standard map," *Complexity*, vol. 2020, Article ID 3026972, 23 pages, 2020.
[59] F. Masood, J. Ahmad, S. A. Shah, S. S. Jamal, and I. Hussain, "A novel hybrid secure image encryption based on julia set of fractals and 3D Lorenz chaotic map," *Entropy*, vol. 22, no. 3, p. 274, 2020.
[60] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Information Sciences*, vol. 546, pp. 1063–1083, 2021.
[61] U. Erkan, A. Toktas, S. Enginoglu, E. Akbacak, and D. N. H. Thanh, "An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN," *Multimedia Tools and Applications*, vol. 81, no. 78, pp. 7365–7391, 2022.
[62] C. Xu, J. Sun, and C. Wang, "A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems," *Multimedia Tools and Applications*, vol. 79, no. 9-10, pp. 5573–5593, 2020.
[63] G. Ye, K. Jiao, X. Huang, B.-M. Goi, and W.-S. Yap, "An image encryption scheme based on public key cryptosystem and quantum logistic map," *Scientific Reports*, vol. 10, no. 1, p. 19, 2020.
[64] M. A. A.-J. A. Mizher, R. Sulaiman, A. M. A. Abdalla, and M. A. A. Mizher, "A simple flexible cryptosystem for meshed 3D objects and images," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 6, pp. 844–851, 2019.
[65] Y. Luo, X. Ouyang, J. Liu, and L. Cao, "An image encryption method based on elliptic curve elgamal encryption and chaotic systems," *IEEE Access*, vol. 7, pp. 38507–38522, 2019.
[66] H.-Y. Gu, W.-Q. Yan, and J.-H. Zhang, "A novel image encryption scheme based on hyperchaotic cellular automaton," *Journal of Computers*, vol. 31, no. 6, pp. 155–168, 2020.
[67] Y. Dong, X. Huang, and G. Ye, "Visually meaningful image encryption scheme based on DWT and schur decomposition," *Security and Communication Networks*, vol. 2021, Article ID 6677325, 16 pages, 2021.
[68] L. M. Heucheun Yepdia, A. Tiedeu, and G. Kom, "A robust and fast image encryption scheme based on a mixing technique," *Security and Communication Networks*, vol. 2021, Article ID 6615708, 17 pages, 2021.
[69] B. Mondal, P. K. Behera, and S. Gangopadhyay, "A secure image encryption scheme based on a novel 2D sine-cosine cross-chaotic (SC3) map," *Journal of Real-Time Image Processing*, vol. 18, no. 1, pp. 1–18, 2021.
[70] R. I. Abdelfattah, H. Mohamed, and M. E. Nasr, "Secure image encryption scheme based on DNA and new multi chaotic map," *Journal of Physics: Conference Series*, vol. 1447, no. 1, pp. 1–11, Article ID 012053, 2020.
[71] X. Hu, L. Wei, W. Chen, Q. Chen, and Y. Guo, "Color image encryption algorithm based on dynamic chaos and matrix convolution," *IEEE Access*, vol. 8, pp. 12452–12466, 2020.
[72] X. Wang and Y. Su, "Color image encryption based on chaotic compressed sensing and two-dimensional fractional Fourier transform," *Scientific Reports*, vol. 10, pp. 1–19, Article ID 18556, 2020.
[73] D. Zhang, L. Chen, and T. Li, "Hyper-chaotic color image encryption based on transformed zigzag diffusion and RNA operation," *Entropy*, vol. 23, no. 3, p. 361, 2021.

Hindawi
Complexity
Volume 2022, Article ID 8495451, 8 pages
https://doi.org/10.1155/2022/8495451

WILEY | Hindawi

Research Article

Fixed Point Results of Dynamic Process $\check{D}(\Upsilon, \mu_0)$ through F_I^C-Contractions with Applications

Amjad Ali,[1] Eskandar Ameer,[2] Muhammad Arshad,[1] Hüseyin Işık,[3] and Mustafa Mudhesh[1]

[1]*Department of Mathematics and Statistics, International Islamic University, Islamabad 44000, Pakistan*
[2]*Department of Mathematics, Taiz University, Taiz, Yemen*
[3]*Department of Engineering Science, Bandirma Onyedi Eylül University, Bandirma, Balikesir 10200, Turkey*

Correspondence should be addressed to Eskandar Ameer; eskanderameer@taiz.edu.ye and Hüseyin Işık; isikhuseyin76@gmail.com

Received 1 September 2021; Accepted 27 December 2021; Published 3 February 2022

Academic Editor: Padmapriya Praveenkumar

This article constitutes the new fixed point results of dynamic process D(Υ, μ0) through FIC-integral contractions of the Ciric kind and investigates the said contraction to iterate a fixed point of set-valued mappings in the module of metric space. To do so, we use the dynamic process instead of the conventional Picard sequence. The main results are examined by tangible nontrivial examples which display the motivation for such investigation. The work is completed by giving an application to Liouville-Caputo fractional differential equations.

1. Introduction and Preliminaries

In the recent past, the study of metric fixed point theory untied a portal to a new area of pure and applied mathematics, the fixed point theory and its application. This concept was prolonged by either extending metric space into its extensions or by modifying the structure of the contractions (see [1–7]). The most classical structure known as Banach contraction principle (or contraction) theorem was introduced by a Polish mathematician Banach in 1922 [8]. The applications of fixed points of Banach contraction mappings defined for different kinds of spaces is the guarantee of the existence and uniqueness of solutions of differential and integral type equations. The variety of these nonlinear problems imposes the search for more and better tools, which are recently very remarkable in the literature. One of such tools was recently conveyed by Wardowski [9], where the author originated a new class of contractive mapping called F-contraction.

Nadler [10] using the idea of Pompeiu–Hausdorff metric discussed the Banach contraction mappings for set-valued functions rather than single-valued functions. Let (Δ, δ) be a metric space. For $\mu_1, \mu_2 \in \Delta$ and $A, B \subseteq \Delta$, define the Pompeiu–Hausdorff metric $\hat{H}$ induced by δ on $\mathrm{CB}(\Delta)$ as follows:

$$\hat{H}(A, B) = \max\left\{\sup_{\mu_1 \in A} \check{D}(\mu_1, B), \sup_{\mu_2 \in B} \check{D}(\mu_2, A)\right\}, \quad (1)$$

for each $A, B \in \mathrm{CB}(\Delta)$, where $\mathrm{CB}(\Delta)$ denotes the set of all nonempty closed bounded subsets of Δ and $\check{D}(\mu_1, B) = \inf_{\mu_2 \in B} \delta(\mu_1, \mu_2)$. An element $\mu \in \Delta$ is called a fixed point of a set-valued mapping, i.e., $\Upsilon: \Delta \longrightarrow \mathrm{CB}(\Delta)$, then $\mu \in \Upsilon(\mu)$. Also, denote the family of nonempty compact subsets of Δ by $K(\Delta)$.

Some well-known results are related to this section and hereafter.

Lemma 1. *Let A and B be nonempty proximal subsets of a metric space (Δ, δ). If $\alpha \in A$, then*

$$\delta(\alpha, B) \leq H(A, B). \quad (2)$$

Lemma 2 (see [11]). *Let (Δ, δ) be a metric space and a sequence $(\mu_i)_{i \in N}$ in (Δ, δ) such that*

$$\lim_{i \longrightarrow \infty} \delta(\mu_i, \mu_{i+1}) = 0 \quad (3)$$

is not a Cauchy sequence. Then, there exists $\varepsilon > 0$ and subsequences of positive integers (μ_{i_j}) and (μ_{l_j}), $\mu_{i_j} > \mu_{l_j} > j$ such that

$$\left[\delta\left(\mu_{i_j},\mu_{l_j}\right),\delta\left(\mu_{i_j+1},\mu_{i_j}\right),\delta\left(\mu_{i_j},\mu_{l_j-1}\right),\delta\left(\mu_{i_j+1},\mu_{l_j-1}\right),\delta\left(\mu_{i_j+1},\mu_{l_j+1}\right)\right]\longrightarrow\varepsilon^{+},\text{ as } j\longrightarrow+\infty. \tag{4}$$

Definition 1 (see [12]). Let $\Upsilon\colon \Delta\longrightarrow N(\Delta)$ be a multivalued mapping and $\mu_0\in\Delta$ be arbitrary and fixed. Define

$$\check{D}(\Upsilon,\mu_0)=\left\{\left(\mu_j\right)_{j\in N\cup\{0\}}\colon \mu_j\in\Upsilon\left(\mu_{j-1}\right),\text{for all } j\in N\right\}. \tag{5}$$

Each element of $\check{D}(\Upsilon,\mu_0)$ is called a dynamic process of Υ starting point μ_0. The dynamic process $(\mu_j)_{j\in N\cup\{0\}}$ onward be written as (μ_j).

Example 1 (see [12]). Let $\Delta=C([0,1])$ be a Banach space with a norm $\|\mu\|=\sup_{r\in[0,1]}|\mu(r)|$, $\mu\in\Delta$. Let $\Upsilon\colon\Delta\longrightarrow 2^{\Delta}$ be such that, for every $\mu\in\Delta$, $\Upsilon(\mu)$ is a collection of the functions

$$r\mapsto k\int_0^r\mu(t)\mathrm{d}t,\quad k\in[0,1], \tag{6}$$

that is,

$$(\Upsilon(\mu))(r)=\left\{k\int_0^r\mu(t)\mathrm{d}t\colon k\in[0,1]\right\},\mu\in\Delta, \tag{7}$$

and let $\mu_0(r)=r$, $r\in[0,1]$, then the sequence $(1/(j!(j+1)!)r^{j+1})$ is a dynamic process of Υ with starting point μ_0.

A mapping $\Upsilon\colon\Delta\longrightarrow R$ is said to be $\check{D}(\Upsilon,\mu_0)$-dynamic lower semicontinuous at $\mu\in\Delta$, if for every dynamic process $(\mu_j)\in\check{D}(\Upsilon,\mu_0)$ and for every subsequence $(\mu_{j(i)})$ of (μ_j) convergent to μ, we get $\Upsilon(\mu)\le\liminf_{i\longrightarrow\infty}\Upsilon((\mu_{j(i)}))$. If Υ is $\check{D}(\Upsilon,\mu_0)$-dynamic lower semicontinuous at each $\mu\in\Delta$, then Υ is said to be $\check{D}(\Upsilon,\mu_0)$-dynamic lower semicontinuous. If for every sequence $(\mu_j)\subset\Delta$ and $\mu\in\Delta$ such that $\mu_j\longrightarrow\mu$, we have $\Upsilon(\mu)\le\liminf_{j\longrightarrow\infty}\Upsilon(\mu_j)$, then Υ is known as lower semicontinuous.

As of now, Branciari [5] generalized the second well-known contraction of Banach contraction mappings is determined, i.e., let (Δ,δ) be a metric space and a mapping $\Upsilon\colon\Delta\longrightarrow\Delta$ such that

$$\int_0^{\delta\left(\Upsilon\mu_1,\Upsilon\mu_2\right)}\varphi(s)ds\le\beta\int_0^{\delta\left(\mu_1,\mu_2\right)}\varphi(s)ds \tag{8}$$

for all $\mu_1,\mu_2\in\Delta$, where $\beta\in(0,1)$, $\varphi\in\Phi$, and Φ is the class of all functions $\varphi\colon[0,+\infty)\longrightarrow[0,+\infty)$ which is Lebesgue integrable, summable on each compact subset of $[0,+\infty)$ and $\int_0^{\varepsilon}\varphi(s)\mathrm{d}s>0$ for all $\varepsilon>0$. Then, Υ has a fixed point.

The following lemmas are helpful for our main results. We shall also suppose that $\varphi\in\Phi$.

Lemma 3 (see [6]). *Let $(\mu_i)_{i\in N}$ be a nonnegative sequence in such a way that $\lim_{i\longrightarrow+\infty}\mu_i=\mu$. Then,*

$$\lim_{i\longrightarrow+\infty}\int_0^{\mu_i}\varphi(s)\mathrm{d}s=\int_0^{\mu}\varphi(s)\mathrm{d}s. \tag{9}$$

Lemma 4 (see [6]). *Let $(\mu_i)_{i\in N}$ be a nonnegative sequence. Then,*

$$\lim_{i\longrightarrow+\infty}\int_0^{\mu_i}\varphi(s)\mathrm{d}s=0\Leftrightarrow\lim_{i\longrightarrow+\infty}\mu_i=0. \tag{10}$$

In 2012, Wardowski [9] initiated the term of F-contraction and implemented on fixed point theorem related with F-contraction. So, with the intent that, he generalizes contraction theorem which is a purely altered from many past results in the literature frame.

Definition 2 (see [9]). Let $\Upsilon\colon\Delta\longrightarrow\Delta$ is called an $\mathcal{F}$-contraction on a metric space (Δ,δ), if there exist $\mathcal{F}\in\nabla_{\mathrm{F}}$ and $\tau\in R_+$ in such a way that, $\delta(\Upsilon\mu_1,\Upsilon\mu_2)>0$ implies

$$\tau+\mathcal{F}(\delta(\Upsilon\mu_1,\Upsilon\mu_2))\le\mathcal{F}(\delta(\mu_1,\mu_2)). \tag{11}$$

For each $\mu_1,\mu_2\in\Delta$, where ∇_{F} is the class of all functions $\mathcal{F}\colon R_+\longrightarrow R$ such that

$(\mathcal{F}_i)$ $\mu_1<\mu_2$ implies $\mathcal{F}(\mu_1)<\mathcal{F}(\mu_2)$ for all $\mu_1,\mu_2\in R_+$.

$(\mathcal{F}_{ii})$ For each sequence $\{\mu_j\}$ of positive real numbers,

$$\lim_{j\longrightarrow\infty}\mu_j=0\text{ iff }\lim_{j\longrightarrow\infty}\mathcal{F}\left(\mu_j\right)=-\infty. \tag{12}$$

$(\mathcal{F}_{\mathrm{iii}})$ There is $k\in(0,1)$ in such a way that $\lim_{c\longrightarrow0^+}c^k\mathcal{F}(c)=0$.

From now, we present some well-defined examples of $\mathcal{F}$-contraction that are listed as follows:

$(\mathcal{F}a)$: $\mathcal{F}(\mu)=\ln\mu$

$(\mathcal{F}b)$: $\mathcal{F}(\mu)=\ln\mu+\mu$

$(\mathcal{F}c)$: $\mathcal{F}(\mu)=-1/\sqrt{\mu}$

$(\mathcal{F}\ d)$: $\mathcal{F}(\mu)=\ln(\mu^2+\mu)$

Owing to $(\mathcal{F}_i)$ and (11), clearly, we conclude that every F-contraction Υ is a contractive mapping. Consequently, every F-contraction is a continuous mapping (see more [13]).

The main purpose of this manuscript is to introduce the new concept of dynamic iterative process $\check{D}(\Upsilon,\mu_0)$ based on F_I^C-integral contractions and prove some related multivalued fixed point results in the class of metric space. To approximate our main results by tangible examples are also determined. At the end, the work is completed by giving an application to Liouville–Caputo fractional differential equations.

2. Main Result

First, we give our main definition.

Definition 3. Let (Δ,δ) be a metric space, $\mu_0\in\Delta$, $\mathcal{F}\in\nabla_{\mathrm{F}}$ and $\varphi\in\Phi$. A set-valued map $\Upsilon\colon\Delta\longrightarrow\mathrm{CB}(\Delta)$ is said to be F_I^C-integral contraction with respect to a dynamic process $(\mu_i)\in\check{D}(\Upsilon,\mu_0)$, if there exists $\tau\colon R_+\longrightarrow R_+$ such that

$$\widehat{H}(\Upsilon\mu_i, \Upsilon\mu_{i+1}) > 0 \Rightarrow \tau(U(\mu_{i-1}, \mu_i)) + \mathscr{F}\left(\int_0^{\widehat{H}(\Upsilon\mu_i, \Upsilon\mu_{i+1})} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F} \cdot \left(\int_0^{U(\mu_{i-1}, \mu_i)} \varphi(s)\mathrm{d}s\right), \tag{13}$$

for all $i \in N$, where

$$U(\mu_{i-1}, \mu_i) = \max\left\{\delta(\mu_{i-1}, \mu_i), \check{D}(\mu_{i-1}, \Upsilon\mu_{i-1}), \check{D}(\mu_i, \Upsilon\mu_i), \cdot \frac{\check{D}(\mu_{i-1}, \Upsilon\mu_i) + \check{D}(\mu_i, \Upsilon\mu_{i-1})}{2}\right\}. \tag{14}$$

Remark 1. For the act of continuing our results, we consider only the dynamic processes $(\mu_i) \in \check{D}(\Upsilon, \mu_0)$ satisfying the following structure:

$$\delta(\mu_i, \mu_{i+1}) > 0 \Rightarrow \delta(\mu_{i-1}, \mu_i) > 0 \text{ for each } i \in N. \tag{15}$$

If the investigated process does not satisfy (15), then there is $i_0 \in N$ such that

$$\delta\left(\mu_{i_0}, \mu_{i_{0+1}}\right) > 0 \tag{16}$$

and

$$\delta\left(\mu_{i_{0-1}}, \mu_{i_0}\right) = 0. \tag{17}$$

Then, we get $\mu_{i_{0-1}} = \mu_{i_0} \in \Upsilon\mu_{i_{0-1}}$ which implies the existence of fixed point due to this consideration of dynamic process that satisfying (15) does not depreciate a generality of our approach.

Example 2. Let $\mathscr{F}: R_+ \longrightarrow R$ be defined by $\mathscr{F}(\mu) = \ln\mu$. Each set-valued F_I^C-integral contraction Υ on a metric space (Δ, δ) with respect to dynamic process $\check{D}(\Upsilon, \mu_0)$ assures that

$$\tau(U(\mu_{i-1}, \mu_i)) + \mathscr{F}\left(\int_0^{\widehat{H}(\Upsilon\mu_i, \Upsilon\mu_{i+1})} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F}\left(\int_0^{U(\mu_{i-1}, \mu_i)} \varphi(s)\mathrm{d}s\right). \tag{18}$$

Upon setting, we have

$$\int_0^{\widehat{H}(\Upsilon\mu_i, \Upsilon\mu_{i+1})} \varphi(s)\mathrm{d}s \leq e^{-\tau(U(\mu_{i-1}, \mu_i))} \int_0^{U(\mu_{i-1}, \mu_i)} \varphi(s)\mathrm{d}s, \tag{19}$$

for all $i \in N$, $(\mu_i) \in \check{D}(\Upsilon, \mu_0)$, and $\Upsilon\mu_{i-1} \neq \Upsilon\mu_i$. In view of the above observations, clearly, for (μ_{i_0-1}), $(\mu_{i_0}) \in \check{D}(\Upsilon, \mu_0)$ such that $\Upsilon\mu_{i_{0-1}} = \Upsilon\mu_{i_0}$, the following inequality also holds through $\check{D}(\Upsilon, \mu_0)$

$$\int_0^{\widehat{H}\left(\Upsilon\mu_{i_0-1}, \Upsilon\mu_{i_0}\right)} \varphi(s)\mathrm{d}s \leq e^{-\tau\left(U\left(\mu_{i_0-1}, \mu_{i_0}\right)\right)} \int_0^{U\left(\mu_{i_0-1}, \mu_{i_0}\right)} \varphi(s)\mathrm{d}s, \tag{20}$$

that is, Υ is a contraction.

Theorem 1. *Let (Δ, δ) be a complete metric space, $\mu_0 \in \Delta$ and $\Upsilon: \Delta \longrightarrow K(\Delta)$ be a set-valued F_I^C-integral contraction with respect to the dynamic process $(\mu_i) \in \check{D}(\Upsilon, \mu_0)$. Assume that*

Proof. In view of $(\mu_i) \in \check{D}(\Upsilon, \mu_0)$, if there exists $i_0 \in N$ such that $\mu_{i_0} = \mu_{i_0+1}$, then the existence of a fixed point is obvious. Therefore, if we let $\mu_i \notin \Upsilon\mu_i$, then $\check{D}(\mu_i, \Upsilon\mu_i) > 0$ for every $i \in N$. Using (15) and by Lemma 1, one writes

$$\mathscr{F}\left(\int_0^{\check{D}(\mu_i, \Upsilon\mu_i)} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F}\left(\int_0^{\widehat{H}(\Upsilon\mu_i, \Upsilon\mu_{i+1})} \varphi(s)\mathrm{d}s\right), \tag{21}$$

$$\leq \mathscr{F}\left(\int_0^{U(\mu_{i-1}, \mu_i)} \varphi(s)\mathrm{d}s\right) - \tau(U(\mu_{i-1}, \mu_i)) = \mathscr{F}\left(\int_0^{\max} \left\{\begin{array}{c} \delta(\mu_{i-1}, \mu_i), \check{D}(\mu_{i-1}, \Upsilon\mu_{i-1}), \check{D}(\mu_i, \Upsilon\mu_i), \\ \dfrac{\check{D}(\mu_{i-1}, \Upsilon\mu_i) + \check{D}(\mu_i, \Upsilon\mu_{i-1})}{2} \end{array}\right\}\right) \tag{22}$$

Moreover, since $\Upsilon\mu_i$ is compact, we obtain $\mu_{i+1} \in \Upsilon\mu_i$ such that $\delta(\mu_i, \mu_{i+1}) = \check{D}(\mu_i, \Upsilon\mu_i)$. Using (21), we have

$$\mathscr{F}\left(\int_0^{\delta(\mu_i, \mu_{i+1})} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F}\left(\int_0^{\widehat{H}(\Upsilon\mu_{i-1}, \Upsilon\mu_i)} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F}\left(\int_0^{\delta(\mu_{i-1}, \mu_i)} \varphi(s)\mathrm{d}s\right) - \tau(\delta(\mu_{i-1}, \mu_i)) < \mathscr{F}\left(\int_0^{\delta(\mu_{i-1}, \mu_i)} \varphi(s)\mathrm{d}s\right). \tag{23}$$

In view of the above observations, $\{\delta(\mu_i, \mu_{i+1})\}$ is decreasing and hence convergent. We now show that $\lim_{i\longrightarrow\infty}\delta(\mu_i, \mu_{i+1}) = 0$. In the light of (D1), there exist $\sigma > 0$ and $i_0 \in N$ such that $\tau(\delta(\mu_{i-1}, \mu_i)) > \sigma$ for all $i > i_0$. So, we have

$$\begin{aligned}
\mathscr{F}\left(\int_0^{\delta(\mu_i,\mu_{i+1})}\varphi(s)\mathrm{d}s\right) &\le \mathscr{F}\left(\int_0^{\delta(\mu_{i-1},\mu_i)}\varphi(s)\mathrm{d}s\right) - \tau(\delta(\mu_{i-1},\mu_i))\\
&\le \mathscr{F}\left(\int_0^{\delta(\mu_{i-2},\mu_{i-1})}\varphi(s)\mathrm{d}s\right) - \tau(\delta(\mu_{i-2},\mu_{i-1})) - \tau(\delta(\mu_{i-1},\mu_i))\\
&\vdots\\
&\le \mathscr{F}\left(\int_0^{\delta(\mu_0,\mu_1)}\varphi(s)\mathrm{d}s\right) - \tau(\delta(\mu_0,\mu_1)) - \cdots - \tau(\delta(\mu_{i-1},\mu_i))\\
&= \mathscr{F}\left(\int_0^{\delta(\mu_0,\mu_1)}\varphi(s)\mathrm{d}s\right) - \left(\tau(\delta(\mu_0,\mu_1)) + \cdots + \tau\left(\delta\left(\mu_{i_{0-1}},\mu_{i_0}\right)\right)\right)\\
&\quad - \tau\left(\delta\left(\mu_{i_0},\mu_{i_{0+1}}\right)\right)\Big) + \cdots + \tau(\delta(\mu_{i-1},\mu_i))\\
&\le \mathscr{F}\left(\int_0^{\delta(\mu_0,\mu_1)}\varphi(s)\mathrm{d}s\right) - (i - i_0)\sigma.
\end{aligned} \tag{24}$$

Let us set $\lambda_i = \int_0^{\delta(\mu_i,\mu_{i+1})}\varphi(s)\mathrm{d}s > 0$ for $i = 0, 1, 2, \ldots$ and from (24), we see that $\lim_{i\longrightarrow\infty}\mathscr{F}(\lambda_i) = -\infty$. By means of $(\mathscr{F}_{\mathrm{ii}})$, we have

$$\lim_{i\longrightarrow\infty}\lambda_i = 0. \tag{25}$$

Also, in the light of $(\mathscr{F}_{\mathrm{iii}})$, there is $\alpha \in (0, 1)$ such that

$$\lim_{i\longrightarrow\infty}[\lambda_i]^{\alpha}\mathscr{F}[\lambda_i] = 0. \tag{26}$$

Furthermore, from (24), we can write for all $i > i_0$

$$\begin{aligned}
[\lambda_i]^{\alpha}\mathscr{F}[\lambda_i] - [\lambda_i]^{\alpha}\mathscr{F}[\lambda_0] &\le [\lambda_i]^{\alpha}(\mathscr{F}(\lambda_0) - (i - i_0)\sigma)\\
&\quad - [\lambda_i]^{\alpha}\mathscr{F}[\lambda_0]\\
&= -[\lambda_i]^{\alpha}(i - i_0)\sigma \le 0.
\end{aligned} \tag{27}$$

Taking limit as $i \longrightarrow \infty$ in (27) and using (26), we have

$$\lim_{i\longrightarrow\infty} i[\lambda_i]^{\alpha} = 0. \tag{28}$$

Let us perceive that, from (28), there is $i_1 \in \mathbb{N}$ such that $i[\lambda_i]^{\alpha} \le 1$ for all $i \ge i_1$. We have

$$\lambda_i \le \frac{1}{i^{1/\alpha}}. \tag{29}$$

Now, in order to show that $\{\mu_i\}$ is a Cauchy sequence, we consider $j_1, j_2 \in \mathbb{N}$ such that $j_1 > j_2 \ge i_1$. From (29) and by virtue of metric condition, we have

$$\begin{aligned}
&\int_0^{\delta\left(\mu_{j_1},\mu_{j_2}\right)}\varphi(s)\mathrm{d}s\\
&\quad\le \int_0^{\delta\left(\mu_{j_1},\mu_{j_1+1}\right)}\varphi(s)\mathrm{d}s\\
&\quad\quad + \int_0^{\delta\left(\mu_{j_1+1},\mu_{j_1+2}\right)}\varphi(s)\mathrm{d}s + \cdots + \int_0^{\delta\left(\mu_{j_2-1},\mu_{j_2}\right)}\varphi(s)\mathrm{d}s\\
&\quad= \lambda_{j_1} + \lambda_{j_1+1} + \cdots + \lambda_{j_2-1}\\
&\quad= \sum_{l=j_1}^{j_2-1}\lambda_l \le \sum_{l=j_1}^{\infty}\lambda_l \le \sum_{l=j_1}^{\infty}\frac{1}{l^{1/\alpha}}.
\end{aligned} \tag{30}$$

In the light of (30) and view of convergence of series $\sum_{l=j_1}^{\infty}1/l^{1/\alpha}$, we see that $\int_0^{\delta(\mu_{j_1},\mu_{j_2})}\varphi(s)\mathrm{d}s \longrightarrow 0$. Hence, $\{\mu_i\}$ is Cauchy sequence in (Δ, δ). Furthermore, for the completeness of Δ, there is $\mu^* \in \Delta$ such that $\lim_{i\longrightarrow\infty}\mu_i = \mu^*$. Since Υ is compact, then we have $\Upsilon\mu_i \longrightarrow \Upsilon\mu^*$. By Lemma 1, one writes

$$\check{D}(\mu_i, \Upsilon\mu^*) \le \widehat{H}(\Upsilon\mu_{i-1}, \Upsilon\mu^*). \tag{31}$$

So, $\check{D}(\mu^*, \Upsilon\mu^*) = 0$ and $\mu^* \in \Upsilon\mu^*$. Suppose, on the contrary, $\mu^* \notin \Upsilon\mu^*$. Then, there exist $i_0 \in \mathbb{N}$ and subsequence $\{\mu_{i_k}\}$ of $\{\mu_i\}$ such that $\check{D}(\mu_{i_k+1}, \Upsilon\mu^*) > 0$ for each $i_k \ge i_0$ (otherwise, there is $i_1 \in \mathbb{N}$ such that $\mu_i \in \Upsilon\mu^*$ for every $i \ge i_1$, which yields that $\mu^* \in \Upsilon\mu^*$). By contractive condition, one writes

$$\mathscr{F}\left(\int_0^{\check{D}\left(\mu_{i_{k+1}},\Upsilon\mu^*\right)}\varphi(s)\mathrm{d}s\right)\leq\mathscr{F}\left(\int_0^{\widehat{H}\left(\Upsilon\mu_{i_k},\Upsilon\mu^*\right)}\varphi(s)\mathrm{d}s\right)\leq\mathscr{F}\left(\int_0^{U\left(\mu_{i_k},\mu^*\right)}\varphi(s)\mathrm{d}s\right)-\tau\left(U\left(\mu_{i_k},\mu^*\right)\right). \tag{32}$$

Upon letting $k\longrightarrow\infty$ in (32),

$$\mathscr{F}\left(\int_0^{\check{D}\left(\mu^*,\Upsilon\mu^*\right)}\varphi(s)\mathrm{d}s\right)\leq\mathscr{F}\left(\int_0^{\check{D}\left(\mu^*,\Upsilon\mu^*\right)}\varphi(s)\mathrm{d}s\right)-\tau\left(\check{D}\left(\mu^*,\mu^*\right)\right)<\mathscr{F}\left(\int_0^{\check{D}\left(\mu^*,\Upsilon\mu^*\right)}\varphi(s)\mathrm{d}s\right). \tag{33}$$

which is a contradiction. On the other hand, we see that the mapping $\Delta\ni\mu_i\mapsto\delta(\mu_i,\Upsilon\mu_i)$ is $\check{D}(\Upsilon,\mu_0)$-dynamic lower semicontinuous, we have

$$\int_0^{\check{D}\left(\mu^*,\Upsilon\mu^*\right)}\varphi(s)\mathrm{d}s\leq\liminf_{n\longrightarrow\infty}\int_0^{\check{D}\left(\mu_{i_k},\Upsilon\mu_{i_k}\right)}\varphi(s)\mathrm{d}s\leq\liminf_{n\longrightarrow\infty}\int_0^{\check{D}\left(\mu_i,\Upsilon\mu_i\right)}\varphi(s)\mathrm{d}s=0 \tag{34}$$

and by virtue of closedness of $\Upsilon\mu^*$ implies that $\mu^*\in\Upsilon\mu^*$ which means that μ^* is a fixed point of Υ. □

Remark 2. If in Theorem 1, instead of the contractive condition (13), we assume the following condition

$$\widehat{H}\left(\Upsilon\mu_i,\Upsilon\mu_{i+1}\right)>0\Rightarrow\tau\left(U_j\left(\mu_{i-1},\mu_i\right)\right)+\mathscr{F}\left(\int_0^{\widehat{H}\left(\Upsilon\mu_i,\Upsilon\mu_{i+1}\right)}\varphi(s)\mathrm{d}s\right)\leq\mathscr{F}\left(\int_0^{U_j\left(\mu_{i-1},\mu_i\right)}\varphi(s)\mathrm{d}s\right), \tag{35}$$

where $j\in\{1,2,3\}$ and

$$U_1(\mu_{i-1},\mu_i)=\delta(\mu_{i-1},\mu_i),$$

$$U_2(\mu_{i-1},\mu_i)=\max\{\delta(\mu_{i-1},\mu_i),\check{D}(\mu_{i-1},\Upsilon\mu_{i-1}),\check{D}(\mu_i,\Upsilon\mu_i)\},$$

$$U_3(\mu_{i-1},\mu_i)=\max\left\{\delta(\mu_{i-1},\mu_i),\frac{\check{D}(\mu_{i-1},\Upsilon\mu_{i-1})+\check{D}(\mu_i,\Upsilon\mu_i)}{2},\cdot\frac{\check{D}(\mu_{i-1},\Upsilon\mu_i)+\check{D}(\mu_i,\Upsilon\mu_{i-1})}{2}\right\}, \tag{36}$$

for all $i\in N$, $(\mu_i)\in\check{D}(\Upsilon,\mu_0)$, then there exists a fixed point of the mapping Υ with the assumptions (D1) and (D2) on Theorem 1.

Corollary 1. *Let (Δ,δ) be a complete metric space, $\mu_0\in\Delta$, $\mathscr{F}\in\nabla_{\mathrm{F}}$, $\varphi\in\Phi$, and $\Upsilon\colon\Delta\longrightarrow K(\Delta)$. Assume that there exists $\tau\colon R_+\longrightarrow R_+$ such that*

$$\widehat{H}\left(\Upsilon\mu_i,\Upsilon\mu_{i+1}\right)>0\Rightarrow\tau\left(U\left(\mu_{i-1},\mu_i\right)\right)-\frac{1}{\int_0^{\widehat{H}\left(\Upsilon\mu_i,\Upsilon\mu_{i+1}\right)}\varphi(s)\mathrm{d}s}\leq-\frac{1}{\int_0^{U\left(\mu_{i-1},\mu_i\right)}\varphi(s)\mathrm{d}s}, \tag{37}$$

for all $i\in N$, $\mu_i\in\check{D}(\Upsilon,\mu_0)$, where

$$U\left(\mu_{i-1},\mu_i\right)=\max\left\{\delta\left(\mu_{i-1},\mu_i\right),\check{D}\left(\mu_{i-1},\Upsilon\mu_{i-1}\right),\check{D}\left(\mu_i,\Upsilon\mu_i\right),\cdot\frac{\check{D}\left(\mu_{i-1},\Upsilon\mu_i\right)+\check{D}\left(\mu_i,\Upsilon\mu_{i-1}\right)}{2}\right\}. \tag{38}$$

Then, there exists a fixed point of the mapping Υ with the assumptions (D1) and (D2) on Theorem 1.

Proof. If we choose $\mathscr{F}(\mu)=-1/\mu$, the proof follows from Theorem 1. □

Corollary 2. *Let (Δ,δ) be a complete metric space, $\mu_0\in\Delta$, $\mathscr{F}\in\nabla_{\mathrm{F}}$, $\varphi\in\Phi$, and $\Upsilon\colon\Delta\longrightarrow K(\Delta)$. Assume that there exists $\tau\colon R_+\longrightarrow R_+$ such that*

$$\widehat{H}\left(\Upsilon\mu_i,\Upsilon\mu_{i+1}\right)>0\Rightarrow\tau\left(U\left(\mu_{i-1},\mu_i\right)\right)+\frac{1}{1-\exp\int_0^{\widehat{H}\left(\Upsilon\mu_i,\Upsilon\mu_{i+1}\right)}\varphi(s)\mathrm{d}s}\leq\frac{1}{1-\exp\int_0^{U\left(\mu_{i-1},\mu_i\right)}\varphi(s)\mathrm{d}s}, \tag{39}$$

for all $i\in N$, $\mu_i\in\check{D}(\Upsilon,\mu_0)$, where

$$U\left(\mu_{i-1},\mu_i\right)=\max\left\{\delta\left(\mu_{i-1},\mu_i\right),\check{D}\left(\mu_{i-1},\Upsilon\mu_{i-1}\right),\check{D}\left(\mu_i,\Upsilon\mu_i\right),\frac{\check{D}\left(\mu_{i-1},\Upsilon\mu_i\right)+\check{D}\left(\mu_i,\Upsilon\mu_{i-1}\right)}{2}\right\}. \tag{40}$$

Then, there exists a fixed point of the mapping Υ with the assumptions (D1) and (D2) on Theorem 1.

Proof. If we choose $\mathscr{F}(\mu)=1/(1-\exp(\mu))$, the proof follows from Theorem 1.

The direct consequence of Theorem 1 for single-valued maps is the following. □

Corollary 3. *Let (Δ,δ) be a complete metric space, $\mu_0\in\Delta$, $\mathscr{F}\in\nabla_{\mathrm{F}}$, $\varphi\in\Phi$, and $\Upsilon\colon\Delta\longrightarrow\Delta$. Assume that there exists $\tau\colon R_+\longrightarrow R_+$ such that $\delta(\Upsilon^i\mu_0,\Upsilon^{i+1}\mu_0)>0$ implies*

$$\tau\left(\delta\left(\Upsilon^{i-1}\mu_0, \Upsilon^{i}\mu_0\right)\right) + \mathscr{F}\left(\int_0^{\delta\left(\Upsilon^{i}\mu_0, \Upsilon^{i+1}\mu_0\right)} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F}\left(\int_0^{\delta\left(\Upsilon^{i-1}\mu_0, \Upsilon^{i}\mu_0\right)} \varphi(s)\mathrm{d}s\right), \tag{41}$$

for all $i \in N$ *and* $\liminf_{k\longrightarrow l^+}\tau(k) > 0$ *for each* $l \geq 0$. *Suppose also that a mapping* $\Delta \ni \mu \mapsto \delta(\mu, \Upsilon\mu)$ *is* $\check{D}(\Upsilon, \mu_0)$*-dynamic lower semicontinuous. Then,* Υ *has a fixed point.*

Corollary 4. *Let* (Δ, δ) *be a complete metric space,* $\mathscr{F} \in \nabla_F$, $\varphi \in \Phi$, *and* $\Upsilon: \Delta \longrightarrow \Delta$. *Assume that there exists* $\tau: R_+ \longrightarrow R_+$ *such that* $\delta(\Upsilon\mu, \Upsilon^2\mu) > 0$ *implies*

$$\tau(\delta(\mu, \Upsilon\mu)) + \mathscr{F}\left(\int_0^{\delta\left(\Upsilon\mu, \Upsilon^2\mu\right)} \varphi(s)\mathrm{d}s\right) \leq \mathscr{F}\left(\int_0^{\delta(\mu, \Upsilon\mu)} \varphi(s)\mathrm{d}s\right), \tag{42}$$

for all $\mu \in \Delta$ *and* $\liminf_{k\longrightarrow l^+}\tau(k) > 0$ *for each* $l \geq 0$. *Suppose also that a mapping* $\Delta \ni \mu \mapsto \delta(\mu, \Upsilon\mu)$ *is lower semicontinuous. Then,* Υ *has a fixed point.*

Example 3. Let $\Delta = [0, +\infty)$ with the usual metric, Δ constitutes a complete metric space. Consider a mapping $\Upsilon: \Delta \longrightarrow K(\Delta)$ by $\Upsilon(\mu) = [0, \mu/2]$, $\mu > 0$ and $\tau: R_+ \longrightarrow R_+$ by

$$\tau(\mu) = \begin{cases} -\ln\mu, \ \mu \in \left(0, \dfrac{1}{2}\right) \\ \ln 2, \ \mu \in \left[\dfrac{1}{2}, \infty\right) \end{cases} \tag{43}$$

Define dynamic iterative process $\check{D}(\Upsilon, \mu_0)$: a sequence $\{\mu_i\}$ is given by $\mu_i = \mu_0 g^{i-1}$ for all $i \in N$ with initial point $\mu_0 = 2$ and $g = 1/2$ such that

$i \geq 2$	$\mu_i = i_0 g^{i-1}$	$\in$	$\Upsilon\mu_{i-1} = [0, \mu/2]$
$\mu_{i=2}$	1	–	$\Upsilon\mu_{i=1} = [0, 1]$
$\mu_{i=3}$	1/2	–	$\Upsilon\mu_{i=2} = [0, 1/2]$
$\mu_{i=4}$	1/4	–	$\Upsilon\mu_{i=3} = [0, 1/4]$
$\mu_{i=5}$	1/8	–	$\Upsilon\mu_{i=4} = [0,1/8]$

Continuing the above iterative process, we see that

$$\check{D}(\Upsilon, \mu_0) = \left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\right\} \tag{44}$$

is a dynamic iterative process of Υ starting from the point $\mu_0 = 2$. Setting $\varphi(s) = 1$ for all $s \in R$ and $\mathscr{F}(s) = \ln(s)$. For $\mu_i \in \check{D}(\Upsilon, \mu_0)$ and $\widehat{H}(\Upsilon\mu_i, \Upsilon\mu_{i+1}) > 0$, we have

$$\begin{cases} e^{\tau\left(|\mu_{i-1}-\mu_i|\right)+\mathscr{F}\left(\int_0^{\frac{|\mu_{i-1}-\mu_i|}{2}} \varphi(s)\mathrm{d}s\right)} \leq e^{\mathscr{F}\left(\int_0^{|\mu_{i-1}-\mu_i|} \varphi(s)\mathrm{d}s\right)} \\ e^{\tau\left(|\mu_{i-1}-\mu_i|\right)+\ln\left(\int_0^{\frac{|\mu_{i-1}-\mu_i|}{2}} \varphi(s)\mathrm{d}s\right)} \leq e^{\ln\left(\int_0^{|\mu_{i-1}-\mu_i|} \varphi(s)\mathrm{d}s\right)} \\ e^{\tau\left(|\mu_{i-1}-\mu_i|\right)} e^{\ln\left(\int_0^{\frac{|\mu_{i-1}-\mu_i|}{2}} \varphi(s)\mathrm{d}s\right)} \leq e^{\ln\left(\int_0^{|\mu_{i-1}-\mu_i|} \varphi(s)\mathrm{d}s\right)} \\ e^{\tau\left(|\mu_{i-1}-\mu_i|\right)} \int_0^{\frac{|\mu_{i-1}-\mu_i|}{2}} \varphi(s)\mathrm{d}s \leq \int_0^{|\mu_{i-1}-\mu_i|} \varphi(s)\mathrm{d}s \\ \dfrac{|\mu_{i-1}-\mu_i|}{2} \leq e^{-\tau\left(|\mu_{i-1}-\mu_i|\right)}|\mu_{i-1}-\mu_i| \end{cases}, \tag{45}$$

and so

$$\tau\left(U\left(\mu_{i-1},\mu_i\right)\right)+\mathscr{F}\left(\int_0^{\widehat{H}\left(\Upsilon_{\mu_i},\Upsilon_{\mu_{i+1}}\right)}\varphi(s)\mathrm{d}s\right)\leq\mathscr{F}\left(\int_0^{U\left(\mu_{i-1},\mu_i\right)}\varphi(s)\mathrm{d}s\right).\tag{46}$$

Hence, all the required hypotheses of Theorem 1 are satisfied and hence 0 is a fixed point of Υ.

3. An Application

In this frame of study, we deal with some new aspects of Liouville–Caputo fractional differential equations in module of complete metric space. Several earlier developments on fixed point theory and its applications involving fractional calculus can be found in [14].

Define the Liouville–Caputo fractional differential equations based on order κ ($\check{D}_{(c,\kappa)}$) by

$$\check{D}_{(c,\kappa)}(\alpha(g))=\frac{1}{\Gamma(i-\kappa)}\int_0^g(g-t)^{i-\kappa-1}\alpha^{(i)}(t)\mathrm{d}t,\tag{47}$$

where $i-1<\kappa<i$, $i=[\kappa]+1$, $\alpha\in C^i([0,+\infty])$, and the collection $[\kappa]$ represents positive real number and Γ represents the Gamma function. Let $\Delta:=C(I,R)$ be the space of all continuous real-valued functions on I. And, complete metric space δ_ς: $\Delta\times\Delta\longrightarrow[0,+\infty)$ be given by

$$\delta_\varsigma(\varepsilon_1,\varepsilon_2)=\sup_{a\in I}\left|\varepsilon_1(a)-\varepsilon_2(a)\right|.\tag{48}$$

Now, consider the following fractional differential equations and its integral boundary valued problem:

$$\check{D}_{(c,\kappa)}(\beta(g))=L(g,\beta(g)),\tag{49}$$

where $g\in(0,1)$, $\kappa\in(1,2]$ and

$$\begin{cases}\beta(0)=0,\\ \beta(1)=\int_0^{\vartheta}\beta(g)\mathrm{d}g,\ \vartheta\in(0,1),\end{cases}\tag{50}$$

where $I=[0,1]$, $\beta\in C(I,R)$ and L: $I\times R\longrightarrow R$ be a continuous function. Let P: $\Delta\longrightarrow\Delta$ be defined by

$$\mathrm{Pv}(r)=\begin{cases}\frac{1}{\Gamma(\kappa)}\int_0^g(g-t)^{\kappa-1}L(t,v(t))\mathrm{d}t\\ -\frac{2g}{\left(2-\vartheta^2\right)\Gamma(\kappa)}\int_0^1(1-t)^{\kappa-1}L(t,v(t))\mathrm{d}t\\ +\frac{2g}{\left(2-\vartheta^2\right)\Gamma(\kappa)}\int_0^{\vartheta}\left(\int_0^{g_1}(g_1-t_1)^{\kappa-1}L(t_1,v(t_1))\mathrm{d}t_1\right)\mathrm{d}t\end{cases},\tag{51}$$

for $v\in\Delta$ and $g\in[0,1]$. Now, we start the main result of this section.

Theorem 2. *Let L: $I\times R\longrightarrow R$ be a continuous function, nondecreasing on second variable and there is a nonconstant function τ such that $\varepsilon_i\in\check{D}_\varsigma(\Upsilon,\varepsilon_0)$ and $g\in[0,1]$ implies*

$$\left|P\varepsilon_{i-1}(r)-P\varepsilon_i(r)\right|\leq\Omega\frac{U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)}{\left(1+\tau\sqrt{\max_{g\in I}U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)}\right)^2},\tag{52}$$

where $\Omega=(2\kappa-1)(\Gamma(\kappa+1))/2(5\kappa+2)$ and

$$U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)=\max\left\{\begin{array}{l}\left|\varepsilon_{i-1}(r)-\varepsilon_i(r)\right|,\left|\varepsilon_{i-1}(r)-\Upsilon\varepsilon_{i-1}(r)\right|,\left|\varepsilon_i(r)-\Upsilon\varepsilon_i(r)\right|,\\ \frac{\left|\varepsilon_{i-1}(r)-\Upsilon\varepsilon_i(r)\right|+\left|\varepsilon_i(r)-\Upsilon\varepsilon_{i-1}(r)\right|}{2}\end{array}\right\}.\tag{53}$$

Then, equations (49) and (50) has at least one solution on Δ.

Proof. For every $g\in I$ and owing to operator P: $\Delta\longrightarrow\Delta$, one writes

Upon setting, we see that

In the light of above observation, we have

which implies that

$$\left|P\varepsilon_{i-1}(r)-P\varepsilon_i(r)\right|\leq\frac{U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)}{\left(1+\tau\sqrt{\max_{g\in I}U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)}\right)^2}.\tag{54}$$

By above virtue, we have

$$\delta_\varsigma\left(P\varepsilon_{i-1}(r)-P\varepsilon_i(r)\right)=\sup_{a\in I}\left|P\varepsilon_{i-1}(r)-P\varepsilon_i(r)\right|\leq\frac{U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)}{\left(1+\tau\sqrt{\max_{g\in I}U\left(\varepsilon_{i-1},\varepsilon_i\right)(r)}\right)^2}.\tag{55}$$

Furthermore, by contractive condition (13) upon setting of $\varphi(s)=1$ for all $s\in R$ and $\mathscr{F}(s)=-1/\sqrt{s}$, we have

$$\widehat{H}\left(\Upsilon\varepsilon_i,\Upsilon\varepsilon_{i+1}\right)>0\Rightarrow\tau\left(U\left(\varepsilon_{i-1},\varepsilon_i\right)\right)+\mathscr{F}\left(\int_0^{\widehat{H}\left(\Upsilon_{\varepsilon_i},\Upsilon_{\varepsilon_{i+1}}\right)}\varphi(s)\mathrm{d}s\right)\leq\mathscr{F}\left(\int_0^{U\left(\varepsilon_{i-1},\varepsilon_i\right)}\varphi(s)\mathrm{d}s\right),\tag{56}$$

for all $i \in N$, $\varepsilon_i \in \check{D}_{\varsigma}(\Upsilon, \varepsilon_0)$ and for each given $\epsilon > 0$ such that $\int_0^{\epsilon} \varphi(s)\mathrm{d}s > 0$. Thus, all the required hypotheses of Theorem 1 are satisfied, and hence equations (49) and (50) has at least one solution on Δ. □

Example 4. Based upon the Liouville–Caputo fractional differential equations based on order $\kappa(\check{D}_{(c,\kappa)})$. Let us consider the following integral boundary-value problem:

$$\check{D}_{\left(c,\frac{3}{2}\right)}(\beta(g)) = \frac{1}{(g+3)^2}\frac{|\beta(g)|}{1+|\beta(g)|} \tag{57}$$

and

$$\begin{cases} \beta(0) = 0, \\ \beta(1) = \int_0^{3/4} \beta(g)\mathrm{d}g, \ \vartheta \in (0,1), \end{cases} \tag{58}$$

where $\kappa = 3/2$, $\vartheta = 34$, and $L(t, v(t)) = 1/(g+3)^2|\alpha(g)|/1 + |\alpha(g)|$. So, the above setting is an example of equations (49) and (50). Hence, here is clearly the pair of equations (57) and (58) has at least one solution.

4. Conclusions

In this paper, we have investigated the preexisting results of fixed point for set-valued mappings rather than the conventional mappings. Based upon a Wardowski integral and with a nonnegative Lebesque integrable mapping, we have transformed the conventional theorems of fixed point into the module of F_I^C. Instead of the traditional Picard sequence, the dynamic process $\check{D}(\Upsilon, \mu_0)$ is adopted to iterate the fixed point. Afterwards, the results have been explained by rendering concrete examples, and some foremost corollaries have been deduced from the prime results. Also, we provide illustrative applications to Liouville–Caputo fractional differential equations.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All authors contributed equally in writing this article. All authors read and approved the final manuscript.

References

[1] H. Kaddouri, H. Işık, and S. Beloul, "On new extensions of *F*-contraction with an application to integral inclusions," *U.P.B. Sci. Bull. Series A*, vol. 81, no. 3, pp. 31–42, 2019.

[2] A. Ali, H. Işık, H. Aydi, E. Ameer, J. R. Lee, and M. Arshad, "On multivalued Suzuki-type θ-contractions and related applications," *Open Mathematics*, vol. 18, no. 1, pp. 386–399, 2020.

[3] H. Işık and C. Ionescu, "New type of multivalued contractions with related results and applications," *U.P.B. Sci. Bull. Series A*, vol. 80, no. 2, pp. 13–22, 2018.

[4] A. Ali, M. Arshad, M. Arshad et al., "On multivalued maps for $ \varphi $-contractions involving orbits with application," *AIMS Mathematics*, vol. 6, no. 7, pp. 7532–7554, 2021.

[5] A. Branciari, "A fixed point theorem for mappings satisfying a general contractive condition of integral type," *International Journal of Mathematics and Mathematical Sciences*, vol. 29, no. 9, pp. 531–536, 2002.

[6] Z. Liu, J. Li, and S. M. Kang, "Fixed point theorems of contractive mappings of integral type," *Fixed Point Theory and Applications*, vol. 2013, no. 1, p. 300, 2013.

[7] D. Sekman and V. Karakaya, "On the F-contraction properties of multivalued integral type transformations," *Methods Funct.Anal. Topology*, vol. 25, pp. 282–288, 2019.

[8] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fundamenta Mathematicae*, vol. 3, pp. 133–181, 1922.

[9] D. Wardowski, "Fixed points of a new type of contractive mappings in complete metric spaces," *Fixed Point Theory and Applications*, vol. 2012, no. 1, p. 94, 2012.

[10] S. Nadler, "Multi-valued contraction mappings," *Pacific Journal of Mathematics*, vol. 30, no. 2, pp. 475–488, 1969.

[11] J. Vujakovic and S. Radenovic, "On some F-contraction of Piri-Kumam-Dung type mappings in metric spaces," *Vojnoteh. Glas-Tech. Cour.*vol. 68, pp. 697–714, 2020.

[12] D. Klim and D. Wardowski, "Fixed points of dynamic processes of set-valued F-contractions and application to functional equations," *Fixed Point Theory and Applications*, vol. 2015, no. 1, p. 22, 2015.

[13] A. Ali, F. Uddin, M. Arshad, and M. Rashid, "Hybrid fixed point results via generalized dynamic process for F-HRS type contractions with application," *Physica A*, vol. 538, 2020.

[14] F. Jarad, T. Abdeljawad, and Z. Hammouch, "On a class of ordinary differential equations in the frame of Atangana-Baleanu fractional derivative," *Chaos, Solitons & Fractals*, vol. 117, pp. 16–20, 2018.

[15] M. Cosentino and P. Vetroa, "Fixed point results for F-contractive mappings of Hardy-Rogers-type," *Filomat*, vol. 28, no. 4, pp. 715–722, 2014.

[16] E. Kryeyszig, *Introductory Functional Analysis with Applications*John Wiley & Sons, New York, 1978.

[17] B. E. Rhoades, "Two fixed-point theorems for mappings satisfying a general contractive condition of integral type," *International Journal of Mathematics and Mathematical Sciences*, vol. 2003, no. 63, pp. 4007–4013, Article ID 194879, 2003.

[18] M. Sgroi and C. Vetro, "Multi-valued F-contractions and the solutions of certain functional and integral equations," *Filomat*, vol. 27, no. 7, pp. 1259–1268, 2013.

[19] T. Suzuki, "A new type of fixed point theorem in metric spaces," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 11, pp. 5313–5317, 2009.

Hindawi
Complexity
Volume 2022, Article ID 3865820, 22 pages
https://doi.org/10.1155/2022/3865820

Research Article

A Simple Image Encryption Based on Binary Image Affine Transformation and Zigzag Process

Adélaïde Nicole Kengnou Telem,[1,2] **Cyrille Feudjio**,[1] **Balamurali Ramakrishnan**,[3] **Hilaire Bertrand Fotsin**,[2] **and Karthikeyan Rajagopal**[3]

[1]*Department of Electrical and Electronic Engineering, College of Technology (COT), University of Buea, P.O. Box 63, Buea, Cameroon*
[2]*Laboratoire de recherche de Matière Condensée, d'Electronique et de Traitement du Signal (LAMACETS), Département de Physique, Faculté des Sciences, Université de Dschang, Dschang, Cameroon*
[3]*Centre for Nonlinear Systems, Chennai Institute of Technology, Chennai, India*

Correspondence should be addressed to Adélaïde Nicole Kengnou Telem; adelkengnou@gmail.com

Received 20 April 2021; Revised 23 September 2021; Accepted 19 November 2021; Published 7 January 2022

Academic Editor: Padmapriya Praveenkumar

In this paper, we propose a new and simple method for image encryption. It uses an external secret key of 128 bits long and an internal secret key. The novelties of the proposed encryption process are the methods used to extract an internal key to apply the zigzag process, affine transformation, and substitution-diffusion process. Initially, an original gray-scale image is converted into binary images. An internal secret key is extracted from binary images. The two keys are combined to compute the substitution-diffusion keys. The zigzag process is firstly applied on each binary image. Using an external key, every zigzag binary image is reflected or rotated and a new gray-scale image is reconstructed. The new image is divided into many nonoverlapping subblocks, and each subblock uses its own key to take out a substitution-diffusion process. We tested our algorithms on many biomedical and nonmedical images. It is seen from evaluation metrics that the proposed image encryption scheme provides good statistical and diffusion properties and can resist many kinds of attacks. It is an efficient and secure scheme for real-time encryption and transmission of biomedical images in telemedicine.

1. Introduction

The amazing developments in the field of network communications during the past years have created a great requirement for secured image transmission over the Internet [1]. Image data, such as medical images, military images, images of electronic publishing, and fingerprint images from authentication systems, must be kept private and confidential. The confidentiality of these images is capital and cannot be guaranteed through the Internet. To ensure the security of information during transmission, encryption techniques are used. Cryptography aims to ensure data confidentiality, integrity, and authentication during communication. In telemedicine, medical data need to be processed with total discretion. This justifies the use of encryption technology in telemedicine.

Cryptographic methods are based on two fundamental principles, namely, substitution and diffusion. Substitution involves replacing certain letters or values of the original message with others. Diffusion consists of dispersing the position of the letters or the values of the original message to scramble the message. Most encryption techniques have been developed to secure text data. Unfortunately, these techniques are not suitable for images because the images have a rather complex structure and are quite large in size compared to the text data. Yet, the images may contain private and fairly confidential information. Hence, there is a need to find an effective technique to secure images. Thus, designing less complex image encryption algorithms becomes very crucial. Much research has focused on the issue of image security. Digital images have several properties

such as information redundancy, a strong correlation between pixels, and large data capacities. An image encryption algorithm must take into account all these properties.

Conventional image encryption techniques are divided into two groups, namely, asymmetric encryption (with private and public keys) and symmetric encryption (with a secret key) [2]. Several algorithms have been proposed in the past for the encryption of images. We have data encryption standard (DES), triple data encryption algorithm (TDEA), advanced encryption standard (AES), Rivest, Shamir, and Adleman (RSA), and fast image encryption algorithm (FEAL) [2–8]. In recent years, chaotic systems have been used by many researchers in image encryption. A chaos-based image encryption technique typically uses both substitution and diffusion processes. This technique generally uses an external key and one or more generators to generate chaotic sequences that will be used for the substitution/diffusion process. Ahmad and Farooq in [9] approved that the generation of high-quality key stream decides the level of security offered by the Cipher. They combined the simple logistic map with the cubic map to generate PN sequences for the proposed encryption scheme. Those PN sequences from the proposed generator have good autocorrelation and have been tested randomness. Li et al. in [10] introduced a performance-enhanced image encryption scheme based on depth-conversion integral imaging and hybrid cellular automata (CA). The aim is to meet the requirements of secured image transmission. The input image is firstly decomposed into an elemental image array (EIA) using the depth-converted integral imaging technique. The CA model and chaotic sequence are used to encrypt the elemental images. In [11], Ahmad et al. used particle swarm optimization and chaotic map to propose an optimized image encryption. Initial conditions of the chaotic map depend on the pending plain image, so the algorithm is resistant because, from one image to another, key stream will be different. Niu et al. in [12] proposed an efficient method for image encryption based on the chaos theory and a deoxyribonucleic acid (DNA) sequence database using the characteristics of chaos, such as randomness, regularity, ergodicity, and initial value sensitiveness, combined with the unique space conformation of DNA molecules and their unique information storage and processing ability. In [13], Kengnou et al. proposed an encryption algorithm using a 3D chaotic system and DNA coding. Two keys are used, an internal one and an external one. During the chaotic process, the sequences generated from the different variables of the chaotic generator are not used separately. They are combined using the zigzag process and used simultaneously. Each 3D chaotic system can be used. Twenty four DNA rules and 16 join operations of DNA coding are used during the DNA coding process. A Fast Walsh Transform (FWT) has been combined with two chaotic logistic maps by Telem et al. in [14] to propose a new scheme of image encryption. Chaotic encryption methods are combined with the two-dimensional FWT of images. Liu and Miao in [15] proposed an image encryption method based on a logistic chaotic map and dynamical algorithm. In their method, the parameter of the logistic map is varied and used to shuffle the plain image. Then, the dynamical algorithm comes to encrypt the image. Many other chaotic encryption methods have been proposed in [15–23].

Cryptanalysis evaluates the efficiency of cryptosystems to resist against attacks and therefore confirms their validity. Several studies have shown that some cryptosystems based on chaos could be cryptanalyzed [24–27]. To face this challenge, Pareek et al. in [28] proposed an image encryption system without chaos. From a 128-bit external key, it provides an efficient algorithm using 16 rounds with satisfactory results. In [29], Jolfaei et al. have demonstrated the shortcomings of the image encryption scheme proposed by Pareek et al. [28]. The method is not secured, and the secret key can be deduced by a chosen-ciphertext attack. Security flaws of the encryption scheme have been discussed and solutions have been proposed in [29]. Houas et al. proposed in [30] a new algorithm to encrypt binary images based on several steps. Firstly, they reduced the amount of data required to present the image. The next step consists to divide the image into d blocks. Those blocks are used to construct a new image of the same size as the original one but represented on a new basis. The construction of the new basis is inspired by the work of Mokhtari and Melkemi [31]. After that transformation, they obtained a key image and used it to encrypt and decrypt images. The encrypted image is the representation on a new basis. The decryption algorithm consists of subtraction between each encrypted image and the key image and the sum of them.

Some image encryption methods are based on mathematical transform such as cosine transform, and Fourier transform is proposed. The method proposed by Lima et al. in [32] is based on the cosine number transform (CNT), a mathematical tool whose application requires modular arithmetic only. A CNT is very sensitive to changes in the vector being transformed, so 2 slightly different vectors may have significantly distinct CNTs, which are desirable for cryptography applications [32]. The method consists of dividing an image into blocks that overlap horizontally and vertically the corresponding adjacent block. The blocks are then sequentially taken and submitted to the recursive computation of a two-dimensional CNT. The method is limited to noncompressed images and particularly to medical images complying with the Digital Imaging and Communications in Medicine (DICOM) standard. Lima et al. in [33] proposed a fast computation of Cosine transform over fields of characteristic 2 and the application to image encryption. Annaby et al. in [34] have proposed a cryptosystem based on Fourier transform.

In [35–38], methods to reduce the workload of the time-consuming diffusion part are proposed. The encryption process is taken over the entire image. The image is not divided into many subblocks. Those methods gain in execution time but have several security problems.

In [39–41], affine transformations combined with other mathematical functions have been used to propose image encryption methods. Zhu et al. [39] have used affine cipher and generalized Arnold map to propose an image encryption scheme. Ahmad and Hwang in [40] have combined affine transformation with the chaotic map to propose their encryption method. In [41], Shah et al. have combined affine transformation with linear fractional transformation.

In this work, affine transformations are not combined with other mathematical models or functions. We propose a new image encryption algorithm using an external key, an internal key, affine transformations (reflections and rotations), zigzag process, and substitution-diffusion processes. Each pixel of the plain image is converted into its equivalent 8-bit binary, and the bits of rank n ($n = 1, 2, 3, 4, 5, 6, 7, 8$) of the different pixels constitute the binary image of this rank. So, the plain image is converted into 8 binary images. From binary images, an internal key is deduced. Each binary image is submitted to the zigzag process to yield zigzag binary images. Using an external key, the zigzag binary images are mapped using affine transformations and a new image is reconstructed. The reconstructed image is then submitted to substitution and diffusion processes to produce a cipher image. The novelties of this work are

(i) The use of the two keys: an internal key is extracted from the binary components of the plain image to be encrypted. In [13], an internal key is extracted directly from the pixel of plain image and not from the binary components of the plain image. In this work, an internal key is used in combination with the external key to generate the substitution and diffusion sequences. Internal keys are different from one plain image to another, and consequently, the substitution and diffusion sequences are also different even in the case of the same external key. To decrypt the cipher image, one must have the exact internal key and external key and well-known algorithm used in the cryptosystem.

(ii) The proposed algorithm does not use chaotic generators or complex mathematic functions to compute the substitution-diffusion sequences. Rather, those sequences are computed from simple logical bit operation by using external and internal keys and the part of the previous result of the encryption process.

(iii) The application of affine transformation on the binary component of the plain image.

(iv) The method to apply zigzag process: depending on the context, it can act as a diffusion or substitution process. Previous works have applied the zigzag process on the gray-scale image. The proposed method applied the zigzag process on the binary version of the plain image. So, it acts as a substitution process.

Hence, the proposed method does not gain on execution time but gain more on efficiency, security, and robustness. The proposed method is very helpful in telemedicine to secure medical images before transmitting them from one hospital to another. The method can also be used in every domain where we need to secure images as military domain and so on.

In the rest of the work, we present the details of our encryption algorithm in Section 2. The experimental results and security analysis tests are given in Section 3. At the end of the work is a conclusion.

2. Proposed Cryptosystem

2.1. Block Diagram of the Proposed Algorithm. In this work, we used an external key of 128 bits long, an internal key, reflection and rotation mappings, zigzag process, substitution, and diffusion processes to propose a new image encryption method. The plain image is converted into 8 binary images. From binary images, an internal key is extracted. Each binary image is submitted to the zigzag process to yield zigzag binary images. Using an external key, the zigzag binary images are mapped and a new image is reconstructed. This new image is divided into nonoverlapping squared subblocks. Each subblock is then submitted to the substitution and diffusion processes for K round using the combination of the two keys. The originality of this method dwells on the method used to generate the internal key, the use of reflection or rotation mapping, and the method to apply the substitution and the zigzag processes, the method used to compute substitution and diffusion sequences. The internal key is provided by the image being encrypted. In [28], the zigzag operation is applied on the pixel values, and it acts to change just the position of the pixel or the pixel location in the subblock. In this work, the whole image or the considered subblock is converted into binary images or binary subblock. The bits in a binary image/subblock are then reshuffled within the image/block by a zigzag path, and a new image/subblock is reconstructed. Consequently, the zigzag process changes the value of the pixels and acts as a substitution process. After the zigzag process, an external key is used to choose the corresponding type of transformation to be applied on each zigzag binary image. In [39–41], authors combine affine transformations with other mathematical models or functions. In this work, we do not use any chaotic generator or mathematical function to generate the codes used for the substitution-diffusion processes as usually. Rather, we combine the two keys (internal and external) and pixels of the image to compute the codes used on substitution-diffusion processes. Those features are the particularities of this algorithm. The block diagram of the proposed algorithm is shown in Figure 1.

2.1.1. External and Internal Keys. The proposed algorithm uses an external secret key of thirty-two hexadecimal numbers as shown in this example: «**ABCDEFGHI JKLMNOPRSTUVW*αβγηθλξρτφ***». This key is used for both substitution-diffusion processes and the choice of the type of affine transformation to be applied on binary images. An original image I_{mxn} is decomposed into 8 binary images Ibi_{mxn} ($i = 1, 2, \ldots, 8$). Ibi_{mxn} are submitted to "XOR" operation and yield one binary matrix. This matrix is then converted into a pixel matrix. The "XOR" operations between lines of the matrix produce the first part of the internal key named "keyi1," and while the "XOR" operations between columns produce the second part of the internal key named "keyi2." Table 1 presents the internal keys of two images 'ANTAMOEBACOLI' and 'Lena,' respectively. One can see the difference between the two internal keys. So, internal keys are different from image to another. "keyi1"

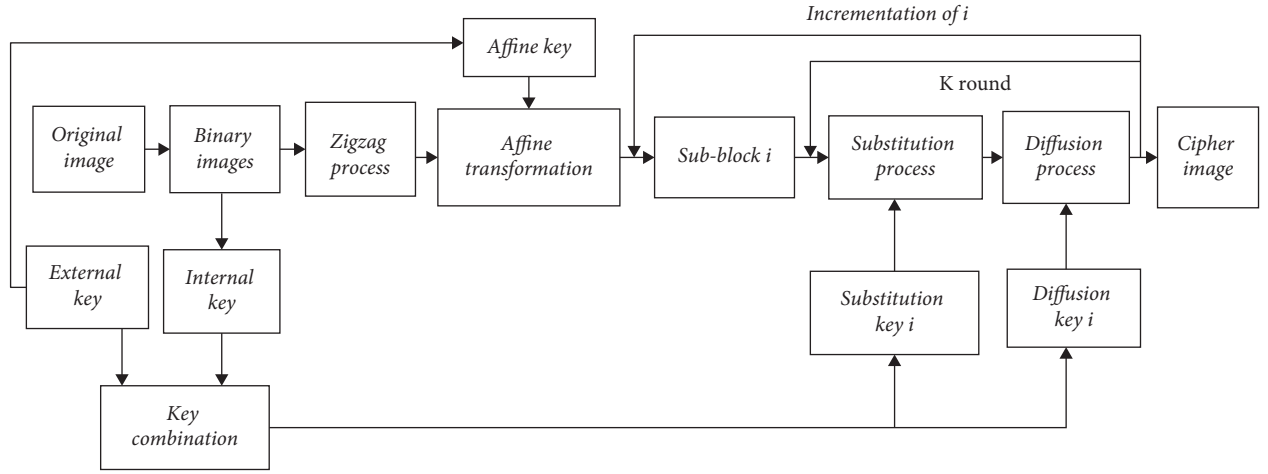


Figure 1: The block diagram of our proposed algorithm.

Table 1: Internal keys of two images: ANTAMOEBACOLI and 'Lena.'

78	188	117	104	223	128	20	173	163	149	88	177	175	200	20	226
129	69	76	140	81	116	102	113	169	172	48	18	59	53	221	31
65	64	217	252	193	182	106	40	4	45	101	69	209	255	63	181
92	174	248	130	243	120	29	157	150	246	46	245	134	62	174	142
158	34	176	210	11	152	156	2	78	50	54	71	55	220	66	248
45	194	164	89	70	238	221	204	56	57	58	43	223	104	67	249
143	234	255	50	51	52	53	54	60	61	207	64	65	68	70	72
55	56	57	58	59	60	61	62	73	74	75	76	77	79	80	81
Keyi1 of ANTAMOEBACOLI image										Keyi2 of ANTAMOEBACOLI image					
138	97	232	153	162	85	200	238	115	47	160	107	94	224	6	152
94	245	231	183	50	17	213	186	120	102	48	76	237	84	246	154
253	230	53	118	255	202	59	218	77	255	168	114	20	176	204	53
117	195	102	64	39	112	25	133	143	11	70	232	75	79	156	78
119	154	69	206	73	95	24	244	18	95	140	147	141	101	184	127
91	45	21	122	101	176	63	211	100	106	221	87	98	13	124	178
108	103	208	216	31	30	90	190	16	80	5	138	144	89	81	58
224	242	37	239	126	112	10	180	208	171	142	34	199	22	128	82
Keyi1 of lena image										Keyi2 of lena image					

and "keyi2" are combined with an external key to generate substitution-diffusion sequences, and consequently, substitution-diffusion sequences are also different for one image to another, and this makes the algorithm high secured and resistant against attacks.

2.1.2. Zigzag Process. The zigzag process aims to scramble the image by changing the position of the pixel or the bit. Depending on the context, it can act as a diffusion or substitution process. In this algorithm, the zigzag process applied to the whole binary image acts as a diffusion process. In the case of the binary subblock, it acts as a substitution process. A zigzag process to scramble the binary images is shown in Figure 2.

2.1.3. Affine Transformation Process. On the Euclidean plane, let w: $R^2 \longrightarrow R^2$ be an affine transformation; its equation is given by

$$\begin{aligned} w(x, y) &= (ax + by + e, cx + dy + f) \\ &= (x', y'), \end{aligned} \tag{1}$$

where (x, y) is the coordinate point, a, b, c, d, e, and f are real numbers, and (x', y') is the new coordinate point. We can also write this same transformation with the equivalent notations:

$$\begin{aligned} w(u) &= w\begin{pmatrix} x \\ y \end{pmatrix} \\ &= \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \\ &= \begin{pmatrix} x' \\ y' \end{pmatrix} \\ &= Au + T, \end{aligned} \tag{2}$$

where A is a 2×2 real matrix and $T = \begin{pmatrix} e \\ f \end{pmatrix}$ represents translations.

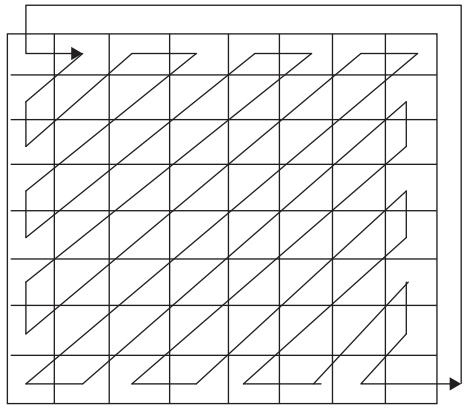


Figure 2: A zigzag path to scramble the binary subblock.

The matrix A can also be written in the form of

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 \cos\theta_1 & -r_2 \sin\theta_2 \\ r_1 \sin\theta_1 & r_2 \cos\theta_2 \end{pmatrix}, \tag{3}$$

where (r_1, θ_1) are the polar coordinates of the point (a, c) and $(r_2, (\theta_2 + (\pi/2)))$ are the polar coordinates of the point (b, d). In other words,

$$r_1 = \sqrt{a^2 + c^2}, \quad \tan\theta_1 = \frac{c}{a}, \qquad r_2 = \sqrt{b^2 + d^2}, \quad \tan\theta_2 = \frac{b}{d}. \tag{4}$$

Various transformations can be performed in R^2 such as dilations, reflections, translations, rotations, and similitudes. In this work, we are restricted to reflection and rotation transformations.

A reflection on the x-axis can be written in as $w_{rx}(x, y) = (x, -y)$, while a reflection on the y-axis is written as $w_{ry}(x, y) = (-x, y)$. In matrix representation, these reflections are given by

$$(x', y'), \qquad w_{ry}(u) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{5}$$

A rotation mapping has the form $w_r(x, y) = (x \cos\theta - y \sin\theta, x \sin\theta + y \cos\theta)$.

Also, it is expressed as

$$w_r(u) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{6}$$

For the rotation angle θ, $0 \le \theta < 2\pi$.

The available reflection and rotation transformations are

(0) Identity (I):

(1) Orthogonal reflection around midvertical axis of block (Rmv)

(2) Orthogonal reflection around midhorizontal axis of block (Rmh)

(3) Orthogonal reflection around the first diagonal of block (Rfdiag)

(4) Orthogonal reflection around the second diagonal of block (Rsdiag)

(5) Rotation around the center of block $+90^\circ$ $(R + 90)$

(6) Rotation around the center of block $+180^\circ (R + 180)$

(7) Rotation around the center of block $-90^\circ (R - 90)$

In the algorithm step, we use zigzag binary matrices named $Ibzi_{mxn}$, $i = 1, 2, \ldots, 8$. An external key is used to choose the corresponding type of transformation to be applied on each zigzag binary image $Ibzi_{mxn}$. Let «**ABCDEFGHIJKLMNOPRSTUVW*αβγηθλξρτφ***» be an external key; to choose the corresponding affine transformations, we extract eight numbers «**D, H, L, P, U, *β*, *λ*, *φ***» from an external key. Each number is converted into its corresponding octave number. The obtained octave number gives the corresponding transformation (from 0 to 7 as describe previously) to be applied on the corresponding zigzag binary image. An external key should be chosen such as all these obtained octave number must not be null. At the end of the affine transformation process, the transformed binary images are used to reconstruct a new gray-scale image I'_{mxn}. This image I'_{mxn} is subdivided into nonoverlapping squared block before the substitution-diffusion process.

2.1.4. Substitution Process. The operation is taken into two steps, zigzag operation and substitution, using the corresponding key. The chosen subblock SI'_{kxk} is converted into a binary subblock SbI'. The zigzag operation is taken on SbI', and it is reconverted into the pixel value. A zigzag path to scramble the binary block is shown in Figure 2. The next step is to combine an external key with «*keyi1*» to generate the sequence which will be used for substitution operation. Let L be the size of a subblock. Having an external key «*externalkey*» and an internal key «*substitutioncode*», the algorithm to generate the sequence of each block is defined as follows:

(a) For the first round and the first subblock,

```
Lon = length (keyi1);
lng = Lon/2;
codesub = keyi1;
cc1 = bitxor (externalkey, codesubu (1 : lng));
cc2 = bitxor (cc1, codesub ((lng + 1) : Lon));
usingcode = [cc1 cc2];
```

(b) For the next round K and the other subblocks,

```
codesubu = usingcode;
c1 = bitxor (clefdecfin, codesubu ((lng + 1) : Lon));
cc1 = bitxor (c1, vectsub (1 : 1 : lng));
c2 = bitxor (cc1, codesubu (lng:−1 : 1));
cc2 = bitxor (c2, vectsub ((lng + 1) : Lon));
usingcode = [cc1 cc2];
```

« vectsub » is the last result.

The sequence «usigncode» will be used for the substitution process of the corresponding subblock. The second step of the substitution process is performed using «XOR» operation.

2.1.5. Diffusion Process. The subblock coming from the substitution process is then sent through the permutation process. This process will modify the pixel location. The same algorithm used during the substitution process is used to generate the sequence for the diffusion process in which « keyi2» is used. The sequence of each subblock is rearranged in the ascending order and used to modify the position of the pixel in the subblock.

2.1.6. The Overall Proposed Encryption Algorithm. Let I_{mxn} be an original image. We can describe our encryption algorithm as follows:

Step 1: generate an external secret key.

Step 2: convert image Ibi_{mxn} images.

Step 3: extract an internal secret key.

Step 4: apply zigzag process on binary images.

Step 5: select and apply affine transformation on each zigzag binary images using the external key.

Step 6: reconstruct a new gray-scale image using transformed zigzag binary images.

Step 7: subdivide a new image into many nonoverlapping squared subblock.

Step 8: for each subblock and for K round,

Step 8.1: convert each pixel into the binary vector.
Step 8.2: apply scan zigzag operation to permute the position of the binary element in the binary matrix.
Step 8.3: reconstruct a gray-scale subblock.
Step 8.4: combine an external key and an internal key to generate the substitution sequence of the subblock.
Step 8.5: take out a second part of the substitution process.
Step 8.6: combine an external key and an internal key to generate the permutation sequence of the subblock.
Step 8.7: take out a diffusion process.

The decryption algorithm is the inverse operations of the encryption process.

2.2. Evaluation Metrics. A robust and good cryptosystem should present many features. Firstly, the key space should be large enough to make the brute-force attack infeasible [17]. Secondly, the histograms of the cipher image should be uniformly distributed. The correlation between adjacent pixels (vertically, horizontally, and diagonally) in the cipher image should be approximately zero to confirm the effectiveness of the method.

2.2.1. Correlation Coefficient. The correlation metric is used to evaluate the similarity between two images. If the images are identical, the correlation value is equal to one. When the correlation value is closed to zero, there is no similarity between these images. For an efficient encryption scheme, the correlation between plain image and cipher image must be close to zero. The correlation coefficient (Co) between original and encrypted images is defined as follows:

$$Co = \frac{N_p \sum_{j=1}^{N_p}\left(x_j \times y_j\right) - \sum_{j=1}^{N_p} x_j \times \sum_{j=1}^{N_p} y_j}{\sqrt{N_p \sum_{j=1}^{N_p} x_j^2 - \left(\sum_{j=1}^{N_p} x_j{}^2\right) \times \left(N_p \sum_{j=1}^{N_p} y_j^2 - \left(\sum_{j=1}^{N_p} y_j\right)^2\right)}}, \tag{7}$$

where x and y are gray-scale pixel values of the original and encrypted images and N_p is the total number of pixels.

The correlation coefficient γ of each pair of adjacent pixels is calculated using [42]

$$\gamma(x, y) = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)\sqrt{D(y)}}}, \tag{8}$$

where

$$\operatorname{cov}(x, y) = \frac{1}{N_{ap}} \sum_{i=1}^{N_{ap}} [x_i - E(x)][y_i - E(y)], \tag{9}$$

$$E(x) = \frac{1}{N_{ap}} \sum_{i=1}^{N_{ap}} x_i, \tag{10}$$

$$D(x) = \frac{1}{N_{ap}} \sum_{i=1}^{N_{ap}} [x_i - E(x)]^2. \tag{11}$$

In equations (8)–(11), x and y are the gray values of two adjacent pixels in the image and N_{ap} is the total number of adjacent pairs of pixels. $E(x)$ is the expectation of variable x, $D(x)$ is the variance, and $\operatorname{cov}(x, y)$ is the covariance of two adjacent pixels in the image. For a good cryptosystem, the correlation coefficient Co between plain and cipher images should be close to zero. Likewise, the correlation coefficient γ of each pair of adjacent pixels in the cipher image should be close to zero.

2.2.2. Entropy Information. The information entropy, introduced by Shannon, is one of the most important features of randomness. It is used to evaluate the quantity of information in the image. Information entropy $H(s)$ is calculated in [28] using

$$H(s) = \sum_{i=0}^{N_{gl}-1} P(s_i) \log_2\left(\frac{1}{p(s_i)}\right), \tag{12}$$

where N_{gl} is the total number of gray levels in the image and $P(s_i)$ shows the probability of appearance of the symbol s_i. The entropy value of encrypted images should be closed to eight.

The (k, TB)-local Shannon entropy with respect to local image blocks may be computed by the following steps [43]. First, nonoverlapping image blocks S1, S2, ... Sk with TB pixels for a test image S are randomly selected. Then,

information entropy $H(S_i)$ for all image blocks via equation (12) may be obtained. Finally, the local Shannon entropy over these k image blocks is computed using

$$H_{k,T_B}(m) = \sum_{i=1}^{k} \frac{H(S_i)}{k}. \quad (13)$$

2.2.3. Differential Attacks. The change of a single pixel on a plain image should have an important effect on the cipher image. Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are used to test the influence of changing a single pixel in the original image on the whole encrypted image [17]. Therefore, if $A(i, j)$ and $B(i, j)$ are the pixels in row i and column j of the encrypted images A and B, with only one-pixel difference between the respective plain images, then the NPCR is calculated by using the following formula [44]:

$$\text{NPCR} = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\%, \quad (14)$$

where W and H are the width and height of A or B. $D(i, j)$ is calculated as follows:

$$D(i, j) = \begin{cases} 1, & \text{if } A(i, j) \neq B(i, j), \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

UACI is calculated by the following formula:

$$\text{UACI}(A, B) = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|A(i, j) - B(i, j)|}{255}\right] \times 100\%. \quad (16)$$

The estimated score for NPCR and UACI are 99.6094% and 33.4635% for gray-scale images [45].

2.2.4. Image Quality Criterion. After the encryption/decryption processes, we need to evaluate the performance of the cryptosystem and the quality of images. This is done by evaluating the mean square error (MSE), the peak signal-to-noise ratio, (PSNR) and the encryption quality.

Let P and P', respectively, be a plain image, an encrypted image, and a decrypted image. MSE is defined as follows:

$$\text{MSE} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} [P(m, n) - P'(m, n)]^2}{M \times N}, \quad (17)$$

where M is the total number of lines in the image and N is the total number of columns. The PSNR is defined as follows:

$$\text{PSNR} = 10\text{Log}_{10}\left(\frac{255^2}{\text{MSE}}\right). \quad (18)$$

When the decrypted image P' is identical to the plain one, MSE is zero, and consequently, PSNR is infinite.

The total changes in pixels' values between the plain image and the encrypted image allow to confirm the encryption quality. So, encryption quality gives us the average number of changes to each gray level between plain image and its corresponding encrypted image. It is defined as follows:

$$\overline{\text{Encryption quality}} = \frac{\sum_{L}^{255} |H_L(C) - H_L(P)|}{256}, \quad (19)$$

where C denotes the encrypted image, L is the gray level, $L = 0, 1, 2, \ldots 255$, and H_L is the total number of occurrence of L in the image.

3. Experimental Results

In this work, we used medical and nonmedical images from different databases. Some of the medical images (parasite images) have been taken from different hospital laboratories. The others are from [46, 47]. Nonmedical images are from [48]. Azafack and Guefack images have been taken with a Smartphone techno-Y4. Our algorithm should be carried out using MATLAB R2014a in COMPAQ Intel ® core ™ i3-2328M CPU @ 2.20 GHz 2.20 GHz. The time for encrypting/decrypting an image of 512×512 is 1.1 s. The times for encrypting/decrypting of many images are presented in Table 2.

3.1. Visual Test. To appreciate the effect of the zigzag and affine transformation processes, we present in Figure 3 some plain images and their corresponding transformed images. Physically, these processes have destroyed the correlations between the adjacent pixels in the plain images. Figure 4 presents some encrypted and decrypted images using our proposed cryptosystem where the external key is "A23C56789ABADEF7167DEAB6789367A9». The size of the subblock is 4×4 pixels and the number of rounds on one subblock is five. It is obvious from visual inspection of Figures 4(a)–4(h)) that there is no correlation between the original image and encrypted image. It is therefore impossible, by observing the encrypted image, to deduce the original image. This ensures the physical privacy of the cryptosystem. When observing original and decrypted images in Figures 4(a), 4(c), 4(d), 4(f), 4(g), and 4(i)), it is obvious that the image decrypted is similar to the original image. This test was performed on many other images using different keys, and all the results were conclusive. Thus, visually, the efficiency of the cryptosystem is guaranteed. Thereafter, we conducted a statistical analysis in order to confirm the results of the visual tests.

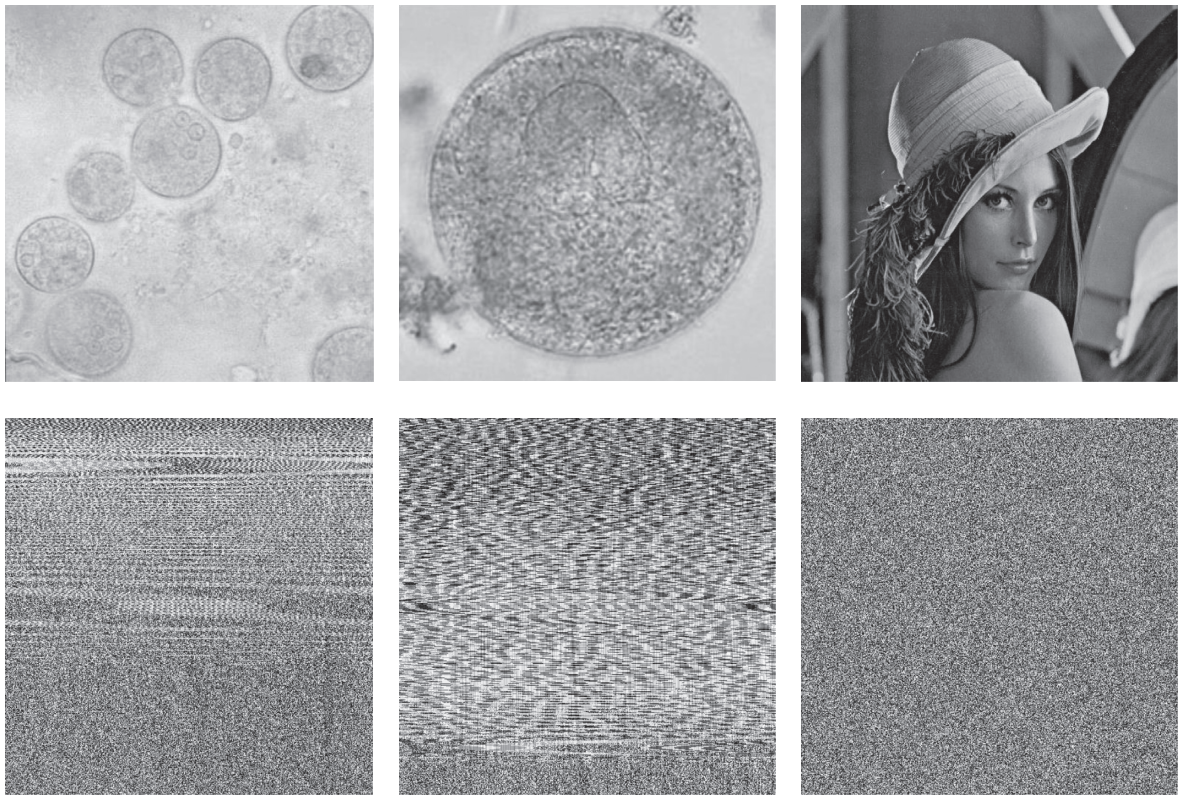
3.2. Key space Analysis. The key space of a good image encryption algorithm should be large enough to make any brute-force attack ineffective [17, 42]. The proposed algorithm used two keys. Firstly, an external key of 128 bits is long; thus, the cipher image has 2^{128} different combinations of the secret key. Secondly, we use an internal secret key of 32 or 64 gray values which are long coming from the decomposition of an original image. So, the size of the key is large enough.

3.3. Correlation Tests

3.3.1. Correlation between Plain and Cipher Image. In Table 3, the correlation coefficients between plain and cipher images of several medical gray-scale images are given. It is

Table 2: Encryption/decryption time.

Image size	Encryption/decryption time (second)
512×512	1.1
256×256	0.553
128×128	0.289
64×64	0.104



(a) (b) (c)

Figure 3: Effect of zigzag and affine transformation processes on some images using the external key «A23C56789AB-ADEF7167DEAB6789367A9»: (a) "Antamoebacoli" and its corresponding transformed image. (b) A plain image "Balantidum Coli cyst" and its corresponding transformed image. (c) A plain image "Girl (Lena, 4.2.04)" and its corresponding transformed image.

observed that all the correlation coefficients are negligible. The highest value (0, 00348) is obtained for the "*ossify*" image. The cipher images are not correlated with plain images. The correlation between the decrypted and the original images is always "1" confirming that both images are identical. The algorithm has been applied on other types of nonmedical images. We have used the USC-SIPI image database which is a collection of the digitized image available and maintained by the University of Southern California [42]. Table 4 shows the results of those images. The same as for medical images, correlation coefficients of images in Table 4 are closed to zero. The maximum value (−0, 00504) of the correlation coefficient is very low compared to the critical value (1). This confirms that the proposed algorithm is efficient for every type of image.

3.3.2. Correlation of Adjacent Pixels. Table 5 shows the correlation coefficients between two vertically, two horizontally, and two diagonally adjacent pixels in several medical plain images and also in their corresponding encrypted images. A high correlation is noted between vertically, horizontally, and diagonally adjacent pixels of original images. The lower value (0.67636) is obtained between diagonal adjacent pixels on the image «ANTAMOEBACOLI». For encrypted images, these values are approximately zero showing that two vertically, two horizontally, and two diagonally adjacent pixels of encrypted images are not correlated. The highest value (−0, 00514) is obtained in the case of medical images «article_*oeuf_taeniaC2*». This is a significant feature proving the effectiveness of our cryptosystem. The same observation is made in the

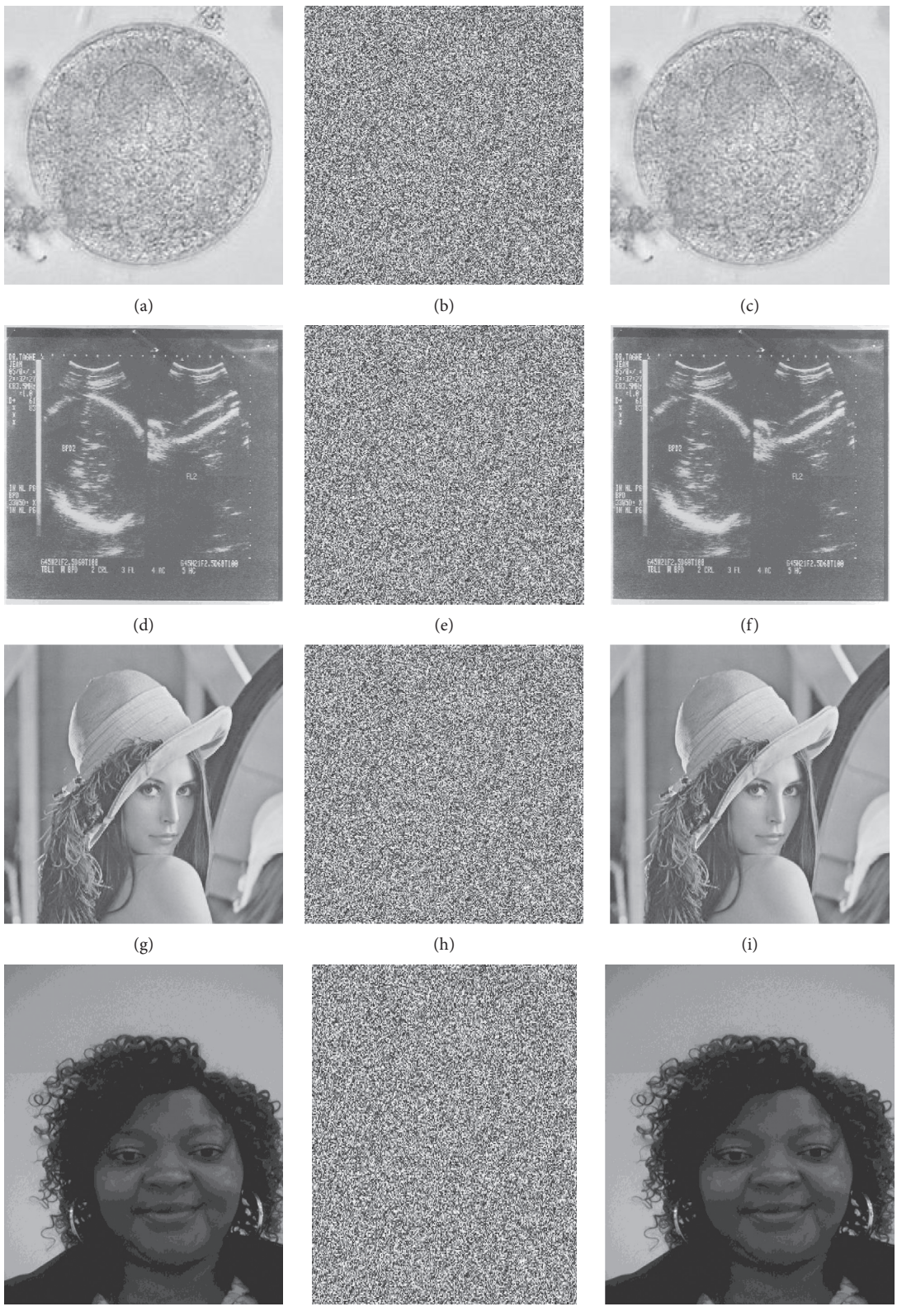


Figure 4: Visual test on some images using the secret key «A23C56789ABADEF7167DEAB6789367A9»: (a), (b), and (c) A plain image "Balantidum Coli cyst" and its corresponding cipher and decrypted image, respectively. (d), (e), and (f) A plain image "echopelv" and its corresponding cipher and decrypted image, respectively. (g), (h), and (i) A plain image "Girl (Lena, 4.2.04)" and its corresponding cipher and decrypted image, respectively. (j), (k), and (l) A plain image "Guefack" and its corresponding cipher and decrypted image, respectively.

Table 3: Correlation coefficients and entropy information of some medical images.

Image name	Size	Correlation coefficients between plain and cipher images	Entropy value of plain image	Entropy value of cipher image	Correlation coefficients between plain and decrypted images
ANTAMOEBACOLI	398×407	$-1.27e{-}03$	6.9714	7.9993	1
article_oeuf_taeniaC2	200×200	$3.23\ e{-}03$	7.2348	7.9993	1
Balantidium coli cyst	200×200	$-2.54e{-}03$	7.0118	7.9993	1
Balantidium coli_trophozoite	200×200	$-1.76e{-}03$	5.6742	7.9992	1
DICROCOELIUM	400×341	$1.66e{-}03$	6.6175	7.9994	1
Entamoeba coli trophozoite	200×200	$-9.34e{-}04$	7.1234	7.9992	1
Entamoeba histolytica cyst	200×200	$-1.08e{-}03$	5.8102	7.9994	1
Entamoeba histolytica-cyst-Gini	130×130	$1.58e{-}03$	6.5016	7.9993	1
Entamoeba histolytica trophozoite	200×200	$8.49e{-}04$	7.6217	7.9994	1
Entamoeba histolytica trophozoite_redim	120×120	$-3.26e{-}03$	7.8522	7.9992	1
Entamoeba histolytica trophozoite_redim2	172×160	$7.73e{-}05$	7.7367	7.9993	1
oeuf_ascarisc	266×200	$-1.48e{-}04$	6.9107	7.9993	1
S- Hematobium egg	400×300	$-1.03e{-}03$	7.1983	7.9992	1
S- Mansoni egg	400×300	$9.58e{-}04$	6.4561	7.9993	1
tropho_entamoeba_histolytica2	332×213	$-1.99e{-}03$	3.8234	7.9993	1
tropho_iodamoeba_butschlii	200×200	$7.70e{-}04$	6.3247	7.9993	1
Angio	64×64	$-3.21e{-}03$	7.2769	7.9992	1
DisLocElbow	64×64	$9.58e{-}04$	5.5326	7.9993	1
echo1	64×64	$1.61e{-}03$	6.3281	7.9993	1
I1_200	64×64	$-7.02e{-}04$	6.4746	7.9993	1
Node2	64×64	$-8.81e{-}04$	6.8732	7.9994	1
Ossify	64×64	$3.48e{-}03$	6.9989	7.9993	1
Pelvis	64×64	$1.29e{-}03$	6.4653	7.9993	1
Ribs	64×64	$1.12e{-}03$	6.2298	7.9991	1
Dirofilaria	241×500	$-2.59e{-}03$	6.7666	7.9993	1
Headirm	256×256	$1.38e{-}03$	5.0299	7.9993	1
Abdomenirm	256×256	$-6.18e{-}04$	6.9551	7.9993	1
Pelvisirm	256×256	$1.08e{-}03$	6.7379	7.9993	1
Gastrointestinal_parasites	512×392	$5.58e{-}07$	7.6634	7.9994	1
Echo fetus at 12 weeks	300×210	$3.41e{-}04$	6.4062	7.9993	1
Ultrasound of fetus of 3months	512×396	$1.90e{-}03$	7.2999	7.9994	1
Echopelv	601×711	$-6.74e{-}04$	6.5896	7.9998	1
CT-MONO2-8-abdo	128×128	$-4.3347e{-}04$	5.8925	7.9992	1
OT-MONO2-8-colon	128×128	$-2.7e{-}03$	6.7468	7.9993	1

case of nonmedical images in Table 6. The proposed cryptosystem produces encrypted images completely different from original images. In Figure 5, we have presented the distribution of horizontally and vertically adjacent pixels in the plain image and its corresponding encrypted image.

3.4. Histograms. The histograms of both the plain and the encrypted images using our proposed method are shown in Figure 6. The histograms of the encrypted images are uniformly distributed while those of plain images are not. This confirms the toughness of the method over any statistical attack. The security of the encryption method is therefore very strong.

3.5. Key Sensitivity Test. An ideal cipher image should be extremely sensitive with respect to the key used in the algorithm during the encryption and decryption processes.

3.5.1. Key Sensitivity Test of Encryption Process. An insignificant change in the encryption key should be sensitive to the cipher images. We take out the key sensitivity test during the encryption process by using different keys to encrypt the same image. The difference between keys is a single bit change. The procedure is described as follows:

(a) We used «A23C56789ABADEF7167DEAB6789367A9» as first external key to encrypt«S-Hematobium egg» image. The encrypted image is presented on Figure 7(b).

(b) We change a single bit on the external key by substituting the fourth character "C" into "D." The external key in this case becomes «A23D56789ABADEF7167DEAB6789367A9». The obtained encrypted image is presented on Figure 7(c).

(c) In the second case, we change the sixteenth character 7 in the external key into 6. The used external key is

Table 4: Correlation coefficients and entropy information of some nonmedical images.

File name	Description	Size	Correlation coefficients between plain and cipher images	Entropy value of plain image	Entropy value of cipher image	Correlation coefficients between plain and decrypted images
4.1.01	Girl	256 × 256	3.46*e*−03	7.1835	7.9994	1
4.1.02	Couple	256 × 256	1.05*e*−03	6.5007	7.9993	1
4.1.03	Girl	256 × 256	6.15*e*−04	5.9549	7.9994	1
4.1.04	Girl	256 × 256	−2.82*e*−04	7.6031	7.9993	1
4.1.05	House	256 × 256	1.95*e*−03	7.0902	7.9993	1
4.1.06	Tree	256 × 256	−3.07*e*−03	7.5634	7.9993	1
4.1.07	Jelly beans	256 × 256	−1.85*e*−03	6.6098	7.9993	1
4.1.08	Jelly beans	256 × 256	1.44*e*−03	6.8863	7.9993	1
4.2.01	Splash	512 × 512	9.15*e*−04	7.3232	7.9994	1
4.2.02	Girl (tiffany)	512 × 512	−2.28*e*−03	6.6691	7.9993	1
4.2.03	Mandrill (a.k.a. Baboon)	512 × 512	−5.77*e*−04	7.7659	7.9993	1
4.2.04	Girl (lena. or lena)	512 × 512	4.40*e*−04	7.7548	7.9993	1
4.2.05	Airplane (F-16)	512 × 512	1.89*e*−04	6.6879	7.9994	1
4.2.06	Sailboat on lake	512 × 512	5.04*e*−03	7.7675	7.9994	1
4.2.07	Peppers	512 × 512	2.20*e*−03	7.7253	7.9993	1
5.1.09	Moon surface	256 × 256	−1.16*e*−03	6.719	7.9993	1
5.1.10	Aerial	256 × 256	1.71*e*−03	7.322	7.9992	1
5.1.11	Airplane	256 × 256	−3.61*e*−03	6.4658	7.9994	1
5.1.12	Clock	256 × 256	1.95*e*−03	6.7111	7.9993	1
5.1.13	Resolution chart	256 × 256	−1.47*e*−03	2.2863	7.9994	1
5.1.14	Chemical plant	256 × 256	8.36*e*−05	7.3473	7.9993	1
5.2.08	Couple	512 × 512	−5.45*e*−04	7.2187	7.9994	1
5.2.09	Aerial	512 × 512	1.37*e*−03	7.0015	7.9994	1
5.2.10	Stream and bridge	512 × 512	−1.92*e*−03	7.721	7.9992	1
7.1.01	Truck	512 × 512	8.68*e*−04	6.5836	7.9993	1
7.1.02	Airplane	512 × 512	−3.11*e*−05	5.4408	7.9993	1
7.1.03	Tank	512 × 512	−1.54*e*−03	6.4078	7.9994	1
7.1.04	Car and APCs	512 × 512	−1.91*e*−04	6.8064	7.9992	1
7.1.05	Truck and APCs	512 × 512	−1.27*e*−03	7.1124	7.9994	1
7.1.06	Truck and APCs	512 × 512	2.49*e*−03	7.0571	7.9992	1
7.1.07	Tank	512 × 512	2.88*e*−04	6.5399	7.9993	1
7.1.08	APC	512 × 512	−2.56*e*−03	5.9022	7.9993	1
7.1.09	Tank	512 × 512	1.01*e*−03	6.9868	7.9993	1
7.1.10	Car and APCs	512 × 512	−1.11*e*−03	6.6201	7.9994	1
boat.512	Fishing boat	512 × 512	9.47*e*−04	7.2151	7.9992	1
elaine.512	Girl (elaine)	512 × 512	5.99*e*−04	7.5118	7.9992	1
House	House	512 × 512	1.67*e*−03	6.5802	7.9993	1
gray21.512	21 level step wedge	512 × 512	1.54*e*−03	4.5922	7.9992	1
numbers.512	256 level test pattern	512 × 512	−1.56*e*−03	7.7768	7.9994	1
	Azafack	398 × 512	−1.19*e*−03	7.7018	7.9994	1
	Guefack	365 × 486	8.64*e*−04	6.6996	7.9994	1

then «A23C56789ABADEF6167DEAB6789367A9». The encrypted image is shown in Figure 7(d).

(d) In the test number 3, the used external key is changed into «A23C56789ABADEF7167DEAB6 789367AA». Here, the last character 9 in the first external key is changed into A. Figure 7(e) presents a cipher image obtained.

(e) For the last test, the second character 2 in the external key is changed into 3. The used external key becomes «A33C56789ABADEF7167DEAB6789367 A9». Figure 7(f) presents a cipher image obtained.

For a quantitative assessment of the similarity between those images, we present in Table 7 the correlation coefficients between the different cipher images.

Although the difference from one key to the other is a single bit, we note from Table 7 that the values of the correlations' coefficients between the different encrypted images obtained are closed to zero. The highest correlation

Table 5: Correlation coefficients between two vertically, horizontally, and diagonally adjacent pixels in several medical plain images and also in their corresponding encrypted images.

Image name	Correlation coefficients of original images			Correlation coefficients of cipher images		
	Vert. cor	Hor. cor	Diag. cor	Vert. cor	Hor. cor	Diag. cor
ANTAMOEBACOLI	0.77141	0.81532	0.67636	−2.96*e*−03	1.31*e*−03	1.06*e*−05
Article_oeuf_taeniaC2	0.93107	0.9308	0.87861	−1.94*e*−03	−5.14*e*−03	7.55*e*−04
Balantidium coli cyst	0.76953	0.78613	0.66231	1.74*e*−03	−6.15*e*−04	6.42*e*−04
Balantidium coli_trophozoite	0.9259	0.93259	0.90567	2.19*e*−03	−3.48*e*−03	1.56*e*−04
DICROCOELIUM	0.97729	0.98048	0.96751	3.89*e*−04	3.14*e*−03	−5.42*e*−04
Entamoeba coli trophozoite	0.99129	0.99215	0.98588	−1.85*e*−03	−9.26*e*−04	3.17*e*−03
Entamoeba histolytica cyst	0.93725	0.94537	0.90185	2.47*e*−03	−2.75*e*−03	−2.22*e*−03
Entamoeba histolytica-cyst-Gini	0.92637	0.91525	0.85601	1.69*e*−03	−2.96*e*−03	−2.11*e*−04
Entamoeba histolytica trophozoite	0.98471	0.98541	0.97413	2.36*e*−03	2.87*e*−03	9.35*e*−04
Entamoeba histolytica trophozoite_redim	0.98253	0.98307	0.96996	−1.73*e*−03	−2.91*e*−03	−3.19*e*−04
Entamoeba histolytica trophozoite_redim2	0.98622	0.9867	0.97545	2.20*e*−03	1.51*e*−04	−8.70*e*−05
oeuf_ascarisc	0.93155	0.93445	0.88098	2.59*e*−04	−1.48*e*−03	−6.88*e*−04
S-Hematobium egg	0.88554	0.92383	0.8354	−1.64*e*−03	−2.70*e*−03	−1.60*e*−03
S-Mansoni egg	0.86917	0.9053	0.80198	−3.42*e*−03	4.26*e*−04	1.10*e*−03
Tropho_entamoeba_histolytica2	0.99265	0.99139	0.98605	3.29*e*−03	5.36*e*−05	1.42*e*−03
Tropho_iodamoeba_butschlii	0.9966	0.99688	0.99391	−4.73*e*−04	5.99*e*−04	−1.66*e*−03
Angio	0.89905	0.96795	0.90462	−1.78*e*−03	3.27*e*−03	−8.62*e*−04
DisLocElbow	0.96446	0.99666	0.9608	1.40*e*−03	5.22*e*−04	1.54*e*−03
Echo1	0.88904	0.8776	0.8212	1.77*e*−03	−2.03*e*−04	3.88*e*−04
I1_200	0.99393	0.98775	0.98353	7.66*e*−04	8.84*e*−04	1.20*e*−03
Node2	0.96491	0.95401	0.93398	3.21*e*−03	2.26*e*−03	−9.21*e*−04
Ossify	0.99017	0.94411	0.9303	2.44*e*−03	2.14*e*−03	9.73*e*−04
Pelvis	0.94299	0.99754	0.93903	−1.73*e*−03	1.78*e*−03	−3.68*e*−03
Ribs	0.87322	0.88588	0.85227	−2.05*e*−04	−8.69*e*−04	1.70*e*−04
Dirofilaria	0.98814	0.97363	0.96349	−9.60*e*−04	−8.31*e*−04	−4.47*e*−04
Headirm	0.96343	0.95734	0.93986	−2.08*e*−03	6.53*e*−04	1.32*e*−03
Abdomenirm	0.91929	0.91331	0.8508	−1.35*e*−03	−6.20*e*−05	3.75*e*−04
Pelvisirm	0.96602	0.95567	0.93225	−9.36*e*−04	4.49*e*−04	2.93*e*−04
Gastrointestinal_parasites	0.91404	0.94614	0.88418	−3.06*e*−04	−2.40*e*−03	5.55*e*−05
Echo fetus at 12 weeks	0.91943	0.90556	0.85116	7.24*e*−04	−3.15*e*−03	−1.24*e*−03
Ultrasound of fetus of 3 months	0.96781	0.98601	0.95989	−1.85*e*−03	−1.60*e*−03	2.34*e*−03
Echopelv	0.89428	0.90612	0.82813	−9.96*e*−04	−1.11*e*−03	3.76*e*−04
CT-MONO2-8-abdo	0.93917	0.95915	0.91275	−2.57*e*−03	−3.56*e*−03	−2.56*e*−04
OT-MONO2-8-colon	0.96189	0.96556	0.93819	−2.33*e*−03	5.95*e*−04	−4.25*e*−04

Table 6: Correlation coefficients between two vertically, horizontally, and diagonally adjacent pixels in several nonmedical plain images and also in their corresponding encrypted images from database.

Image Name	Description	Correlation coefficients of original images			Correlation coefficients of cipher images		
		Vert cor	Hor cor	Diag cor	Vert cor	Hor cor	Diag cor
4.1.01	Girl	0.96547	0.9737	0.94928	-2.94*e*−04	1.58*e*−03	5.41*e*−04
4.1.02	Couple	0.95615	0.93889	0.90658	2.09*e*−03	-1.85*e*−04	9.21*e*−04
4.1.03	Girl	0.91432	0.97598	0.89425	-4.05*e*−04	2.47*e*−03	4.50*e*−04
4.1.04	Girl	0.98476	0.96995	0.95805	5.88*e*−05	5.11*e*−04	-1.98*e*−03
4.1.05	House	0.95289	0.9781	0.94157	-3.75*e*−04	-1.20*e*−03	2.38*e*−04
4.1.06	Tree	0.9441	0.96695	0.9285	-2.87*e*−03	-3.22*e*−03	5.65*e*−04
4.1.07	Jelly beans	0.98233	0.9787	0.96461	1.58*e*−03	-4.40*e*−03	-2.56*e*−03
4.1.08	Jelly beans	0.97553	0.97258	0.95246	3.56*e*−03	-7.71*e*−04	-6.16*e*−04
4.2.01	Splash	0.9915	0.98399	0.98054	2.25*e*−04	-1.19*e*−03	-8.85*e*−04
4.2.02	Girl (tiffany)	0.94097	0.93826	0.91514	-2.56*e*−03	-8.88*e*−04	1.63*e*−03
4.2.03	Mandrill (a.k.a. Baboon)	0.75486	0.86269	0.72324	1.36*e*−03	-5.84*e*−04	-4.73*e*−04
4.2.04	Girl (lena. or lena)	0.98485	0.97159	0.9635	9.09*e*−05	-2.57*e*−04	-4.52*e*−04
4.2.05	Airplane (F-16)	0.96394	0.96616	0.94106	-7.67*e*−04	2.28*e*−03	-9.57*e*−05
4.2.06	Sailboat on lake	0.97003	0.97368	0.95768	9.21*e*−04	5.35*e*−04	3.48*e*−04

Table 6: Continued.

Image Name	Description	Correlation coefficients of original images			Correlation coefficients of cipher images		
		Vert cor	Hor cor	Diag cor	Vert cor	Hor cor	Diag cor
4.2.07	Peppers	0.97788	0.97567	0.96806	3.40e−03	-1.47e−03	-3.18e−04
5.1.09	Moon surface	0.93093	0.89444	0.88385	3.49e−03	2.34e−03	-1.91e−03
5.1.10	Aerial	0.85731	0.90234	0.80779	3.27e−04	-1.65e−03	1.42e−03
5.1.11	Airplane	0.93722	0.95697	0.91391	6.44e−04	-1.97e−03	-1.84e−03
5.1.12	Clock	0.97373	0.95613	0.93755	-1.33e−03	-1.34e−03	1.06e−03
5.1.13	Resolution chart	0.86678	0.87242	0.75713	1.78e−03	8.89e−04	-2.11e−04
5.1.14	Chemical plant	0.89647	0.94525	0.8672	8.69e−04	-3.28e−03	-2.29e−03
5.2.08	Couple	0.89244	0.93684	0.85528	1.14e−04	9.13e−04	1.12e−03
5.2.09	Aerial	0.85803	0.89863	0.80248	-1.94e−03	-2.45e−04	-6.92e−04
5.2.10	Stream and bridge	0.92548	0.93854	0.89787	1.79e−03	1.26e−03	8.91e−04
7.1.01	Truck	0.91738	0.95927	0.90241	-1.88e−03	4.08e−03	-1.90e−04
7.1.02	Airplane	0.94637	0.9467	0.91936	-1.35e−03	2.23e−03	1.04e−03
7.1.03	Tank	0.92665	0.94073	0.90446	-2.69e−03	-3.46e−03	-2.49e−03
7.1.04	Car and APCs	0.96614	0.97538	0.95392	-1.17e−03	6.70e−04	1.46e−03
7.1.05	Truck and APCs	0.90661	0.93685	0.88725	-4.51e−03	1.12e−03	-8.10e−04
7.1.06	Truck and APCs	0.90045	0.93465	0.88065	-3.92e−03	6.38e−04	3.34e−04
7.1.07	Tank	0.86813	0.8768	0.82074	-3.03e−03	-2.04e−04	-1.32e−03
7.1.08	APC	0.92423	0.95333	0.91346	-1.88e−03	3.75e−03	2.08e−04
7.1.09	Tank	0.92675	0.96239	0.91523	-3.11e−03	1.84e−03	-2.76e−03
7.1.10	Car and APCs	0.94504	0.96181	0.92815	-2.16e−04	-3.24e−03	7.66e−04
Boat.512	Fishing boat	0.96993	0.93638	0.92308	5.46e−04	-3.22e−03	1.83e−04
Elaine.512	Girl (elaine)	0.9697	0.97256	0.96796	-1.19e−03	5.92e−04	-1.12e−03
House	House	0.95059	0.97305	0.93846	4.29e−03	-8.68e−04	-2.47e−03
Gray21.512	21-level step wedge	0.99984	0.99653	0.99636	1.55e−03	-1.49e−03	8.19e−04
Numbers.512	256-level test pattern	0.71603	0.73889	0.64186	1.32e−03	-3.03e−05	1.43e−03
	Azafack	0.95872	0.94002	0.92429	1.94e−04	3.10e−03	-5.24e−04
	Guefack	0.99057	0.99022	0.98472	1.60e−03	5.02e−04	6.64e−04

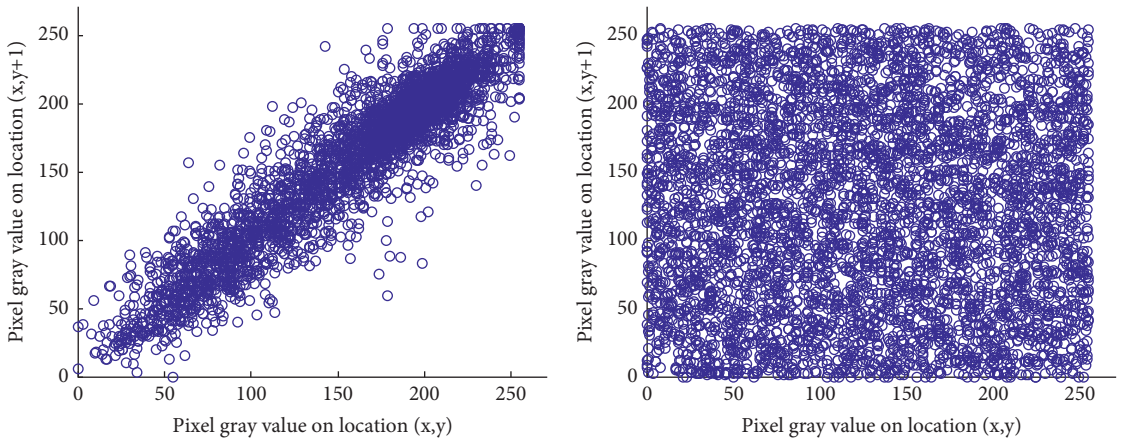


Figure 5: Continued.

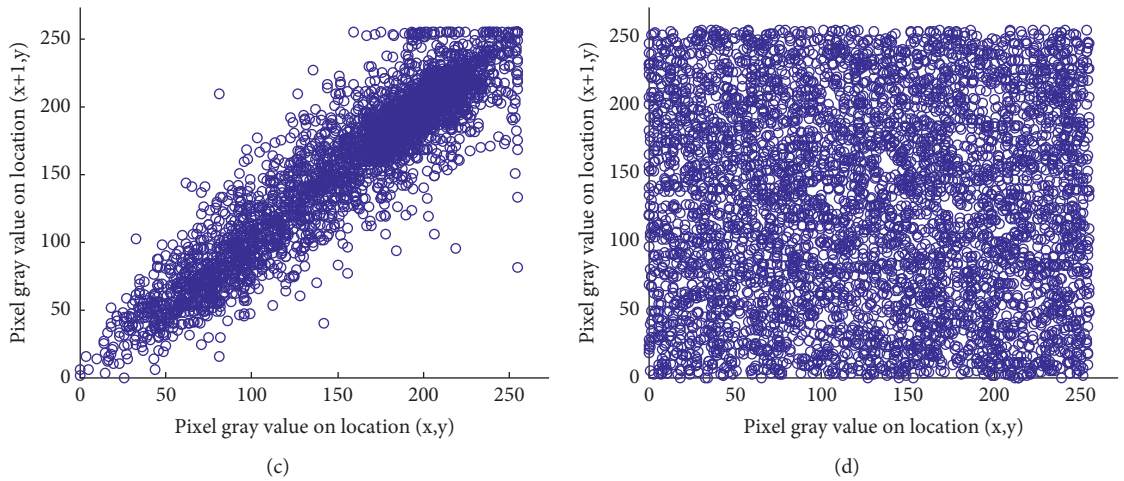


Figure 5: Correlation of adjacent pixels in «article_oeuf_taeniaC2.jpg image. (a) and (c) The distribution of horizontal and vertical adjacent pixels in the plain image. (b) and (d) The distribution of horizontal and vertical adjacent pixels in the corresponding encrypted image.

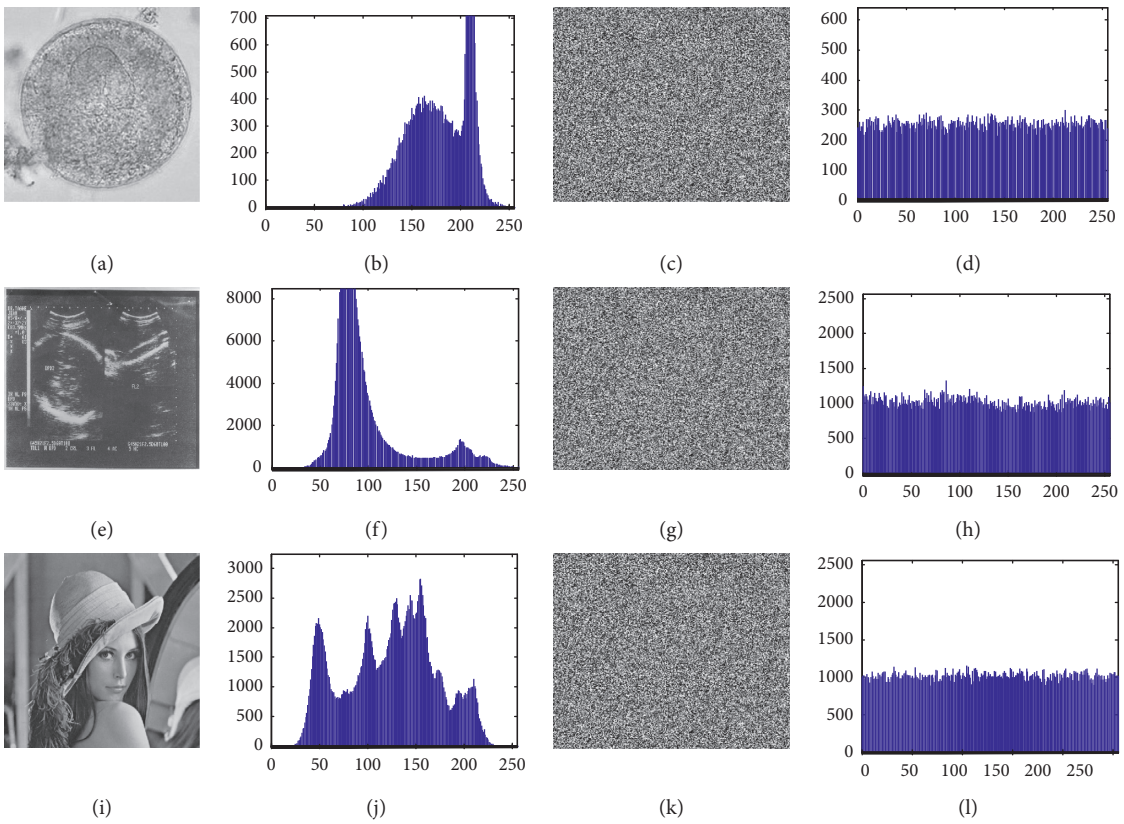


Figure 6: Continued.

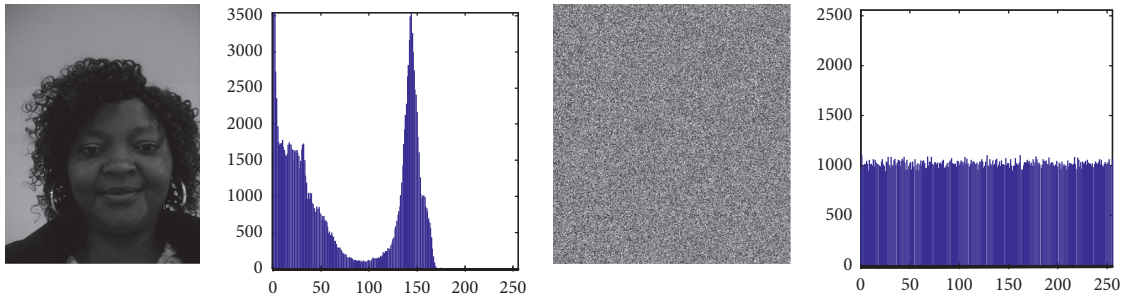


Figure 6: Histogram analysis plain and cipher images using the secret key «A23C56789ABADEF7167DEAB6789367A9»: (a), (b), (c), and (d) A plain image "Balantidium Coli cyst" and its corresponding histogram, cipher image, and cipher image histogram, respectively. (e), (f), (g), and (h) A plain image "echopelv» and its corresponding histogram, cipher image, and cipher image histogram, respectively. (i), (j), (k), and (l) A plain image "Girl (Lena, 4.2.04)» and its corresponding histogram, cipher image, and cipher image histogram, respectively. (m), (n), (o), and (p) A plain image "Guefack» and its corresponding histogram, cipher image, and cipher image histogram, respectively.

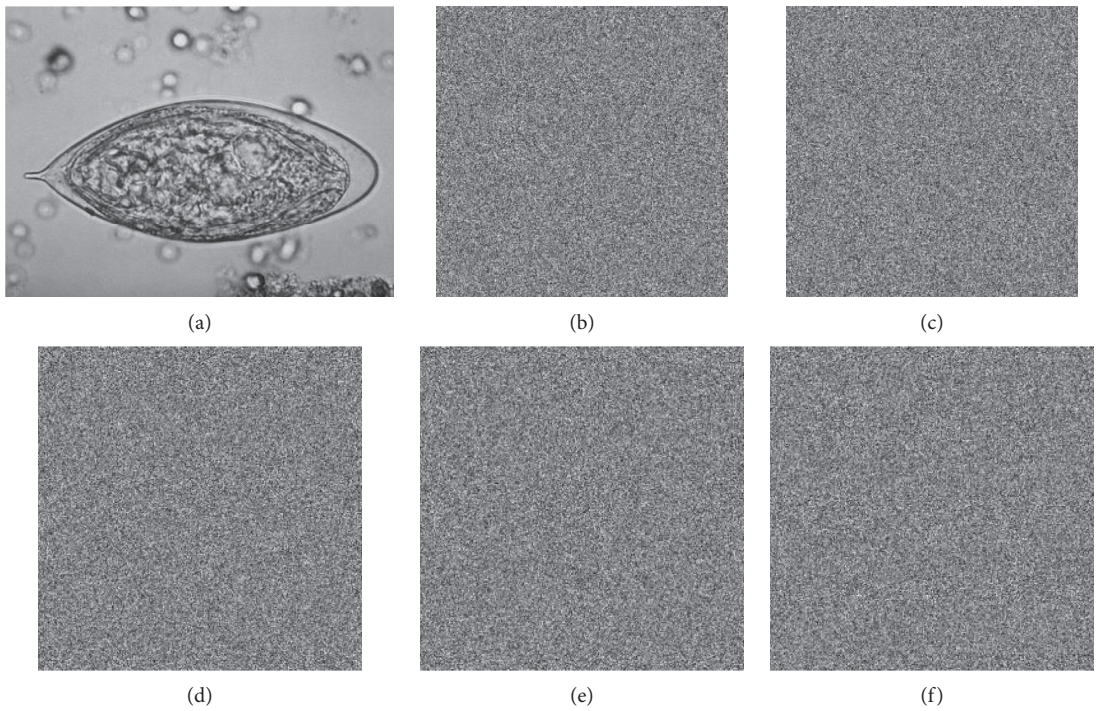


Figure 7: Plain and cipher images. (a) The plain image of «S-Hematobium egg». (b) to (f) The different cipher images obtained with different external keys.

value obtained is 0.0041. This test was performed on several other images and all the results are conclusive. This implies that the encrypted images produced from the proposed cryptosystem are different. This means that the proposed cryptosystem is very sensitive to the encryption key.

3.5.2. Key Sensitivity Test of Decryption Process. In a robust encryption scheme, an insignificant change in the key should not let to the decryption of the cipher image successfully [17]. The key sensitivity test was performed using a slightly different external key to decrypt the encrypted images. Some examples are given below. A "*ultrasound of fetus of 3 months*" image (Figure 8(a)) has been encrypted using the proposed cryptosystem where the external key is "A23C56789ABADEF7167DEAB6789367A9». The encrypted image is shown in Figure 8(b).

(a) Firstly, the encrypted image (Figure 8(b)) is decrypted with a decrypted external key «A23D56789ABADEF7167DEAB6789367A9» which is different to the encryption external key «A23C56789ABADEF7167DEAB6789367A9» by a

Table 7: Correlation coefficients between various cipher images presented in Figure 7

Images	Correlation coefficients
Figures 6(a) and 6(b)	−0.0010
Figures 6(a) and 6(c)	−6.6046*e*−04
Figures 6(a) and 6(d)	−2.6622*e*−04
Figures 6(a) and 6(e)	1.4038*e*−04
Figures 6(a) and 6(f)	0.0014
Figures 6(b) and 6(c)	5.5615*e*−04
Figures 6(b) and 6(d)	0.0031
Figures 6(b) and 6(e)	0.0027
Figures 6(b) and 6(f)	−0.0016
Figures 6(c) and 6(d)	0.0041
Figures 6(c) and 6(e)	−0.0022
Figures 6(c) and 6(f)	8.2013*e*−04
Figures 6(d) and 6(e)	−5.3809*e*−04
Figures 6(d) and 6(f)	8.4005*e*−04
Figures 6(e) and 6(f)	−6.3580*e*−04

single bit. The fourth character C in the encryption external key is changed into D in the decryption external key. The decrypted image is shown in Figure 8(c).

(b) In the second test, the encrypted image (Figure 8(b)) is decrypted by using «A23C56789AB-ADEF6167DEAB6789367A9» as the external key. The sixteenth character **7** in the encryption external key is changed into **6** in the decryption external key. The decrypted image is shown in Figure 8(d).

(c) In this case, the encrypted image (Figure 8(b)) is decrypted using the decrypted external key «A23C56789ABADEF7167DEAB6789367AA». The last character 9 in the encryption external key is changed into A in the decryption external key. Figure 8(e) shows the decrypted image which is not correlated with the original image.

(d) In Figure 8, Figure 8(b) has been decrypted using «A33C56789ABADEF7167DEAB6789367A9» as the decrypted external key. The second character 2 in the encryption external key is changed into 3 in the decryption external key.

Physically, the decrypted images are not similar to a plain image "*ultrasound of fetus of 3 months*," as we can see in Figure 8. The correlation coefficients between the plain image and the decrypted images using a slightly different key have been calculated. They are all closed to zero as we can see in Table 8. Any change on an external secret key affects the angle of the eight used rotations and changes an internal key. Consequently, the resulting substitution-diffusion key is modified. It comes out that, without an exact key, one cannot succeed in the decryption process. This confirms the effectiveness and key sensitivity of the proposed algorithm.

So, the proposed cryptosystem is very sensitive to the encryption and decryption of external keys.

3.6. Attack Analyses

3.6.1. Entropy Information Analysis. It comes out from Tables 3 and 4 that the values of entropy information obtained with our new proposed scheme on medical and nonmedical images are very close to 8. The highest value is 7.9998 and the lowest one is 7.9991 for medical images, while the lowest value is 7.9992 and the highest value is 7.9994 for nonmedical images. These values are very close to eight (ideal value) compared to the entropy of the original images. This indicates that the proposed algorithm has hidden information randomly, and information leakage in the encryption process is negligible. We evaluated the local entropy of many images using TB = 1936 as in [43], and the results are presented in Table 9. We conclude the effectiveness of the algorithm considering the high values of entropy information and the local entropy.

3.6.2. Differential Attacks. Tables 10 and 11 present the values of the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) for medical and nonmedical images.

In all the cases tested, the NPCR values are closed to 99.6% and the UACI is found close to 33.33%. Our new algorithm is very sensitive with respect to a small percentage of pixels' change in the plain image and the rate of influence because one-pixel change in the plain image is very high. According to Tables 10 and 11, the correlation coefficients between the cipher images are negligible. So, a minor pixel change in the plain image has an important effect on the cipher image. The proposed encryption scheme is sensitive to a minor change in the plain image.

We have evaluated MSE and PSNR on all test images; the MSE is zero in all the cases and the PSNR is infinite, as we can see in Tables 10 and 11. The original and the decrypted image are identical in all cases.

4. Discussion

This work proposes a new image encryption algorithm which does not use chaotic functions or mathematical functions which are time-consuming and make the algorithm too complex. It uses an external key of 128 bit size and an internal key. The originality of this method dwells on the combination of external and internal keys and the use of reflection or rotation mapping and the method to apply substitution and zigzag processes. An internal key comes from the decomposition of an image to be encrypted. The method to extract an internal key has been explained in Section 2. This internal key is the first level to ensure the security of the proposed system. To increase the security of this system, we combine the two keys to produce the diffusion-substitution sequences. These two aspects are the first novelty of the work. The second novelty comes from binary image processing. Each binary image is reshuffled within the

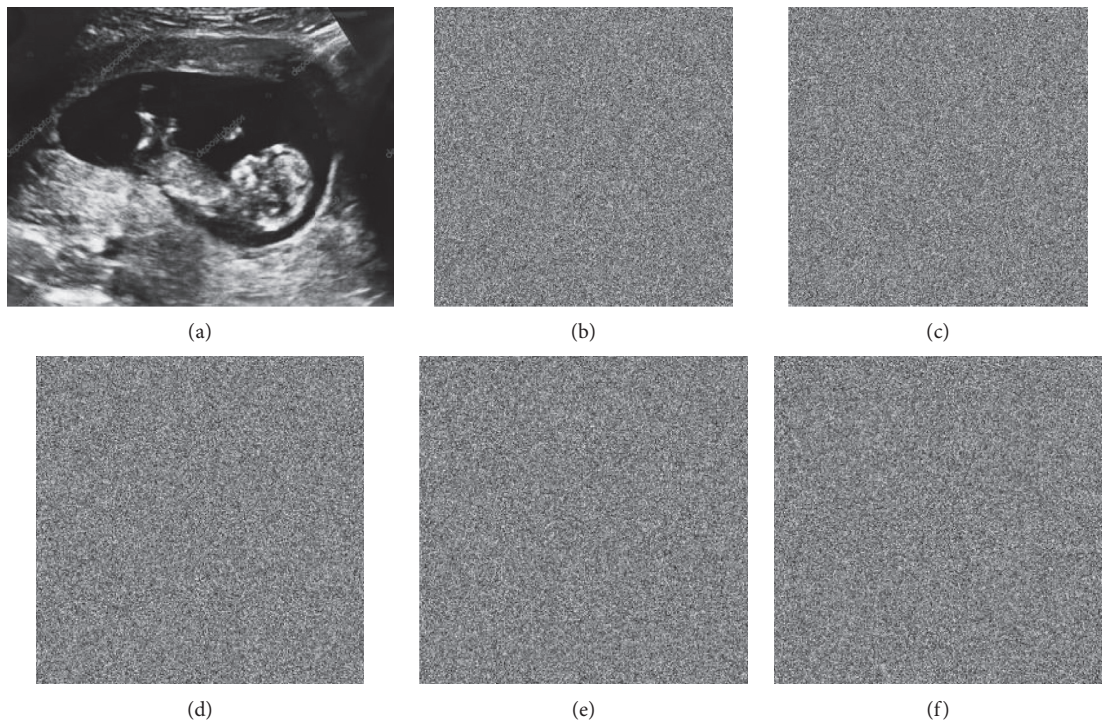


Figure 8: (a and b) The plain image from "ultrasound of fetus of 3 months" image and its corresponding encrypted image. (c)-(f) The decrypted images from the encrypted image of Figure 8(b) using slightly different decryption keys than the key used for encryption.

Table 8: Correlation coefficients between various decrypted images shown in Figure 8.

Images	Correlation coefficients
Figures 7(a) and 7(b)	0.0019
Figures 7(a) and 7(c)	−5.0690*e*−04
Figures 7(a) and 7(d)	9.6489*e*−04
Figures 7(a) and 7(e)	−5.8923*e*−04
Figures 7(a) and 7(f)	1.4796*e*−04
Figures 7(b) and 7(c)	−0.0014
Figures 7(b) and 7(d)	−0.0014
Figures 7(b) and 7(e)	8.0764*e*−04
Figures 7(b) and 7(f)	−0.0018
Figures 7(c) and 7(d)	−0.0016
Figures 7(c) and 7(e)	−1.0391*e*−04
Figures 7(c) and 7(f)	0.0030
Figures 7(d) and 7(e)	0.0038
Figures 7(d) and 7(f)	0.0024
Figures 7(e) and 7(f)	8.9324*e*−04

Table 9: Local entropies for the cipher images.

Images	Balantidium coli cyst (200 × 200)	Echopelv (601 × 711)	Girl (lena, 4.2.04) (512 × 512)	Guefack (365 × 486)
Local entropy information	7.9088	7.9081	7.9091	7.9085

block by the zigzag path. The corresponding reflections and rotations are applied to binary images obtained from the decomposition of the original image. It is also the case during the subblock substitution process. The scan zigzag process is applied not on the pixels of the subblock but on the bits of the pixel. The new pixel block is reconstructed. Consequently, the zigzag operation changes the values of the pixels and acts as a substitution process. In image encryption algorithm, the size of the external key, the size of subblock, the number of the subblock, the number of the rounds on each subblock, the variation of the key from one subblock to another one and from one round to another one, the generator used to generate sequences for the substitution-diffusion process, and the encryption scheme are the factors

Table 10: Values of the number of pixels change rate (NPCR) and unified average changing intensity (UACI) in several medical images.

Image name	Corr (A.B)	NCPR	UACI	MSE	PSNR
ANTAMOEBACOLI	−1.72*e*−03	99.6166	33.4613	0	∞
article_oeuf_taeniaC2	1.26*e*−03	99.6227	33.4476	0	∞
Balantidium coli cyst	−1.43*e*−03	99.6082	33.4647	0	∞
Balantidium coli_trophozoite	3.73*e*−04	99.6044	33.4487	0	∞
DICROCOELIUM	−5.02*e*−04	99.604	33.4573	0	∞
Entamoeba coli trophozoite	7.61*e*−04	99.6201	33.4661	0	∞
Entamoeba histolytica cyst	−1.82*e*−03	99.6105	33.5386	0	∞
Entamoeba histolytica-cyst-Gini	−2.94*e*−03	99.6159	33.5402	0	∞
Entamoeba histolytica trophozoite	6.55*e*−04	99.6124	33.4565	0	∞
Entamoeba histolytica trophozoite_redim	8.47*e*−04	99.6067	33.4592	0	∞
Entamoeba histolytica trophozoite_redim2	2.35*e*−04	99.5987	33.4662	0	∞
Oeuf_ascarisc	7.00*e*−04	99.6132	33.4818	0	∞
S-Hematobium egg	−1.13*e*−03	99.6124	33.44	0	∞
S- Mansoni egg	−1.20*e*−03	99.604	33.5229	0	∞
Tropho_entamoeba_histolytica2	1.82*e*−03	99.6132	33.4328	0	∞
Tropho_iodamoeba_butschlii	1.46*e*−04	99.5926	33.472	0	∞
Angio	5.33*e*−04	99.5983	33.4329	0	∞
DisLocElbow	−1.62*e*−03	99.6235	33.4576	0	∞
Echo1	7.11*e*−04	99.6166	33.4819	0	∞
I1_200	−3.09*e*−03	99.6059	33.5545	0	∞
Node2	9.36*e*−04	99.6075	33.4614	0	∞
Ossify	−7.55*e*−04	99.6227	33.5145	0	∞
Pelvis	−3.33*e*−03	99.6227	33.4752	0	∞
Ribs	3.25*e*−03	99.6136	33.4069	0	∞
Dirofilaria	2.97*e*−04	99.6151	33.4711	0	∞
Headirm	1.65*e*−04	99.5991	33.4599	0	∞
Abdomenirm	−2.53*e*−04	99.6212	33.478	0	∞
Pelvisirm	3.67*e*−03	99.5869	33.3766	0	∞
Gastrointestinal_parasites	−1.53*e*−03	99.6235	33.4498	0	∞
Echo fetus at 12 weeks	3.31*e*−03	99.6189	33.4343	0	∞
Ultrasound of fetus of 3 months	−2.24*e*−03	99.5979	33.5196	0	∞
Echopelv	1.79*e*−05	99.6007	33.4657	0	∞
CT-MONO2-8-abdo	−4.7542*e*−04	99.617	33.4636	0	∞
OT-MONO2-8-colon	−1.6*e*−03	99.6124	33.498	0	∞

that provide to the security, efficiency, and robustness to the method. The encryption method proposed in this work is based on these parameters; when they are not enough, one can easily cryptanalyse the cipher image during transmission. These features and the obtained results make the proposed system resistant to any kind of attack while complicating the task of cryptanalysis. Certainly, our algorithm is not very fast as the others, but we fight against cryptanalysis, and we gain on efficiency and security. Table 12 presents the comparison of the results with those obtained in [28, 33, 37, 38, 44] by presenting the correlation coefficient between vertically, horizontally, and diagonally adjacent pixels of plain and cipher image, entropy, NPCR, and UACIA of "*Lena, Airplane and Baboon*" images.

Our algorithm gives the best vertical correlation coefficient (9.09*e*−05 for "Lena" image, −0.000767 for "Airplane "image, and 0.00136 for "Baboon" image). In terms of horizontal correlation coefficient, the proposed algorithm also gives the best results (0.000257 for "Lena" image and −0.000584 for "Baboon" image). In [44], the horizontal correlation coefficient on "Airplane "image is low (0.0017 compare to ours 0.00228), but we have a high entropy value (7.994 compared to theirs 7.990). For diagonal correlation coefficient obtained in [37] is the lowest (−0.00018) on «Lena» image, but the proposed algorithm also gives the highest entropy value (7.9993 for "Lena" image, 7.9994 for "Airplane" image, and 7.9993 for "Baboon" image). In [44], the nearest value of the entropy 7.9992 has been obtained, but chaotic sequences are used and the algorithm used to obtain these sequences is too complex. In terms of NPCR and UACIA, the values are very close in all cases. According to the results, the proposed method provides better performance than the method based on CNT proposed in [33] in terms of correlation, NPCR, and UACI. Table 13 presents the comparison results in medical images. It comes from Table 13 that the proposed method provides better performance than the method based on CNT proposed in [32]. The single difference is in "OT-MONO2-8-colon" image, where the vertical correlation value (−0.0003) in [32] is higher than ours (−2,33*e*−03). It comes from this table that the proposed method has a performance similar to that achieved by other recently proposed techniques. The advantages of the proposed method are the originality, the simplicity of algorithm, the efficiency, and the lower number of the round. We use five rounds instead of sixteen as in [28].

Table 11: Values of the number of pixels' change rate (NPCR) and unified average changing intensity (UACI) in several nonmedical images.

File name	Description	Size	Corr (A.B)	NCPR	UACI	MSE	PSNR
4.1.01	Girl	256 × 256	−1.35*e*−03	99.5995	33.452	0	∞
4.1.02	Couple	256 × 256	−9.35*e*−04	99.6117	33.4913	0	∞
4.1.03	Girl	256 × 256	−1.38*e*−03	99.6178	33.5003	0	∞
4.1.04	Girl	256 × 256	5.57*e*−04	99.6136	33.4734	0	∞
4.1.05	House	256 × 256	−6.70*e*−05	99.6017	33.481	0	∞
4.1.06	Tree	256 × 256	−3.04*e*−03	99.6166	33.5354	0	∞
4.1.07	Jelly beans	256 × 256	3.16*e*−03	99.614	33.4357	0	∞
4.1.08	Jelly beans	256 × 256	2.05*e*−03	99.604	33.4272	0	∞
4.2.01	Splash	512 × 512	9.57*e*−04	99.612	33.4338	0	∞
4.2.02	Girl (tiffany)	512 × 512	2.78*e*−03	99.6105	33.3975	0	∞
4.2.03	Mandrill (a.k.a. Baboon)	512 × 512	1.94*e*−04	99.6063	33.4353	0	∞
4.2.04	Girl (lena. or lena)	512 × 512	5.32*e*−-04	99.6254	33.4604	0	∞
4.2.05	Airplane (F-16)	512 × 512	1.45*e*−03	99.604	33.4426	0	∞
4.2.06	Sailboat on lake	512 × 512	1.39*e*−04	99.609	33.4513	0	∞
4.2.07	Peppers	512 × 512	−1.55*e*−03	99.6105	33.5116	0	∞
5.1.09	Moon surface	256 × 256	2.42*e*−03	99.6166	33.434	0	∞
5.1.10	Aerial	256 × 256	2.44*e*−03	99.6262	33.4209	0	∞
5.1.11	Airplane	256 × 256	6.09*e*−04	99.6082	33.4821	0	∞
5.1.12	Clock	256 × 256	1.58*e*−03	99.5918	33.4126	0	∞
5.1.13	Resolution chart	256 × 256	−2.49*e*−03	99.6101	33.5335	0	∞
5.1.14	Chemical plant	256 × 256	3.96*e*−03	99.6346	33.343	0	∞
5.2.08	Couple	512 × 512	8.94*e*−04	99.5956	33.4183	0	∞
5.2.09	Aerial	512 × 512	4.95b03	99.6113	33.3598	0	∞
5.2.10	Stream and bridge	512 × 512	−7.71*e*−04	99.6117	33.5107	0	∞
7.1.01	Truck	512 × 512	6.92*e*−04	99.6025	33.4377	0	∞
7.1.02	Airplane	512 × 512	2.96*e*−03	99.6044	33.4076	0	∞
7.1.03	Tank	512 × 512	−6.64*e*−04	99.6159	33.4696	0	∞
7.1.04	Car and APCs	512 × 512	2.12*e*−03	99.6105	33.4304	0	∞
7.1.05	Truck and APCs	512 × 512	7.08*e*−04	99.6254	33.4041	0	∞
7.1.06	Truck and APCs	512 × 512	−4.98*e*−04	99.5941	33.5041	0	∞
7.1.07	Tank	512 × 512	7.41*e*−04	99.6178	33.4429	0	∞
7.1.08	APC	512 × 512	−1.30*e*−05	99.596	33.4592	0	∞
7.1.09	Tank	512 × 512	−6.91*e*−04	99.6185	33.4967	0	∞
7.1.10	Car and APCs	512 × 512	2.65*e*−03	99.5937	33.3973	0	∞
Boat.512	Fishing boat	512 × 512	1.10*e*−03	99.633	33.4121	0	∞
Elaine.512	Girl (elaine)	512 × 512	−3.55*e*−03	99.612	33.475	0	∞
House	House	512 × 512	2.72*e*−04	99.6174	33.4448	0	∞
Gray21.512	21-level step wedge	512 × 512	−4.65*e*−04	99.6159	33.446	0	∞
Numbers.512	256-level test pattern	512 × 512	1.07*e*−03	99.6243	33.4408	0	∞
	Azafack	398 × 512	1.97*e*−04	99.6109	33.4439	0	∞
	Guefack	365 × 486	2.72*e*−03	99.5934	33.3593	0	∞

Table 12: Comparison of results on nonmedical images.

Image	Metric	Cipher image [34]	Cipher image [38]	Cipher image [39]	Cipher image [29]	Cipher image [44]	Cipher image of our algorithm
Lena	Vert. cor	−0.0024	0.003709	0.00085	−0.0016	0.0034	9,09*e*−05
	Hor. cor	0.0076	−0.00084	0.00080	0.0031	0.0026	−0.000257
	Diag. cor	0.003	−0.00018	0.00019	0.0067	0.0019	−0.000452
	Entropy	**7.9992 < e < 7.9994**	7.99748		7.9952	7.9992	7.9993
	NPCR	93,7457	>99.6	99.6553	>96	99,6201	99.6254
	UACI	32.3899	33.4	33.3377	31,79	33,4006	33.4604

Table 12: Continued.

Image	Metric	Cipher image [34]	Cipher image [38]	Cipher image [39]	Cipher image [29]	Cipher image [44]	Cipher image of our algorithm
Airplane (512 × 512)	Vert. cor	−0.0095				−0.0036	−0.000767
	Hor. cor	0.0025				−0.0017	0.00228
	Diag. cor	0.0009				−0.0020	−9.57*e*−05
	Entropy	**7.9992 < e < 7.9994**	7.99925			7.9990	7.9994
	NPCR	93.7482	99.5252			99.6178	99.604
	UACI	32.3293	33.38			33.589	33.4426
Baboon	Vert. cor	−0.0092				−0.0019	0.00136
	Hor. cor	0.0019				−0.0014	−0.000584
	Diag. cor	0.0049				−0.0013	−0.000473
	Entropy	**7.9992 < e < 7.9994**	7.9993			7.9991	7.9993
	NPCR	93.7483	99.9935			99.6109	99.6063
	UACI	31.7602	33.69			33.4757	33.4353

Table 13: Comparison of results on medical images.

Images	References	Vert. cor	Hor. cor	Diag. cor	Entropy	NPCR	UACI
CT-MONO2-8-abdo	[32]	−0.0056	0.0132	−0.0006	7.9856 < e < 7.9992	99.6077	33.4501
	Proposed scheme	−2.57*e*−03	−3.56*e*−03	−2.56*e*−04	7.9992	99.617	33.4636
OT-MONO2-8-colon	[32]	−0.0003	0.0012	−0.0087	7.9856 < e < 7.9992	99.6082	33.462
	Proposed scheme	−2.33*e*−03	5.95*e*−04	−4.25*e*−04	7.9993	99.6124	33.498

5. Conclusion

In this work, a new image encryption algorithm has been proposed in order to secure images during transmission. The cryptosystem uses affine transformations (reflections and rotations), an external secret key of 128 bits long, and an internal secret key coming from the decomposition of the plain image, zigzag process, and substitution-diffusion processes. The particularity of this algorithm is the method to extract the internal secret keys, the use of reflection and rotation mappings on the binary images obtained from the decomposition of the plain image to be encrypted, and the zigzag process not on the gray-scale image but on the binary image and binary block, and finally the method used to combine external and internal keys to generate substitution-diffusion sequences without complex mathematical functions or complex chaotic generators. The size of an internal secret key depends on the size of the original image. The substitution process is also taken in two steps. We have evaluated the proposed algorithm on statistical analysis and key sensitivity analysis. The new proposed system is efficient and robust. The main features of the encryption scheme are its simplicity, its efficiency, and a high security order. Our method also has better confusion, diffusion, and security compared to recent methods in the literature. The combination of external and internal secret keys, the changing of the substitution-diffusion key for one subblock to another, makes the method to be robust against brute-force attacks. The newly proposed method is expected to be useful for real-time encryption and transmission of images in many domains such as telemedicine.

Data Availability

The data used to support the findings of the study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Prasad and K. L. Sudha, "Chaos Image Encryption Using Pixel Shuffling," *Computer Science & Information Technology (CS & IT)*, no. 2, pp. 169–179, 2011.

[2] N. Anane, A. Mohamed, B. Hamid, I. Mohamed, and K. Messaoudi, "Rsa based encryption decryption of medical images," in *Proceedings of the 7th International Multi-Conference on Systems Signals and Devices (SSD)*, pp. 1–4, IEEE, Amman Jordan, June 2010.

[3] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer Science & Business Media, Berlin, Germany, 2012.

[4] W. C. Barker and E. B. Barker, *Recommendation for the Triple Data Encryption Algorithm (Tdea) Block Cipher*, National Institute of Standards & Technology, Gaithersburg, MD, USA, 2012.

[5] N. Baran, "News and views: RSA algorithm in the public domain; Woz joins the inventors hall of fame; entangled photons mean faster, smaller ICs; behemoth mothballed; advanced encryption standard selected; SGI releases sdk as open source; WSDL spec released," *Dr. Dobbâs J. Software Tools*, vol. 25, no. 12, p. 18, 2000.

[6] A. Das and A. Adhikari, "An efficient multi-use multi-secret sharing scheme based on hash function," *Applied Mathematics Letters*, vol. 23, no. 9, pp. 993–996, 2010.

[7] S. Mazloom and A. M. E. Moghadam, "Color image encryption based on coupled nonlinear chaotic map, Chaos," *Solitons Fractals*, vol. 42, no. 3, pp. 1745–1754, 2009.

[8] N. Nithin, A. M. Bongale, and G. P. Hegde, "Image encryption based on feal algorithm," *International Journal of Advances in Computer Science and Technology*, vol. 2, no. 3, pp. 14–20, 2013.

[9] M. Ahmad and O. Farooq, "Chaos based PN sequence generator for cryptographic applications," in *Proceedings of the 2011 International Conference on Multimedia, Signal Processing and Communication Technologies*, pp. 83–86, IEEE, December 2011, Aligarh, India.

[10] X. Li, C. Li, and I. K. Lee, "Chaotic image encryption using pseudo-random masks and pixel mapping," *Signal Processing*, vol. 125, pp. 48–63, 2016.

[11] M. Ahmad, M. Z. Alam, Z. Umayya, S. Khan, and F. Ahmad, "An image encryption approach using particle swarm optimization and chaotic map," *International Journal of Information Technology*, vol. 10, no. 3, pp. 247–255, 2018.

[12] Y. Niu, N. Ying, Z. Xuncai, and H. Feng, "Image encryption algorithm based on hyperchaotic maps and nucleotide sequences database," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 4079793, 9 pages, 2017.

[13] T. Adélaïde, H. Fotsin, and J. Kengne, "Image encryption algorithm based on dynamic dna coding operations and 3d chaotic systems," *Multimedia Tools and Applications*, vol. 80, pp. 1–31, 2021.

[14] A. Telem, D. Tchiotsop, K. Thomas, F. Hilaire, and D. Wolf, "A robust chaotic and fast walsh transform encryption for gray scale biomedical image transmission," *Signal & Image Processing: International Journal*, vol. 6, no. 3, pp. 81–102, 2015.

[15] L. Liu and S. A. Miao, "A new image encryption algorithm based on logistic chaotic map with varying parameter," *Springer Plus*, vol. 5, p. 289, 2016.

[16] Q. Liu, P.-Y. Li, M.-C. Zhang, Y.-X. Sui, and H.-J. Yang, "A novel image encryption algorithm based on chaos maps with Markov properties," *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, pp. 506–515, 2015.

[17] N. K. T. Adelaïde, M. S. Colince, K. Godpromesse, and B. F. Hilaire, "A simple and robust gray image encryption scheme using chaotic logistic map and artificial neural network," *Advances in Multimedia*, vol. 2014, p. 13, Article ID 602921, 2014.

[18] M. Kumari and S. Gupta, "Novel image encryption scheme based on intertwining chaotic maps and RC4 stream cipher," *3D Research*, vol. 9, p. 10, 2018.

[19] W. K. Lee, R. C. W. Phan, W. S. Yap, and B. M. Goi, "SPRING: a novel parallel chaos-based image encryption scheme," *Nonlinear Dynamics*, vol. 92, p. 575, 2018.

[20] H. Yuan, Y. Liu, T. Lin, T. Hu, and L.-H. Gong, "A new parallel image cryptosystem based on 5D hyper-chaotic system," *Signal Processing: Image Communication*, vol. 52, pp. 87–96, 2017.

[21] Y. Zhang, "The image encryption algorithm based on chaos and DNA computing," *Multimedia Tools and Applications*, vol. 77, 2018.

[22] A. Daneshgar and K. Behrooz, "A self synchronised chaotic image encryption scheme," *Signal Processing: Image Communication*, vol. 36, pp. 106–114, 2015.

[23] X. Chai, Z. Gan, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequance operations," *Signal Processing: Image Communication*, vol. 52, pp. 6–19, 2017.

[24] M. Li, Y. Guo, H. Jie, and Y. Lia, "Cryptanalyse of a chaotic image encryption scheme based on permutation-diffusion structure," *Signal Processing: Image Communication*, vol. 62, pp. 164–172, 2018.

[25] X. Wang and G. He, "Cryptanalysis on a novel image encryption method based on total shuffling scheme," *Optics Communications*, vol. 284, no. 24, pp. 5404–5407, 2011.

[26] X. Wang, D. Luan, and X. Bao, "Cryptanalysis of an image encryption algorithm using Chebyshev generator," *Digital Signal Processing: A Review Journal*, vol. 25, no. 1, pp. 244–247, 2014.

[27] R. Rhouma, E. Solak, and S. Belghith, "Cryptanalysis of a new substitution–diffusion based image cipher," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 7, pp. 1887–1899, 2010.

[28] N. K. Pareek, V. Patidar, and K. K. Sud, "Diffusion-substitution based gray image encryption scheme," *Digital Signal Processing*, vol. 23, no. 3, pp. 894–901, 2013.

[29] A. Jolfaei, X. W. Wu, and V. Muthukkumarasamy, "Comments on the security of « Diffusion-substitution based gray image encryption » scheme," *Digital Signal Processing*, vol. 32, pp. 34–36, 2014.

[30] A. Houas, Z. Mokhtari, K. E. Melkemi, and A. Boussaad, "A novel binary image encryption algorithm based on diffuse representation," *Engineering Science and Technology, an International Journal*, vol. 19, pp. 1887–1894, 2016.

[31] Z. Mokhtari and K. Melkemi, "A new watermarking algorithm based on entropy concept," *Acta Applicandae Mathematica*, vol. 116, no. 1, pp. 65–69, 2011.

[32] J. B. Lima, F. Madeiro, and F. J. R. Sales, "Encryption of medical images based on cosine number transform," *Signal Processing: Image Communication*, vol. 35, pp. 1–8, 2015.

[33] J. B. Lima, E. S. D. Silva, and R. M. Campello de Souza, "Cosine transform over fields of characteristic2: fast computation and application to image encryption," *Signal Processing: Image Communication*, vol. 54, pp. 130–139, 2017.

[34] M. H. Annaby, M. A. Rushdi, and E. A. Nehary, "Image encryption via discrete fractional fourier-type transform genereted by random matrices," *Signal Processing: Image Communication*, vol. 49, pp. 25–46, 2016.

[35] S. Lian, J. Sun, and Z. Wang, "A block cipher based on a suitable use of chaotic standard," *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 117–129, 2005.

[36] K. W. W. Wong, B. S. HungKwok, and W. ShingLaw, "A fast image encryption scheme based on chaotic standard map," *Physics Letters A*, vol. 372, no. 15, pp. 2645–2652, 2006.

[37] G. Zhang and Q. Liu, "A novel image encryption method based on total shuffling scheme," *Optics Communications*, vol. 284, pp. 2775–2780, 2011.

[38] Z. Eslami and A. Bakhshandeh, "An improvement over an image encryption method based on total shuffling," *Optics Communications*, vol. 286, pp. 51–55, 2013.

[39] H. Zhu, C. Zhao, X. Zhang, and L. Yang, "An image encryption scheme using generalized Arnold map and affine cipher," *Optik*, vol. 125, no. 22, pp. 6672–6677, 2014.

[40] J. Ahmad and S. O. Hwang, "A secure image encryption scheme based on chaotic maps and affine transformation," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13951–13976, 2016.

[41] D. Shah, T. Shah, and S. S. Jamal, "A novel efficient image encryption algorithm based on affine transformation combine

with linear fractional transformation," *Multidimensional Systems and Signal Processing*, vol. 31, pp. 1–21, 2019.

[42] X. L. Chai, Z. H. Gan, Y. Lu, M. H. Zhang, and Y. R. Chen, "A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system," *Chinese Physics B*, vol. 25, no. 10, Article ID 100503, 2016.

[43] S. Wang, C. Wang, and C. Xu, "An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm," *Optics and Lasers in Engineering*, vol. 128, Article ID 105995, 2020.

[44] J. S. Fouda, J. Y. Effa, S. Sabat, and M. Ali, "A fast chaotic block cipher for image encryption," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 3, pp. 578–588, 2014.

[45] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption. cyber journals: multidisciplinary journals in science and technology," *Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[46] Cornell University, "Vision and Image Analysis Group," 2019, http://www.via.cornell.edu/databases.

[47] barre.com, "Medical Image Samples," 2014, http://www.barre.nom.fr/medical/samples.

[48] A. Weber, "The USC-SIPI Image database," 1977, http://sipi.usc.edu/database.

Hindawi
Complexity
Volume 2021, Article ID 1898998, 19 pages
https://doi.org/10.1155/2021/1898998

Research Article

A Secure and Efficient Image Transmission Scheme Based on Two Chaotic Maps

Wei Feng, Jing Zhang, and Zhentao Qin

School of Mathematics and Computer Science, Panzhihua University, Panzhihua 617000, China

Correspondence should be addressed to Jing Zhang; zjpzh@tom.com

Received 6 July 2021; Accepted 2 November 2021; Published 25 November 2021

Academic Editor: Ahmed A. Abd El-Latif

The application of multimedia sensors is widespread, and people need to transmit images more securely and efficiently. In this paper, an image transmission scheme based on two chaotic maps is proposed. The proposed scheme consists of two parts, secure image transmission between sensor nodes and sink nodes (SIT-SS) and secure image transmission between sensor nodes and receivers (SIT-SR). For resource-constrained environments, SIT-SS utilizes Tent-Logistic Map (TLM) to generate chaotic sequences and adopts TLM-Driven permutation and transformation to confuse image pixels. Then the cipher image is obtained through TLM-Driven two-dimensional compressed sensing. Compared with existing schemes, the secret key design of SIT-SS is more reasonable and requires fewer hardware resources. When sampling ratio is greater than 0.6, its image reconstruction quality has obvious advantages. For environments with huge security threats, SIT-SR adopts dynamic permutation and confusion based on discrete logarithms to confuse the image and exploits dynamic diffusion based on discrete logarithms to generate final cipher image. Similarly, compared with some existing schemes, the design of SIT-SR is more practical, and the statistical characteristics of the cipher image are better. Finally, extensive simulation tests confirm the superiority of the proposed scheme.

1. Introduction

Nowadays, the application of multimedia sensors is increasingly widespread in many fields, such as medicine, transportation, industry, education, and military. In these application scenarios, flexibly deployed sensors need to transmit massive images, such as medical and military images [1, 2]. Since it involves privacy protection, commercial and military security, etc., efficient and secure protection needs to be provided for these images. However, image data has several significant characteristics that are different from text data, such as large volume and strong pixel correlation [3]. And the hardware resources of sensors are limited. Therefore, traditional encryption schemes such as Advanced Encryption Standard (AES) are generally not suitable for heterogeneous application environments [4–7]. In order to continuously improve the efficiency and security of image transmission, researchers have been committed to designing new schemes based on emerging techniques and methods [3–28]. Among these new schemes, the ones based on compressed sensing (CS) and chaotic systems are favored by more and more researchers [11–13, 16–28].

CS [29, 30] is a breakthrough signal acquiring paradigm, which can effectively capture and recover a signal with fewer nonadaptive samples. Once introduced, CS is quickly applied to image related information security applications [4–7, 10, 11, 31–37]. In the past decade, researchers have gradually introduced CS into information security applications in resource-constrained environments. In [4], a scheme called Diffie-Hellman-Hash-Compression was proposed. This scheme uses Semitensor Product (STP) CS to encrypt images of different dimensions and adopts hash algorithm and permutation operations to ensure secure image transmission. Taking into account the high privacy sensitivity and redundancy of medical images, Wang et al. [5] constructed a CS based medical image encryption scheme. This scheme carries out image encryption between sensor nodes by using a measurement matrix as the secret key and can realize image compression, privacy protection, and data aggregation simultaneously. In order to overcome

the resource constraints of sensor nodes and ensure the security of data transmission, an image encryption system was exploited [6]. While enhancing the security of transmission process by integrating the quantization and diffusion operations, the system uses a new CS model and parallel reconstruction algorithm to shorten the encryption/decryption time. In [7], a flexible and secure data encryption system based on CS was proposed. The plain image is first sparsely represented through discrete wavelet transform and then permutated by Arnold scrambling. Finally, after CS and logistic chaotic permutation, the cipher image is obtained. Utilizing structurally random matrices, Unde et al. [10] presented an efficient scheme based on CS. In their scheme, artificial noise is injected into quantized CS measurements, thereby enhancing the ability to resist Chosen-Plaintext Attacks (CPAs).

Chaotic systems have several characteristics that are very suitable for designing cryptosystems [1, 2]. Consequently, more and more researchers leverage chaotic systems to design various image encryption schemes. In [16], an image encryption scheme using memristive chaotic system was provided. This scheme uses Secure Hash Algorithm (SHA) to generate the secret key and calculate the initial value of the chaotic system. And it also introduces a dynamic Deoxyribonucleic Acid (DNA) encoding method to generate two regular DNA matrices for encoding images. In order to protect medical images, Moafimadani et al. [23] presented an image encryption scheme based on a chaotic system, which uses a fast permutation operation to scramble the plain image and utilizes an adaptive diffusion operation to generate the cipher image. In [24], a chaotic image encryption scheme using a new symmetric key generation system was proposed. This scheme exploits block-level permutation and improved zigzag transformation to achieve the confusion effect and adopts pixel shuffling to complete the pixel diffusion operation. With the goal of improving the security and efficiency of image encryption, Zhu et al. [25] proposed an efficient and simple S-box generation method using a new compound chaotic system and then introduced a new image encryption scheme based on double S-boxes. Based on dynamic DNA encoding and two chaotic systems, Zhou et al. [26] proposed an image encryption scheme with a two-round permutation-diffusion structure. This scheme exploits a two-dimensional (2D) rectangular transformation to complete the permutation operation, and before the diffusion operation, the hamming distances of DNA matrices are used to update the initial values of the chaotic systems.

As can be seen from abovementioned works, in terms of designing image encryption schemes, reducing resource consumption and achieving higher security are key motivations. Although these schemes have advantages in some aspects, they all have room for further improvements. For example, the scheme proposed in [4] adopts SHA to resist CPAs. However, the implementation of SHA demands considerable hardware resources and would hinder the applicability of this scheme in resource-constrained environments. In addition, some encryption schemes adopt one-time pad secret key. When a large number of images need to be encrypted, such design is not practical. Therefore, while further improving the efficiency and security of image encryption, to overcome the shortcomings of these schemes, an image transmission scheme based on two chaotic maps, 2D-CS, dynamic perturbation, and discrete logarithms (ITS-CDD) is proposed. The proposed scheme consists of two parts, secure image transmission between sensor nodes and sink nodes (SIT-SS) and secure image transmission between sensor nodes and receivers (SIT-SR). Compared with some existing schemes, ITS-CDD has contributions summarized as follows:

(1) SIT-SS is designed for resource-constrained environment, whereas SIT-SR is designed for environments with huge security threats. Therefore, the applicability and practicability of ITS-CDD are higher.

(2) Dynamic perturbation parameters (DPPs) derived from system times and last encryption times are designed. So, ITS-CDD not only guarantees the diversity of equivalent key streams, but also does not rely on external algorithms.

(3) The secret key design of SIT-SS is more practical and requires fewer hardware resources.

(4) 2D-CS based on lightweight chaotic map can reduce resource overhead.

(5) Discrete logarithms under finite multiplicative group Z_{257}^{*} are introduced to ensure higher security.

The remainder of this paper is organized as follows. 2D-CS, discrete logarithms, and two chaotic systems are introduced in Section 2. ITS-CDD is described in Section 3. Simulation tests and theoretical analyses are carried out in Section 4. Finally, conclusions are drawn in Section 5.

2. Fundamental Knowledge

In SIT-SS, 2D-CS is introduced to realize image data compression and encryption. Discrete logarithms are used to enhance the security of SIT-SR. Two chaotic systems called Tent-Logistic Map (TLM) [38] and 2D Logistic-Sine-Coupling Map (2D-LSCM) [13] are adopted to generate the chaotic sequences.

2.1. 2D-CS. In terms of computational complexity and storage space, 2D-CS has obvious advantages over traditional CS [39, 40]. Assuming that $\mathbf{A}$ and $\mathbf{B}$ are random matrices, they both have the size of $M \times N$ ($M \ll N$). Then, one can obtain the 2D measurements $\mathbf{Y} \in R^{M \times M}$ of an image $\mathbf{X} \in R^{N \times N}$. Specifically,

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{B}^{T}, \tag{1}$$

where $\mathbf{A}$ and $\mathbf{B}$ operate on the rows and columns of $\mathbf{X}$, respectively.

When decoding, one can regularize the image signal recovery by using signal prior information in the form of penalty:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} f(\mathbf{X}) = \frac{1}{2}\left\|\mathbf{Y} - \mathbf{A}\mathbf{X}\mathbf{B}^T\right\|_F^2 + \lambda J(\mathbf{X}), \quad (2)$$

where λ is the regularization parameter, $J(\mathbf{X})$ is a cost function which is used to handle the ill-posed problem, and $(1/2)\|\mathbf{Y} - \mathbf{A}\mathbf{X}\mathbf{B}^T\|_F^2$ is the l_2 data-fidelity term. Moreover, researchers have proposed many 2D-CS reconstruction algorithms to solve the optimization problem mentioned above. In this paper, 2D projected gradient with embedding decryption (2DPG-ED) [12] algorithm is adopted.

2.2. Discrete Logarithms. Discrete logarithm calculation is a complex nonlinear calculation. In the encryption process, the use of discrete logarithms can improve its nonlinearity [14]. For the prime 257 and its corresponding finite multiplicative group Z_{257}^*, one can define the discrete logarithms as follows: if $d \in Z_{257}^*$ satisfies $n \equiv g^d \bmod p$, then d is said to be the discrete logarithm of $n \in Z_{257}^*$. Since Z_{257}^* has 128 generators, we can use them to enhance the diversity of equivalent key streams. To avoid complex discrete logarithm calculation, we calculate the discrete logarithm values under different generators in advance and save them to the 2D matrix **DV M** with the size of 128×256. Consequently, in ITS-CDD, discrete logarithm values can be obtained by directly accessing **DV M**. If one wants to calculate the discrete logarithm value of 107 under the generator 3, namely, calculating $(\log_3 107) \bmod 257$, one can access $\mathbf{DVM}_{1,107}$ to obtain the discrete logarithm value 31. Table 1 shows the discrete logarithm values of 101 to 107 under the first eight generators.

2.3. TLM and 2D-LSCM. To save hardware resources, TLM is adopted in SIT-SS, which is easy to implement and has good chaotic performance. TLM can be defined as

$$x_i = \begin{cases} r_1 r_2 x_{i-1}(1 - r_2 x_{i-1}), & x_{i-1} < 0.5, \\ r_1 r_2 (1 - x_{i-1})(1 - r_2(1 - x_{i-1})), & x_{i-1} \geq 0.5, \end{cases} \quad (3)$$

where x_i is generated by the i-th iteration, x_{i-1} is the input of the i-th iteration, $x_0 \in (0, 1)$ is the initial state, and $r_1 \in [3.57, 4], r_2 \in (1, 2]$ are the control parameters. Figure 1 shows the 2D bifurcation diagram and Lyapunov Exponents (LE) diagrams of TLM.

Compared with TLM, 2D-LSCM has better chaotic performance, but its structure is more complex, so it is more suitable for environments with more hardware resources. 2D-LSCM can be defined as

$$\begin{cases} x_i = \sin\left(\pi\left(4\gamma x_{i-1}(1 - x_{i-1}) + (1 - \gamma)\sin\left(\pi y_{i-1}\right)\right)\right), \\ y_i = \sin\left(\pi\left(4\gamma y_{i-1}(1 - y_{i-1}) + (1 - \gamma)\sin\left(\pi x_{i-1}\right)\right)\right), \end{cases} \quad (4)$$

where (x_i, y_i) is the system state generated by the i-th iteration, (x_{i-1}, y_{i-1}) is the input of the i-th iteration, (x_0, y_0) is the initial state, and γ is the control parameter. The value ranges of all these parameters are $[0, 1]$. Figure 2 shows the 2D bifurcation diagram and LE diagram of 2D-LSCM.

3. Proposed Image Transmission Scheme

Different from some existing schemes, ITS-CDD consists of two parts, secure image transmission between sensor nodes and sink nodes (SIT-SS) and secure image transmission between sensor nodes and receivers (SIT-SR). Figure 3 shows the secure image transmission between sensor nodes and sink nodes.

Compared with the existing schemes, SIT-SS has two main innovations. One is introducing TLM to save the hardware resources of sensors, and the other is introducing DPPs to enhance the ability to resist CPAs. Figure 4 shows the secure image transmission between sink nodes and receivers.

Considering that sink nodes have more resources, there are huge security threats in the process of transmitting images to receivers through the media cloud. We have adopted some measures to improve the security of image transmission, such as the adoption of 2D-LSCM with better chaotic performance and the introduction of discrete logarithms.

3.1. Transmission between Sensor Nodes and Sink Nodes. To save space, in this subsection, we mainly introduce the improvements to 2DCS-ETC [12].

3.1.1. DPP Generation. According to previous cryptanalysis works, the main reason why some schemes cannot resist CPAs is that equivalent key streams only depend on the secret key [41–47]. Therefore, some researchers use the hash value of the plain image to ensure the diversity of equivalent key streams. However, the implementation of hash algorithm is not suitable for sensor nodes with limited resources. Considering that system times and last encryption times are constantly changing and would be affected by many factors, they are used to generate DPPs. The specific generation process of DPPs is as follows:

(i) Step 1: obtain the system time T_s in milliseconds.

(ii) Step 2: get the time T_e spent in the last encryption process in milliseconds. If it is the first time to encrypt, set T_e to an initial value T_i.

(iii) Step 3: one DPP is obtained by $\beta = (T_s + T_e) \bmod 256$.

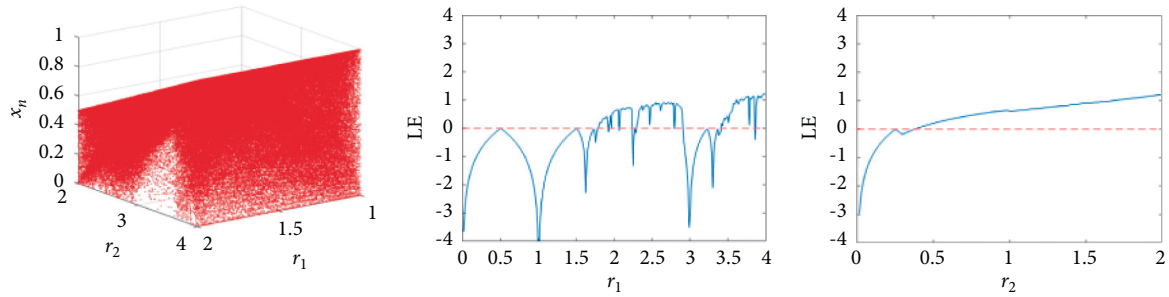
(iv) Step 4: repeat Step 1 through Step 3 until 32 DPPs are obtained, namely, $\beta_1, \beta_2, \ldots, \beta_{32}$.

In this way, we can obtain a set of DPPs. Like the hash value, DPPs can ensure that the equivalent key streams used when encrypting different images are different, thereby effectively resisting CPAs. More importantly, no complicated calculations are required to obtain DPPs, and even if the same plain image is encrypted, different equivalent key streams would be generated.

3.1.2. TLM-Driven Global Permutation. Obviously, confusion is the requirement that must be considered when designing modern cryptosystems. Confusion means that each

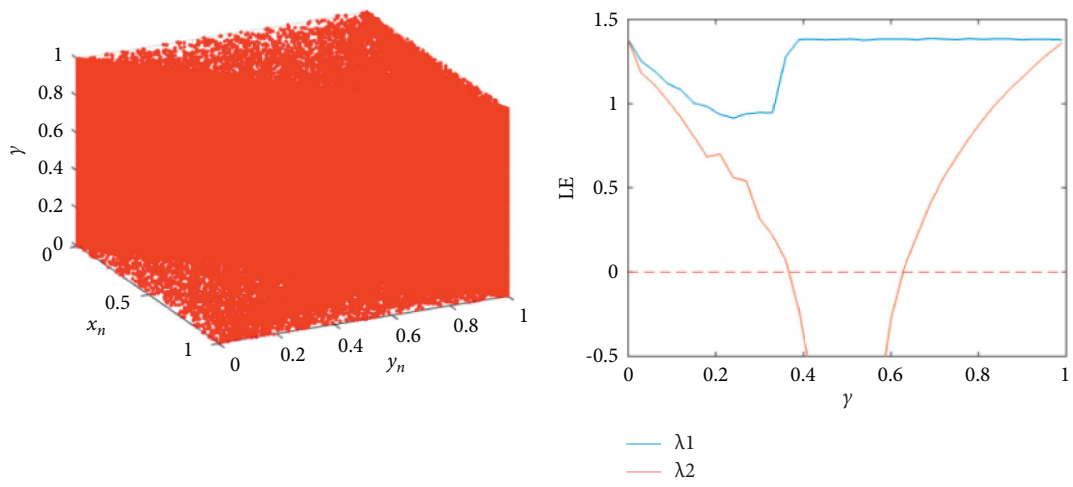
Table 1: Discrete logarithm values of 101 to 107 under the first eight generators.

Row index of **DV M**	Corresponding generator g	n (column index of **DV M**)						
		101 (101)	102 (102)	103 (103)	104 (104)	105 (105)	106 (106)	107 (107)
1	3	75	169	201	250	141	137	31
2	5	141	31	255	214	91	63	89
3	6	59	249	25	26	29	217	79
4	7	31	5	165	18	89	101	163
5	10	125	111	79	246	235	143	137
6	12	43	73	105	58	173	41	127
7	14	143	213	117	50	105	53	83
8	19	103	157	61	2	81	253	203



(a) (b) (c)

Figure 1: 2D bifurcation diagram and LE diagrams of TLM: (a) 2D bifurcation diagram; (b) LE diagram versus parameter r_1; (c) LE diagram versus parameter r_2.



(a) (b)

Figure 2: 2D bifurcation diagram and LE diagram of 2D-LSCM: (a) 2D bifurcation diagram; (b) LE diagram versus parameter γ.

bit of the secret key should affect as many cipher image bits as possible [48]. Permutation operations are commonly used to achieve confusion, but permutation-only image encryption schemes have been proven to be insecure [49]. Therefore, SIT-SS introduces DPPs in the permutation process. This makes the permutation process not only dependent on the secret key, but also dependent on the DPPs that will inevitably change every time the plain image is encrypted. Compared with 2DCS-ETC using the random permutation matrix to complete the permutation and treat it as secret key, we use TLM and DPPs to complete the permutation. This can not only reduce the resource overhead of sensor nodes, but also

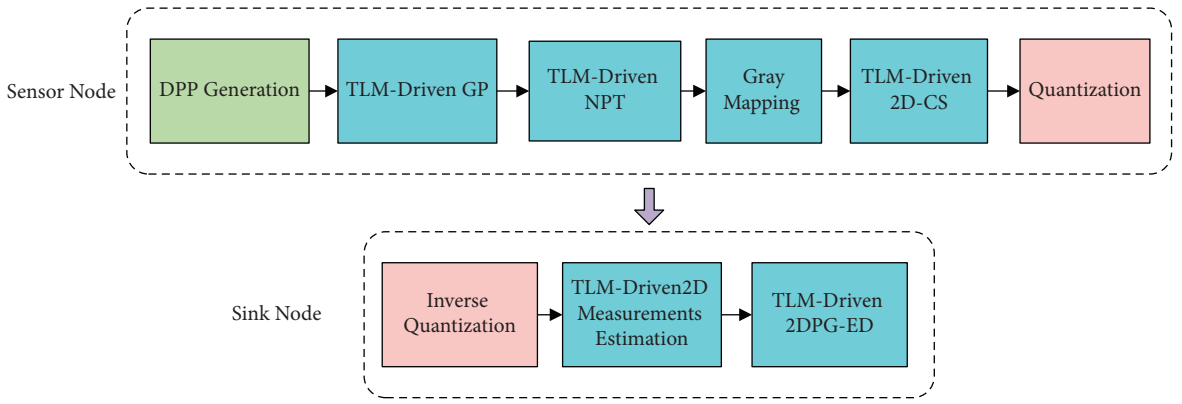


Figure 3: Secure image transmission between sensor nodes and sink nodes.

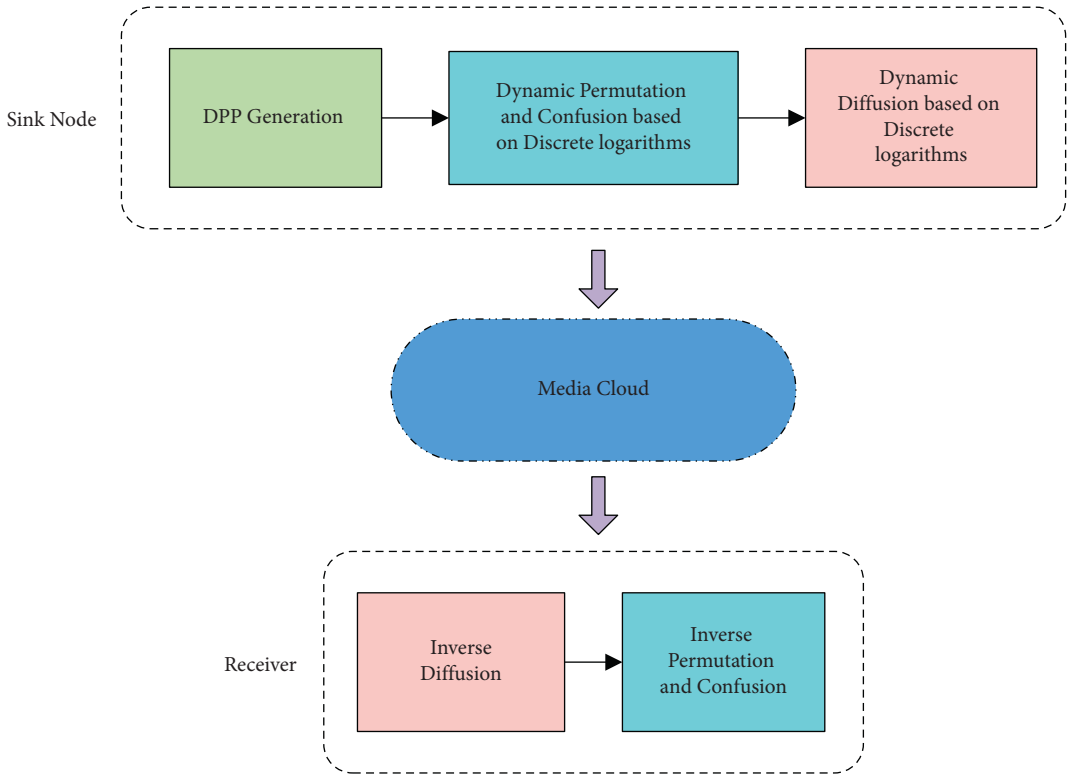


Figure 4: Secure image transmission between sink nodes and receivers.

improve the ability to resist CPAs. The specific process of TLM-Driven Global Permutation (GP) is as follows:

(i) Step 1: use the parameters $(r_1^{(1)}, r_2^{(1)}, x^{(1)})$ to iterate TLM $N \times N + r_3^{(1)}$ times. In order to avoid negative effects, discard the first $r_3^{(1)}$ chaotic state values.

(ii) Step 2: convert the obtained chaotic sequence $\mathbf{S}$ of length $N \times N$ into the integer sequence

$$\mathbf{S}_i = \left(\text{floor}\left(\mathbf{S}_i \times 10^{15}\right) \text{mod}\,(N \times N)\right) + 1, \quad (5)$$

where $i = 1, 2, \ldots, N \times N$, $\text{floor}(\cdot)$ returns the integer part of an operand.

(iii) Step 3: stretch the plain image $\mathbf{P}$ of size $N \times N$ into the 1D sequence $\tilde{\mathbf{P}}$.

(iv) Step 4: calculate the index

$$I = (\mathbf{S}_i \text{mod}\, 32) + 1, \quad (6)$$

of 32 DPPs and the permutation position

$$\alpha = ((\mathbf{S}_i + \beta_I) \text{mod}\,(N \times N)) + 1, \quad (7)$$

where $i = 1, 2, \ldots, N \times N$. Swap two pixels of $\tilde{\mathbf{P}}$ according to α.

3.1.3. TLM-Driven Negative-Positive Transformation. A nonlinear operation called Negative-Positive Transformation (NPT) is introduced by 2DCS-ETC to improve security. Similarly, we use TLM and DPPs to complete NPT instead of using a random matrix in the form of secret key. This can further reduce the resource overhead of sensor nodes and improve the ability to resist CPAs.

(i) Step 1: use parameter $(r_1^{(2)}, r_2^{(2)}, x^{(2)})$ to iterate TLM $N \times N + r_3^{(2)}$ times. In order to avoid negative effects, discard the first $r_3^{(2)}$ chaotic state values.

(ii) Step 2: convert the obtained chaotic sequence $\mathbf{S}$ of length $N \times N$ into the bit sequence

$$\mathbf{S}_i = \text{floor}\left(\mathbf{S}_i \times 10^{15}\right) \bmod 2, \tag{8}$$

where $i = 1, 2, \ldots, N \times N$.

(iii) Step 3: according to $\mathbf{S}$, perform the following NPT operation on $\widetilde{\mathbf{P}}$.

$$\mathbf{C}_i = \begin{cases} \widetilde{\mathbf{P}}_i, & \mathbf{S}_i = 1, \\ 255 - \widetilde{\mathbf{P}}_i, & \mathbf{S}_i = 0, \end{cases} \tag{9}$$

where $i = 1, 2, \ldots, N \times N$.

(iv) Step 4: reshape $\mathbf{C}$ into the 2D cipher image.

3.1.4. TLM-Driven 2D-CS. If the chaotic sequence generated by the chaotic system is assembled into a complete measurement matrix, its performance is usually almost the same as other commonly used random matrices [11]. Moreover, compared with directly using a random matrix and treating it as secret key, the chaotic measurement matrix can significantly save the resource overhead of sensor nodes. In SIT-SS, TLM is used to generate the measurement matrices required for 2D-CS. Suppose the size of the measurement matrices $\mathbf{A}$ and $\mathbf{B}$ to be created is $M \times N$ ($M \ll N$); the specific process of TLM-Driven 2D-CS is as follows:

(i) Step 1: use the parameters $(r_1^{(3)}, r_2^{(3)}, x^{(3)})$ to iterate TLM $N \times N + r_3^{(3)}$ times. In order to avoid negative effects, discard the first $r_3^{(3)}$ chaotic state values.

(ii) Step 2: arrange the obtained chaotic sequence into the square matrix $\mathbf{S}$ of size $N \times N$.

(iii) Step 3: take M rows from the orthogonal basis of $\hat{\mathbf{S}}$ as the measurement matrix $\mathbf{A}$.

(iv) Step 4: repeat Step 1 through Step 3; create the measurement matrix $\mathbf{B}$ in a similar manner.

(v) Step 5: use $\mathbf{A}$ and $\mathbf{B}$ to obtain the 2D measurements of the cipher image $\mathbf{C}$.

In addition to the improvements made above, the other steps of SIT-SS are basically the same as those of 2DCS-ETC, which are not repeated here. Since we have introduced TLM and DPPs in SIT-SS, the security of image transmission between sensor nodes and sink nodes has become higher, and the resource requirements for sensors are also lower. Significantly, SIT-SS still maintains the advantages of 2DCS-ETC, which is demonstrated and discussed in Section 4.1. To save hardware resources, we directly use $r_1^{(1)}, r_2^{(1)}, x^{(1)}, r_1^{(2)}, r_2^{(2)}, x^{(2)}, r_1^{(3)}, r_2^{(3)}, x^{(3)}$ as the secret key of SIT-SS.

3.2. Transmission between Sink Nodes and Receivers. In SIT-SR, we use 2D-LSCM [13] which has better chaotic performance to generate chaotic sequences. Moreover, discrete logarithms and DPPs are introduced to achieve secure image transmission between sink nodes and receivers. It should be noted that through the use of discrete logarithms and our targeted design, DPPs can be directly sent out in plaintext form by sink nodes. When decrypting, receivers can directly use DPPs that arrived in plaintext form. In other words, DPPs are not one-time pad secret keys, nor are they secret parameters. Next, we introduce the specific process of SIT-SR, as shown in Figure 5.

3.2.1. Secret Key and Chaotic System Parameters. In order to avoid the secret key issues pointed out in some cryptanalysis works and simplify the generation process of chaotic system parameters [14, 41, 42], we set the secret key K in this stage as a binary sequence with the length of 270 bits. Namely, $K = a_1 a_2 \cdots a_{270}$. In specific implementation, we directly use nine 32-bit unsigned integers $b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9$ to generate three sets of parameters $(\hat{x}_0^{(1)}, \hat{y}_0^{(1)}, \hat{r}_0^{(1)})$, $(\hat{x}_0^{(2)}, \hat{y}_0^{(2)}, \hat{r}_0^{(2)})$, $(\hat{x}_0^{(3)}, \hat{y}_0^{(3)}, \hat{r}_0^{(3)})$ for 2D-LSCM. As shown in equation (11), this means that the 30×9 bits of K correspond to the 30 bits of each unsigned integer, respectively.

$$\begin{cases} \hat{x}_0^{(1)} = (b_1 \times 2 + 1) \times 2^{-32}, \\ \hat{y}_0^{(1)} = (b_2 \times 2 + 1) \times 2^{-32}, \\ \hat{r}_0^{(1)} = (b_3 \times 2 + 1) \times 2^{-32}, \\ \hat{x}_0^{(2)} = (b_4 \times 2 + 1) \times 2^{-32}, \\ \hat{y}_0^{(2)} = (b_5 \times 2 + 1) \times 2^{-32}, \\ \hat{r}_0^{(2)} = (b_6 \times 2 + 1) \times 2^{-32}, \\ \hat{x}_0^{(3)} = (b_7 \times 2 + 1) \times 2^{-32}, \\ \hat{y}_0^{(3)} = (b_8 \times 2 + 1) \times 2^{-32}, \\ \hat{r}_0^{(3)} = (b_9 \times 2 + 1) \times 2^{-32}, \end{cases} \tag{10}$$

where

$$\begin{cases} b_1 = a_1 a_2 \cdots a_{30}, \\ b_2 = a_{31} a_{32} \cdots a_{60}, \\ b_3 = a_{61} a_{62} \cdots a_{90}, \\ b_4 = a_{91} a_{92} \cdots a_{120}, \\ b_5 = a_{121} a_{122} \cdots a_{150}, \\ b_6 = a_{151} a_{152} \cdots a_{180}, \\ b_7 = a_{181} a_{182} \cdots a_{210}, \\ b_8 = a_{211} a_{212} \cdots a_{240}, \\ b_9 = a_{241} a_{242} \cdots a_{270}. \end{cases} \tag{11}$$

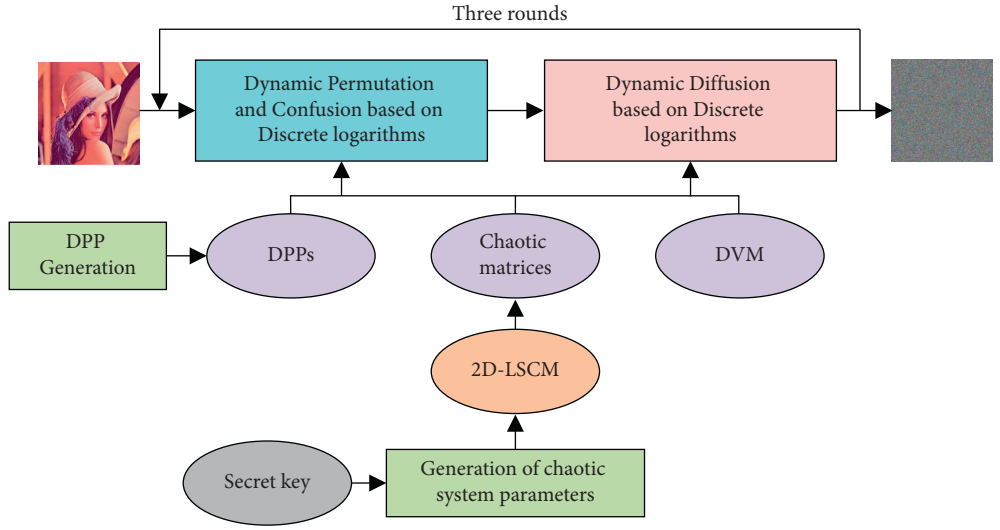


Figure 5: Flowchart of SIT-SR.

Besides, these three sets of chaotic system parameters $(\hat{x}_0^{(1)}, \hat{y}_0^{(1)}, \hat{r}^{(1)})$, $(\hat{x}_0^{(2)}, \hat{y}_0^{(2)}, \hat{r}^{(2)})$, $(\hat{x}_0^{(3)}, \hat{y}_0^{(3)}, \hat{r}^{(3)})$ are used to generate chaotic matrices for the encryption process.

3.2.2. DPP Generation. The generation process of DPPs in SIT-SR is exactly the same as SIT-SS. And we mark 32 DPPs used in SIT-SR as **H**.

3.2.3. Dynamic Permutation and Confusion Based on Discrete Logarithms. As mentioned above, discrete logarithms and DPPs are introduced in dynamic permutation and confusion based on discrete logarithms (DPC-D), so as to enhance the security of image transmission. Specifically, compared with some existing permutation operations, DPC-D has the following advantages:

(1) Use **H** to further perturb the permutation results and adopt different perturbation strategies for the row index and column index. Therefore, the permutation results depend not only on the secret key, but also on **H**.

(2) Based on discrete logarithms, **H** and the sorting results of the chaotic matrix **S** are used to nonlinearly transform the pixel value of each plain image pixel, thereby further improving the security of image transmission.

In order to better describe the specific steps of DPC-D, an algorithm is provided in Algorithm 1.

3.3. Dynamic Diffusion Based on Discrete Logarithms. To further improve security, dynamic diffusion based on discrete logarithms (DD-D) also adopts discrete logarithms and **H**. Specifically, compared with some existing diffusion operations, DD-D has the following advantages:

(1) Considering that multipixel diffusion is of little significance, single-pixel diffusion is adopted, thereby reducing the amount of computation

(2) The nonlinearity of the diffusion process is improved by introducing discrete logarithms; thus the security of image transmission is further improved

In order to better describe the specific steps of DD-D, an algorithm is provided in Algorithm 2.

Since a symmetric encryption structure is adopted in SIT-SR, the decryption process is actually constituted by the corresponding inverse operations of the encryption operations. With the received DPPs and the agreed secret key K, receivers can decrypt the plain image from the cipher image. To save space, these inverse operations are not repeated here.

4. Simulation Tests and Analyses

In this section, extensive simulation tests are performed to demonstrate the superiority of ITS-CDD. ITS-CDD is an image transmission scheme composed of two parts, and the resource conditions and design goals of each part are different. Therefore, SIT-SS is compared with 2DCS-ETC for resource-constrained environments, whereas SIT-SR is compared with more versatile schemes for general application environments. Without loss of generality, randomly generated secret keys are used to complete the tests. Table 2 lists the hardware and software configurations used in the tests.

4.1. Simulation Tests for SIT-SS. Since reducing the resource consumption of sensors and improving the security of image transmission is our goal in designing SIT-SS, the analysis and verification of SIT-SS are mainly focused on these two aspects. The test images used are eight images used in [12].

4.1.1. Encryption and Decryption. Four plain images Lena, Boats, House, and Parrots are shown in Figure 6. Their

Require: the plain image **P** with the size of $N \times N$, the chaotic matrix **S** with the size of $N \times N$, the dynamic perturbation parameters **H** with the size of 1×32 and the discrete logarithm value matrix **DV M** with the size of 128×256.

(1) Set $\mathbf{T} \in N^{N \times N}$;
(2) Set the sum hs of the dynamic perturbation parameters to 0;
(3) Set the row index value g used to represent the adopted generator to 0;
(4) Set the index value idx used to represent the adopted dynamic perturbation parameters to 0;
(5) Sort each column of **S** in ascending order, thus get the column index matrix **O** and sorted result **B**;
(6) **for** $i = 1$ to 32 **do**
(7) $\quad hs = hs + \mathbf{H}_i$;
(8) **end for**
(9) Calculate the row index value of the generator to be used, namely let $g = (hs \bmod 128) + 1$;
(10) **for** $i = 1$ to N **do**
(11) $\quad$ Sort $\mathbf{B}_i$ in ascending order and obtain the row index vector **v**;
(12) $\quad$ **for** $j = 1$ to N **do**
(13) $\quad\quad$ idx $= ((i-1) \times N + j \bmod 32) + 1$;
(14) $\quad\quad \mathbf{T}_{((\mathbf{O}_{i,j}+\mathbf{H}_{\text{idx}}) \bmod N)+1, ((j+hs) \bmod N)+1} = \mathbf{DVM}_{g, ((\mathbf{P}_{\mathbf{O}_{i,\mathbf{v}_j}, \mathbf{v}_j} + \mathbf{H}_{\text{idx}} + \mathbf{v}_j) \bmod 256)+1} - 1$;
(15) $\quad$ **end for**
(16) **end for**

Ensure: the permuted and transformed image **T**.

Algorithm 1: DPC-D algorithm.

Require: the permuted and transformed image **T** with the size of $N \times N$, the chaotic matrix **R** with the size of $N \times N$, the dynamic perturbation parameters **H** with the size of 1×32 and the discrete logarithm value matrix **DV M** with the size of 128×256.

(1) Set $\mathbf{C} \in N^{N \times N}$;
(2) Convert **R** into the integer matrix **IR** with the same format as the pixels of **T**, namely $\mathbf{IR} = (\lfloor \mathbf{R} \times 2^{32} \rfloor) \bmod 256$;
(3) Set the bitwise XOR result hx of the dynamic perturbation parameters to 0;
(4) Set the row index values g_1, g_2 used to represent the adopted generators to 0 s;
(5) **for** $i = 1$ to 32 **do**
(6) $\quad hx = \text{bitxor}(hx, \mathbf{H}_i)$;
(7) **end for**
(8) Calculate the row index values of the generators to be used, let $g_1 = (\text{bitxor}(hx, \mathbf{IR}_{1,1}) \bmod 128) + 1$, $g_2 = (\text{bitxor}(g_1, \mathbf{IR}_{N,N}) \bmod 128) + 1$;
(9) $\mathbf{tmp}_{:,1} = (\mathbf{T}_{:,1} + \mathbf{DVM}_{g_1, (\mathbf{T}_{:,N}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{:,1}+1)}) \bmod 256$;
(10) **for** $i = 2$ to N **do**
(11) $\quad \mathbf{tmp}_{:,i} = (\mathbf{T}_{:,i} + \mathbf{DVM}_{g_1, (\mathbf{tmp}_{:,i-1}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{:,i}+1)}) \bmod 256$;
(12) **end for**
(13) $\mathbf{C}_{1,:} = (\mathbf{tmp}_{1,:} + \mathbf{DVM}_{g_1, (\mathbf{tmp}_{N,:}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{1,:}+1)}) \bmod 256$;
(14) **for** $i = 2$ to N **do**
(15) $\quad \mathbf{C}_{i,:} = (\mathbf{tmp}_{i,:} + \mathbf{DVM}_{g_1, (\mathbf{C}_{i-1,:}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{i,:}+1)}) \bmod 256$;
(16) **end for**

Ensure: the diffused image **C**.

Algorithm 2: DD-D algorithm.

corresponding cipher images and decrypted images generated in SIT-SS are also provided. As can be seen from these images, the cipher images are similar to noise, attackers cannot obtain useful information from them, and there are no significant visual differences between the decrypted images and corresponding plain images.

4.1.2. Reconstruction Quality. Researchers often use Peak Signal-to-Noise Ratio (PSNR) to evaluate image reconstruction quality. Generally, a higher PSNR value indicates a better reconstruction quality. The definition of PSNR is as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{(1/(M \times N)) \sum_{i=1}^{M} \sum_{i=1}^{N} [R(i,j) - P(i,j)]^2}, \tag{12}$$

where $M \times N$ is the size of the reconstructed image R and original image P. PSNR versus sampling ratio for 2DCS-ETC and SIT-SS is listed in Table 3. As can be seen from Table 3, SIT-SS can achieve the same or slightly different PSNR

Table 2: Software and hardware configurations used in simulation tests.

Configuration item	Description
CPU	Intel Xeon CPU E3-1231 v3 3.40 GHz
Memory capacity	8 GB
Operating system	Windows 7 Ultimate (64 bit)
Test platform	MATLAB R2017a (9.2.0538062)

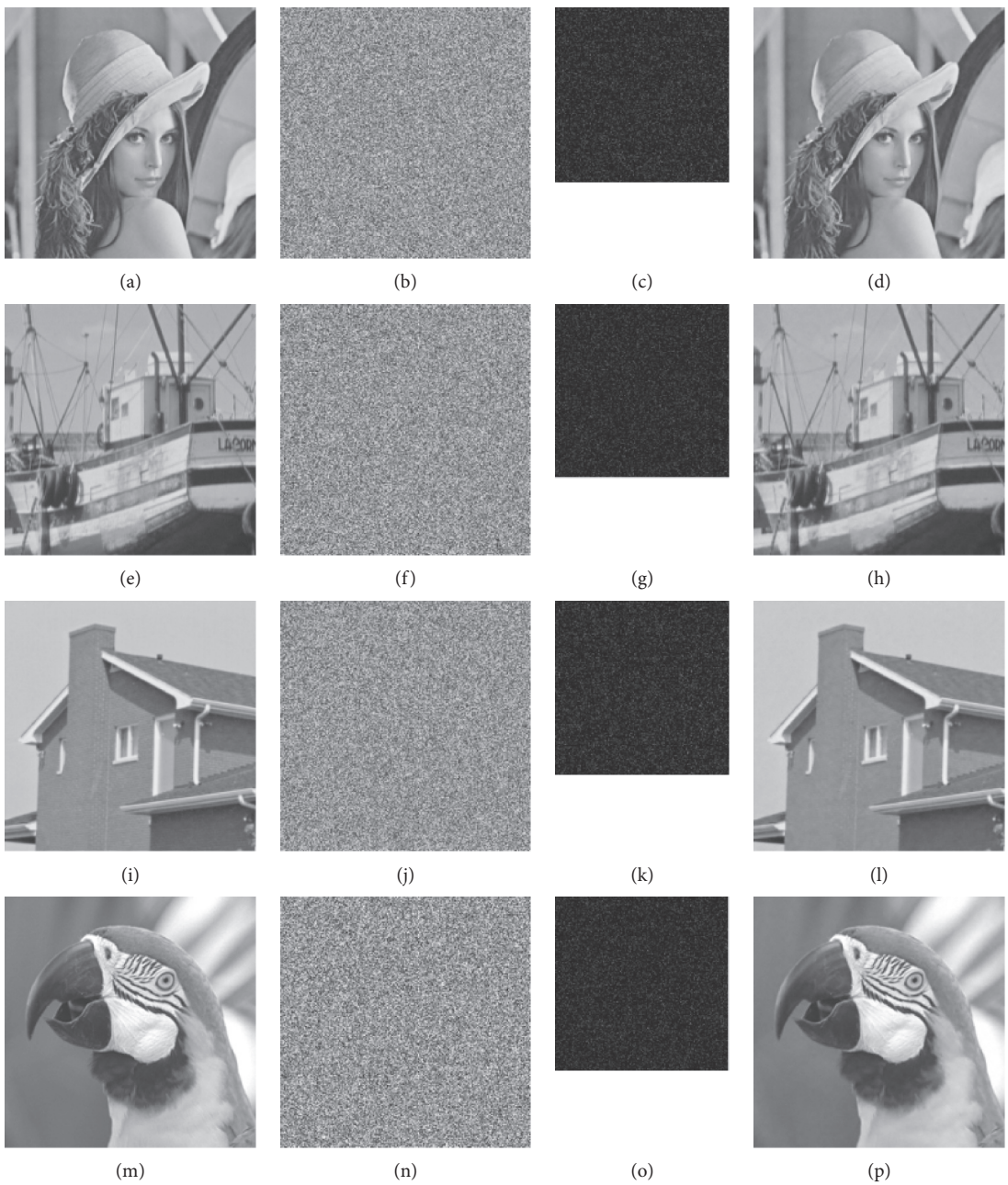


Figure 6: Encryption and reconstruction of four plain images: (a) plain image Lena; (b) cipher image of (a); (c) compressed cipher image of (a); (d) reconstructed image of (c); (e) plain image of Boats; (f) cipher image of (e); (g) compressed cipher image of (e); (h) reconstructed image of (g); (i) plain image of House; (j) cipher image of (i); (k) compressed cipher image of (i); (l) reconstructed image of (k); (m) plain image of Parrots; (n) cipher image of (m); (o) compressed cipher image of (m); (p) reconstructed image of (o).

values as 2DCS-ETC. And when sampling ratio is greater than 0.6, its image reconstruction quality has obvious advantages.

4.1.3. Secret Key. In 2DCS-ETC, the random permutation matrix and random binary integer matrix are used as the secret key, thereby obtaining a huge key space. However, in

Table 3: PSNR (dB) of 2DCS-ETC and SIT-SS under different sampling ratios.

Image	Scheme	Sampling ratio								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Lena	2DCS-ETC	25.1608	**30.7238**	**33.7894**	36.3407	38.5169	39.0429	40.3088	41.1759	41.7017
	SIT-SS	**25.5232**	30.6675	33.7680	**36.3926**	**38.6439**	**40.4935**	**42.6165**	**44.9562**	**47.9543**
Barbara	2DCS-ETC	22.0135	**25.8589**	**28.8967**	**31.7166**	**34.2147**	35.6315	37.3399	39.0916	40.2334
	SIT-SS	**22.1990**	25.8260	28.8175	31.6593	34.1841	**36.4853**	**38.8797**	**41.5256**	**45.0071**
Boats	2DCS-ETC	**24.6290**	**29.7913**	32.8112	35.3011	37.4047	38.0778	39.1809	40.3438	41.5691
	SIT-SS	24.1095	29.6903	**32.8657**	**35.3405**	**37.4213**	**39.3371**	**41.3561**	**43.5876**	**46.6325**
Cameraman	2DCS-ETC	**22.3125**	**28.7665**	**31.8389**	**34.2630**	**36.4953**	36.9751	38.2231	39.0237	39.7931
	SIT-SS	21.8324	28.7046	31.8201	34.2575	36.4449	**38.6534**	**40.9500**	**43.3287**	**46.3889**
Foreman	2DCS-ETC	29.5930	**35.8537**	**38.1558**	38.5413	39.4111	39.8570	40.2641	40.3616	40.6835
	SIT-SS	**29.7572**	35.7866	38.1025	**39.8524**	**41.4978**	**42.9477**	**44.5395**	**46.3175**	**48.4910**
House	2DCS-ETC	**28.9899**	**34.1209**	36.0911	**37.5883**	38.9428	39.1408	40.0489	41.1202	42.1401
	SIT-SS	28.3764	34.1188	**36.1386**	37.5654	**38.9717**	**40.3546**	**42.0492**	**44.1757**	**47.0669**
Monarch	2DCS-ETC	**22.3624**	**29.0054**	32.5729	**35.6175**	**37.9890**	38.1359	39.3164	40.0515	40.9473
	SIT-SS	22.0071	28.8123	**32.6323**	35.3864	37.8005	**39.9611**	**42.1377**	**44.5657**	**47.4934**
Parrots	2DCS-ETC	**25.9860**	**33.0968**	**35.7941**	**37.8344**	38.0873	39.1139	39.7870	40.3589	40.6380
	SIT-SS	25.7881	32.9618	35.6443	37.7954	**39.5943**	**41.2956**	**43.0655**	**45.0295**	**47.6101**

Each bold value means that one of the two compared schemes has a higher PSNR value than the other.

resource-constrained environments, it is not suitable to use such secret key that requires a large amount of storage space. For example, if the size of the plain image is 1024×1024, the secret key used would be at least 2 228 224 bytes in length. In addition, in the encryption and decryption process, the generation and storage of two measurement matrices also bring significant resource requirements. However, in SIT-SS, we only need to store six floating-point numbers used as the secret key. Meanwhile, SIT-SS also enjoys a large enough key space, which is about 10^{133}. Apparently, such a large key space is sufficient to resist brute force attacks.

4.1.4. Chosen-Plaintext Attack. As we know, CPAs are the reasons why some encryption schemes are cracked. It is generally believed that a secure encryption scheme should be able to resist CPAs. Derived from system times and last encryption times, DPPs always change dynamically and will be affected by many factors, thereby ensuring the diversity and unpredictability of equivalent key streams. Without relying on external algorithms such as hash algorithms, the diversity of equivalent key streams brings excellent resistance to CPAs.

4.2. Simulation Tests for SIT-SR. The simulation tests presented in this subsection are to demonstrate the superiority of SIT-SR in terms of security and encryption efficiency. The test images used for SIT-SR are from The USC-SIPI Image Database (https://sipi.usc.edu/database/).

4.2.1. Encryption and Decryption of Different Types of Images. For different types of images, we uniformly treat them as 8-bit grayscale images in ITS-CDD. Specifically, for images with a pixel depth of less than 8 bits, we directly process them as 8-bit grayscale images, whereas for images with pixel depth greater than 8 bits, we encrypt them in groups of 8 bits. For example, if we need to encrypt an image with a pixel depth of 16 bits, we can encrypt the lower 8 bits and higher 8 bits of each plain image pixel separately. Figure 7 shows the encryption and decryption effects of SIT-SR for different types of images. One can see that SIT-SR has excellent encryption effects for different types of images. The generated cipher images are very similar to noise images, and attackers cannot directly obtain any valuable information from these cipher images.

4.2.2. Key Space and Key Sensitivity. Since the key space would affect the ability to resist brute force attacks, a secure encryption scheme should have a sufficiently large key space [50]. We carefully design the secret key, which not only solves the issues about equivalent secret keys, but also expands the key space to 2^{270}. Therefore, SIT-SR is excellent in terms of the ability to resist brute force attacks.

It is well known that a secure encryption scheme should be able to achieve superior confusion effect [50, 51]. That is, one smallest change in the secret key should make the cipher image change significantly. To evaluate the performance of SIT-SR in this regard, we randomly generated the secret key

$$K_R = 03D7\ D6F3E884\ 829564ED\ BA77D868\ 3B811AE4\ FC8655CC\ 7EE7BC30\ 5537BFEA\ 2EE2238E. \quad (13)$$

Using K_R, we encrypted elaine.512.tiff to obtain the corresponding cipher image $\check{C}_R$. Next, we changed one bit of K_R to get two secret keys $K_R^{(1)}$, $K_R^{(2)}$ with single smallest changes. These two changed secret keys also were used to encrypt elaine.512.tiff, thus obtaining the corresponding cipher images $\check{C}_R^{(1)}$, $\check{C}_R^{(2)}$. Finally, the difference images between $\check{C}_R^{(1)}$, $\check{C}_R^{(2)}$, and $\check{C}_R$ were calculated. As can be seen from Figure 8, the difference images between $\check{C}_R^{(1)}$, $\check{C}_R^{(2)}$, and $\check{C}_R$ look similar to an ordinary cipher image. This means that the key sensitivity of SIT-SR in the encryption process is extraordinary.

Similarly, in order to verify the key sensitivity of SIT-SR in the decryption process, $K_R^{(1)}$ and $K_R^{(2)}$ were adopted to

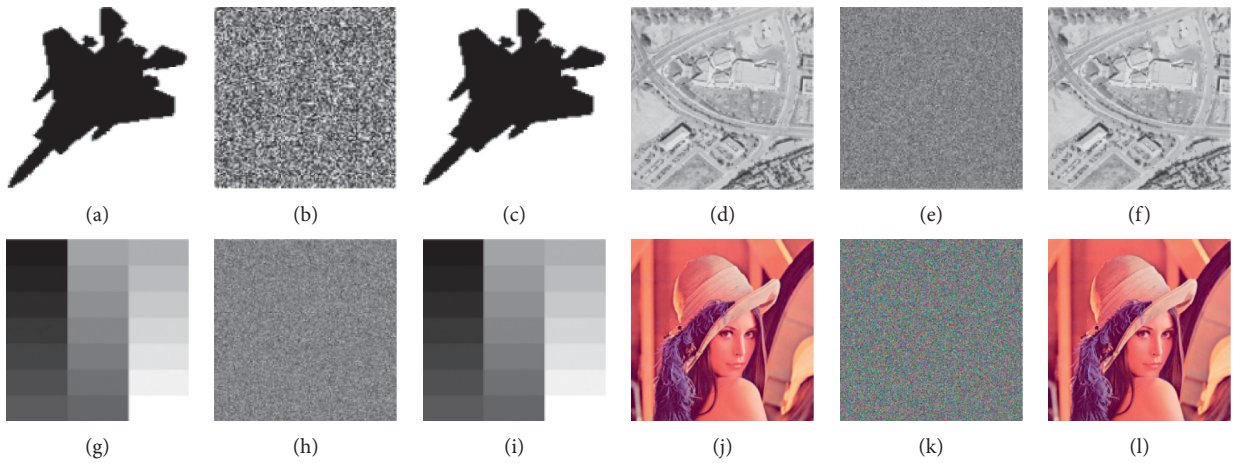


Figure 7: Encryption and decryption effects for different types of images: (a) f15.png; (b) cipher image of (a); (c) decrypted image of (b); (d) 5.2.09.tiff; (e) cipher image of (d); (f) decrypted image of (e); (g) gray21.512.tiff; (h) cipher image of (g); (i) decrypted image of (h); (j) 4.2.04.tiff; (k) cipher image of (j); (l) decrypted image of (k); (m) f15.png; (n) cipher image of (m); (o) decrypted image of (n); (p) f15.png; (q) cipher image of (p); (r) decrypted image of (q); (s) f15.png; (t) cipher image of (s); (u) decrypted image of (t); (v) f15.png; (w) cipher image of (v); (x) decrypted image of (w).

decrypt $\check{C}_R$. The test results are shown in Figure 9. Once again, judging from the difference image between the wrongly decrypted images, the key sensitivity of SIT-SR in the decryption process is excellent.

For measuring the degree of changes between images, NPCR (Number of Pixel Change Ratio) and UACI (Unified Average Change in Intensity) are commonly used indicators [13]. Among them,

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i,j} \mathbf{D}_{i,j} \times 100\%, \tag{14}$$

refers to the ratio of the pixels that change, whereas

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i,j} \frac{\left|\mathbf{C}_{i,j}^{(1)} - \mathbf{C}_{i,j}^{(2)}\right|}{255} \times 100\%, \tag{15}$$

refers to the average intensity of the pixel value changes. In equations (14) and (15), $M \times N$ is the size of the images, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, and $\mathbf{D}$ is the difference matrix between the image $\mathbf{C}^{(1)}$ before the change and the image $\mathbf{C}^{(2)}$ after the change. If $\mathbf{C}_{i,j}^{(1)} \neq C_{i,j}^{(2)}$, then $\mathbf{D}_{i,j} = 1$. Otherwise, $\mathbf{D}_{i,j} = 0$. In order to quantitatively analyze the key sensitivity of SIT-SR, we calculated the NPCR and UACI values between the cipher images before and after the secret key changes. As one can see from Table 4, all the test results are very close to the ideal values, which means that SIT-SR does have extremely high key sensitivity.

4.2.3. Pixel Value Distribution. Unlike plain images, cipher images must have extremely uniform pixel value distributions; otherwise attackers will have the opportunity to conduct statistics based attacks [50, 51]. In order to visually demonstrate the pixel value distribution characteristics of the plain images and the cipher images generated by SIT-SR, the pixel value distribution histograms of these images are plotted. As can be seen from Figure 10, the pixel distributions of the plain images are relatively concentrated, whereas the pixels of the cipher images are extremely uniformly distributed throughout the entire value range.

In addition, the histogram variance analysis and chi-square test analysis are also performed on the cipher images to qualitatively analyze the ability of SIT-SR to resist statistical attacks. In general, if the histogram variance of a cipher image is smaller, then the uniformity of the cipher image is higher. The histogram variance of an 8-bit grayscale image can be defined as follows:

$$V(\mathbf{H}) = \frac{1}{256^2} \sum_{i=1}^{256} \sum_{j=1}^{256} \frac{1}{2}\left(p_i - p_j\right)^2, \tag{16}$$

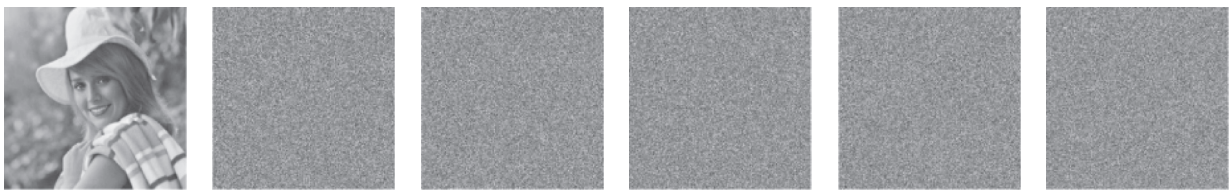
where $\mathbf{H} = \{p_1, p_2, \ldots, p_{256}\}$; p_i and p_j are the numbers of the pixels whose grayscale values are equal to $i - 1$ and $j - 1$. Table 5 lists the histogram variance test results of some images. These images include one random image, the 8-bit grayscale image Lena, and its cipher images generated by different schemes.

From Table 5, one can see that the histogram variance of the plain image is very large, which means that its pixel value distribution is extremely uneven, whereas among the cipher images generated by the four image encryption schemes, the cipher image of SIT-SR has the smallest histogram variance, which indicates that this cipher image has the most uniform pixel value distribution and is closest to the random image.

Another way to quantitatively analyze the uniformity of a cipher image is the chi-square test. The chi-square value of a cipher image can be calculated as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{\left(s_i - H \times W \times p\right)^2}{H \times W \times p}, \tag{17}$$

where $H \times W$ is the height and width of the cipher image, s_i is the number of pixels whose pixel value is $i - 1$, n is the



(a) (b) (c) (d) (e) (f)

Figure 8: Key sensitivity test for encryption process: (a) elaine.512.tiff; (b) $\check{C}_R$ obtained by K_R; (c) $\check{C}_R^{(1)}$ obtained by $K_R^{(1)}$; (d) $\check{C}_R^{(2)}$ obtained by $K_R^{(2)}$; (e) difference image between $\check{C}_R$ and $\check{C}_R^{(1)}$; (f) difference image between $\check{C}_R$ and $\check{C}_R^{(2)}$.

(a) (b) (c) (d) (e)

Figure 9: Key sensitivity test for decryption process: (a) $\check{C}_R$ obtained by K_R; (b) decrypted image $\check{C}_R$ obtained by K_R; (c) decrypted image $\check{C}_R^{(1)}$ obtained by $K_R^{(1)}$; (d) decrypted image $\check{C}_R^{(2)}$ obtained by $K_R^{(2)}$; (e) difference image between $\check{C}_R^{(1)}$ and $\check{C}_R^{(2)}$.

Table 4: UPCR and UACI values between cipher images when secret key changes.

Change	NPCR (%)	UACI (%)
Lowest bit of b_1 is inverted	99.6121	33.4601
Lowest bit of b_2 is inverted	99.6028	33.4625
Lowest bit of b_3 is inverted	99.6097	33.4837
Lowest bit of b_4 is inverted	99.6013	33.4633
Lowest bit of b_5 is inverted	99.6009	33.4764
Lowest bit of b_6 is inverted	99.6174	33.4512
Lowest bit of b_7 is inverted	99.6075	33.4810
Lowest bit of b_8 is inverted	99.6122	33.4706
Lowest bit of b_9 is inverted	99.6130	33.4562
Random image	**99.6094**	**33.4635**

The bold values are the ideal values.

number of all possible pixel values (for 8-bit grayscale images, $n = 256$), and $p = 1/n$. Next, the critical value $\chi^2_{0.05}(255)$ of the chi-square test at the significant level $\alpha = 0.05$ can be determined, which is 293.2478. If a cipher image has a chi-square value less than 293.2478, then this image can be considered to have passed the chi-square test; that is, its pixel value distribution is statistically uniform. Consequently, the smaller the chi-square value of a cipher image is, the better its uniformity is. As can be seen from Table 6, for some commonly used test images, the cipher images generated by SIT-SR have all passed the chi-square test. And in most cases, SIT-SR performs better than another scheme.

4.2.4. Plain Image Sensitivity. When the plain image changes, the corresponding change degree of the cipher image is plain image sensitivity. Generally speaking, to effectively resist differential attacks, an image encryption scheme must have excellent plain image sensitivity. To intuitively show the plain image sensitivity of SIT-SR, we first encrypted some commonly used test images. Next, after changing one pixel bit for each plain image, the plain images with the smallest changes were encrypted. At last, we calculated the difference images between the cipher images obtained before and after the smallest changes. The relevant test results are shown in Figure 11. As one can see from Figure 11, each plain image has undergone only one smallest change, but almost all cipher pixels have changed. In addition to that, these significant differences are independent of where the plain images change and are very close to random images.

UPCR and UACI are also utilized to qualitatively analyze the plain image sensitivity of SIT-SR. The UPCR and UACI values between the cipher images obtained before and after the smallest changes of 15 common test plain images are shown in Table 7. Judging from the test results, SIT-SR has excellent plain image sensitivity. The test results of SIT-SR are closer to the ideal values 99.6094% and 33.4635% and perform better in terms of stability.

4.2.5. Information Entropy. Information entropy is an indicator that can be used to measure the randomness or disorder of an information source. If the information

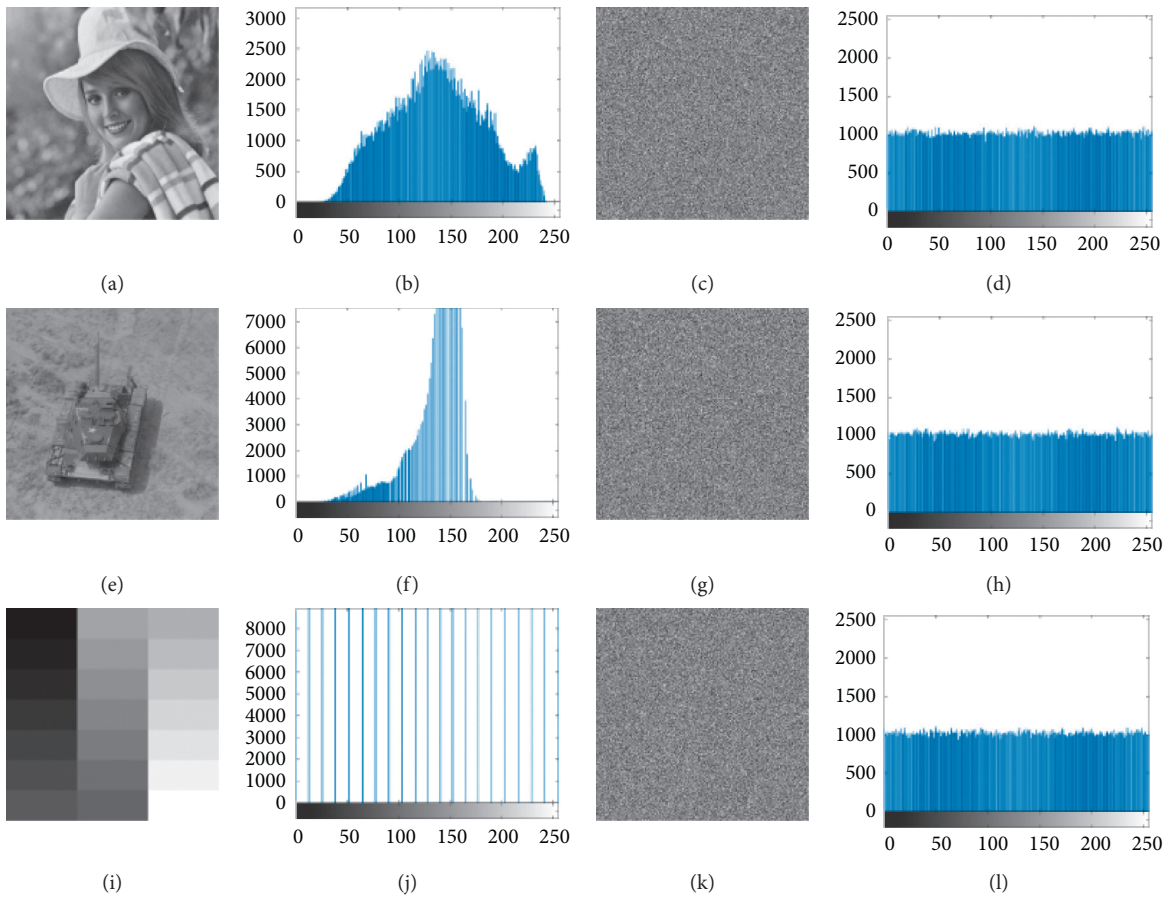


Figure 10: Pixel value distribution histograms of elaine.512.tiff, 7.1.03.tiff, gray21.512.tiff, and their corresponding cipher images: (a) elaine.512.tiff; (b) histogram of (a); (c) cipher image of (a); (d) histogram of (c); (e) 7.1.03.tiff; (f) histogram of (e); (g) cipher image of (e); (h) histogram of (g); (i) gray21.512.tiff; (j) histogram of (i); (k) cipher image of (i); (l) histogram of (k).

Table 5: Histogram variance test results of some images.

Algorithm	Image	Variance
	Lena256.bmp	33860.0546
[13]	Cipher image	266.7578
[16]	Cipher image	260.7188
[17]	Cipher image	276.3906
SIT-SR	Cipher image	**257.1094**
	Random image	253.8946

The bold value means that SIT-SR has the best test result among four compared schemes.

entropy of the information source is higher, it can be considered that the information source has higher randomness or disorder [18–20]. When it comes to an 8-bit grayscale image, the information entropy of the grayscale image can be calculated as follows:

$$H(\mathbf{S}) = -\sum_{i=1}^{n} \left(p(s_i) \times \log_2\ p(s_i)\right), \tag{18}$$

where n is the total number of symbols s_i; $p(s_i)$ is the occurrence probability of symbol s_i. For the 8-bit grayscale cipher images, the ideal value of the information entropy is 8 [18–20]. From Table 8, one can see that the information entropy of each cipher image generated by SIT-SR is very close to the ideal value 8. As shown in Table 9, compared with several image encryption schemes, the information entropy of the Lena cipher image generated by SIT-SR is closest to the ideal value 8. Therefore, SIT-SR performs best in terms of the information entropy.

In order to measure the randomness of cipher images more comprehensively, a measure named Local Shannon Entropy (LSE) is proposed [52]. This measure is increasingly adopted to verify the randomness of cipher images [13]. Mathematically, LSE can be defined as follows:

$$L_{q,s}(r) = \sum_{i=1}^{q} \frac{H(r_i)}{q}, \tag{19}$$

Table 6: Chi-square values of different cipher images.

Filename	Size	Chi-square value	
		[13]	SIT-SR
Lena256.bmp	256 × 256	255.8555	253.3035
5.1.10.tiff	256 × 256	261.3125	254.1953
5.1.12.tiff	256 × 256	256.2578	244.5328
5.1.13.tiff	256 × 256	274.8750	245.3797
5.2.08.tiff	512 × 512	252.7471	247.5434
5.2.09.tiff	512 × 512	274.3906	257.3434
5.3.01.tiff	1024 × 1024	236.3027	229.8125
7.1.02.tiff	512 × 512	252.9141	226.2197
7.1.03.tiff	512 × 512	248.8984	257.2324
7.1.04.tiff	512 × 512	281.2773	258.4043
7.1.05.tiff	512 × 512	275.1055	263.8584
boat.512.tiff	512 × 512	230.2256	232.7012
elaine.512.tiff	512 × 512	266.6377	230.0078
gray21.512.tiff	512 × 512	244.8789	245.3027
ruler.512.tiff	512 × 512	290.8057	223.2813
testpat.1k.tiff	1024 × 1024	258.6455	239.7627

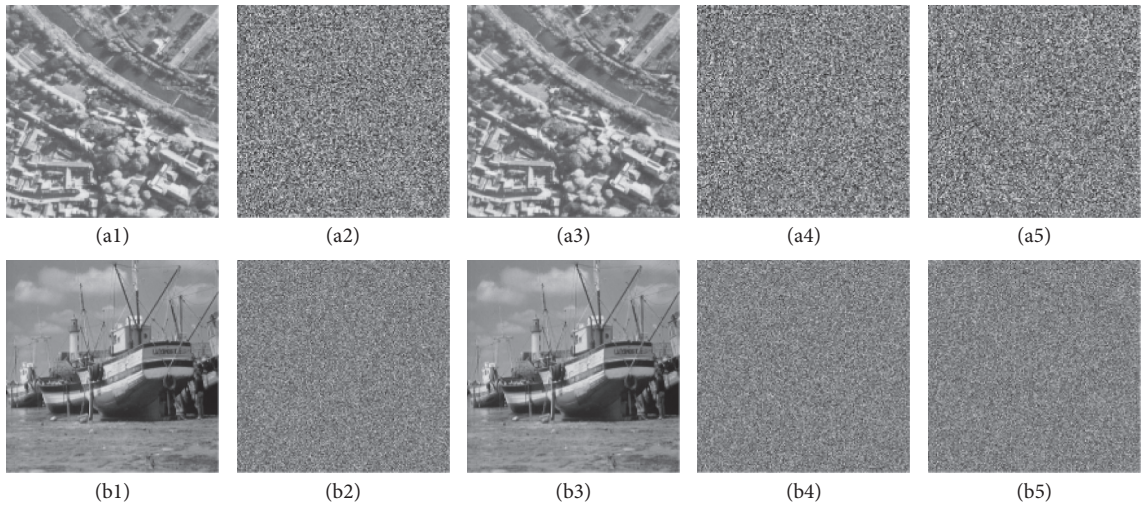


Figure 11: Plain image sensitivity test results for SIT-SR: (a1) 5.1.10.tiff; (a2) cipher image of (a1); (a3) the least significant bit of the pixel at (1,1) in (a1) is reversed; (a4) cipher image of (a3); (a5) difference image between (a2) and (a4); (b1) boat.512.tiff; (b2) cipher image of (b1); (b3) the least significant bit of the pixel at (256,256) in (b1) is reversed; (b4) cipher image of (b3); (b5) difference image between (b2) and (b4).

where $r_1, r_2, \ldots, r_q$ are q randomly selected nonoverlapping image blocks, s is the number of pixels in each block, and $H(r_i)$ is the information entropy of r_i. According to the test method suggested in [52], we carried out the LSE test on the cipher images generated by SIT-SR, and the relevant test results are shown in Table 10. Compared with two recent image encryption schemes, SIT-SR has the best performance in terms of standard deviation and pass rate.

4.2.6. Pixel Correlation. The extremely high correlation between adjacent pixels is one of the salient features of plain images and also one of the important reasons why traditional encryption schemes are not suitable for image encryption [50]. Therefore, a secure image encryption scheme should eliminate the correlation between adjacent pixels as much as possible. CC (correlation coefficient) is an effective indicator to measure the correlation between adjacent pixels, and its definition is as follows:

$$\mathrm{CC}_{ab} = \frac{E((a - E(a)) \times (b - E(b)))}{\sqrt{D(a) \times D(b)}}, \tag{20}$$

where a and b are the grayscale values of two adjacent pixels; $E(a)$ and $D(a)$ are the expectation and variance of the grayscale value a. In order to verify the performance of SIT-SR in terms of the pixel correlation, for the horizontal, vertical, and diagonal directions, we have randomly selected 20,000 pairs of adjacent pixels from each plain image and its corresponding cipher image to calculate the CCs. The relevant test results are shown in Table 11.

From Table 11, one can see that there are very high correlations between adjacent pixels of the plain images; that is, the absolute values of CCs are extremely high, whereas in

Table 7: NPCR and UACI test results on plain image sensitivity.

Filename	[13]		SIT-SR	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
5.1.10.tiff	99.6014	33.4774	99.6162	33.4645
5.1.12.tiff	99.6222	33.4668	99.6093	33.4642
5.1.13.tiff	99.6091	33.4782	99.6119	33.4597
5.2.08.tiff	99.6138	33.4596	99.6080	33.4589
5.2.09.tiff	99.6072	33.4496	99.6117	33.4521
5.3.01.tiff	99.6103	33.4551	99.6107	33.4595
7.1.02.tiff	99.6088	33.4749	99.6140	33.4635
7.1.03.tiff	99.6026	33.4930	99.6081	33.4836
7.1.04.tiff	99.6117	33.4699	99.6025	33.4645
7.1.05.tiff	99.6101	33.4766	99.6083	33.4643
boat.512.tiff	99.6045	33.4618	99.6098	33.4825
elaine.512.tiff	99.6139	33.4918	99.6098	33.4521
gray21.512.tiff	99.6069	33.4660	99.6122	33.4713
ruler.512.tiff	99.6113	33.4394	99.6115	33.4453
testpat.1k.tiff	99.6054	33.4571	99.6091	33.4624
Average	99.6080	33.4695	**99.6113**	**33.4598**
Std. Dev.	0.0042	0.0161	**0.0028**	**0.0045**

The bold values here emphasize that SIT-SR has better performance than the other scheme.

Table 8: Information entropy test results of plain images and cipher images.

Filename	Plain image	Cipher image
5.2.08.tiff	7.2010	7.9993
5.2.09.tiff	6.9940	7.9994
5.3.01.tiff	7.5237	7.9998
7.1.02.tiff	4.0045	7.9993
7.1.03.tiff	5.4957	7.9994
7.1.04.tiff	6.1074	7.9993
7.1.05.tiff	6.5632	7.9994
boat.512.tiff	7.1914	7.9994
elaine.512.tiff	7.5060	7.9994
gray21.512.tiff	4.3923	7.9993
ruler.512.tiff	0.5000	7.9994
testpat.1k.tiff	4.4077	7.9998

Table 9: Information entropy test results of Lena cipher images.

Scheme	[13]	[18]	[19]	[20]	SIT-SR
Information entropy	7.9992	7.9979	7.9909	7.9991	**7.9994**

The bold value here emphasizes that SIT-SR has better performance than other schemes.

the cipher images generated by SIT-SR, there is almost no correlation between adjacent pixels; that is, the absolute values of CCs are extremely low (< 0.006).

In addition, in order to more intuitively show the correlation changes between adjacent pixels caused by the encryption of SIT-SR, the correlation distribution charts of the plain image elaine.512.tiff and its corresponding cipher image are drawn. As can be seen from Figure 12, after the encryption processing of SIT-SR, there is almost no correlation between adjacent pixels in each direction.

4.2.7. Chosen-Plaintext Attack. In fact, almost all simulation tests related to security analysis can only ensure the security of image encryption schemes under ciphertext-only attacks (COAs) [51, 53]. This is exactly why some image encryption schemes have been broken. Among the four types of attacks, which are COAs, Known-Plaintext Attacks (KPAs), CPAs, and Chosen-Ciphertext Attacks (CCAs), CCAs are the most threatening ones, but the attack conditions required by them are practically meaningless [3, 50]. If attackers can choose cipher images arbitrarily, then they do not need to crack at all, because for any cipher image, they can directly recover its plain image. Therefore, it is generally believed that CPAs are the most threatening ones among common practical attacks. Actually, in the cryptanalysis works about image encryption, the vast majority of them adopt CPAs [3, 51]. Next, from the perspective of attackers, the ability of SIT-SR to resist CPAs is analyzed.

Table 10: LSE test results of different schemes.

Filename	[13]	[15]	SIT-SR
5.2.08	7.9023	7.9024	7.9022
5.2.09	7.9020	7.9021	7.9023
7.1.02	7.9020	7.9015	7.9021
7.1.03	7.9026	7.9019	7.9024
7.1.04	7.9019	7.9021	7.9023
boat.512	7.9018	7.9022	7.9024
gray21.512	7.9026	7.9026	7.9025
ruler.512	7.9041	7.9028	7.9026
Std.Dev.	0.0007	0.0004	**0.0002**
Pass/All	6/8	7/8	**8/8**

The bold values indicate that compared with the other two schemes, SIT-SR has the best performance in terms of standard deviation and pass rate.

Apparently, attackers will encounter several problems when they try to break SIT-SR with CPAs. Firstly, we assume that they could obtain the equivalent key streams of the encryption process from chosen plain images and corresponding cipher images. However, because SIT-SR introduce DPPs in serval encryption steps, the equivalent key streams they obtained cannot be used to recover other ordinary plain images, which are encrypted under different DPPs. Secondly, SIT-SR also performs nonlinear

Table 11: Correlation test results for adjacent pixels of plain images and their cipher images.

Filename	Horizontal		Vertical		Diagonal	
	Plain image	Cipher image	Plain image	Cipher image	Plain image	Cipher image
5.2.08.tiff	0.8906	0.0008	0.9322	−0.0038	0.8452	0.0028
5.2.09.tiff	0.8591	−0.0004	0.9000	−0.0018	0.8007	0.0020
5.3.01.tiff	0.9817	0.0015	0.9776	0.0044	0.8981	−0.0033
7.1.02.tiff	0.9446	−0.0024	0.9431	−0.0016	0.9003	0.0001
7.1.03.tiff	0.9317	−0.0021	0.9436	0.0002	0.9059	−0.0017
7.1.04.tiff	0.9672	0.0038	0.9771	0.0028	0.9552	−0.0059
7.1.05.tiff	0.9108	0.0038	0.9425	0.0058	0.8919	0.0024
boat.512.tiff	0.9711	0.0030	0.9394	0.0048	0.9245	0.0002
elaine.512.tiff	0.9720	−0.0011	0.9761	0.0010	0.9696	0.0016
gray21.512.tiff	0.9998	−0.0023	0.9968	−0.0036	0.9966	0.0031
ruler.512.tiff	0.4702	−0.0026	0.4524	0.0004	−0.0312	−0.0023
testpat.1k.tiff	0.7992	0.0035	0.7501	0.0051	0.6997	0.0005

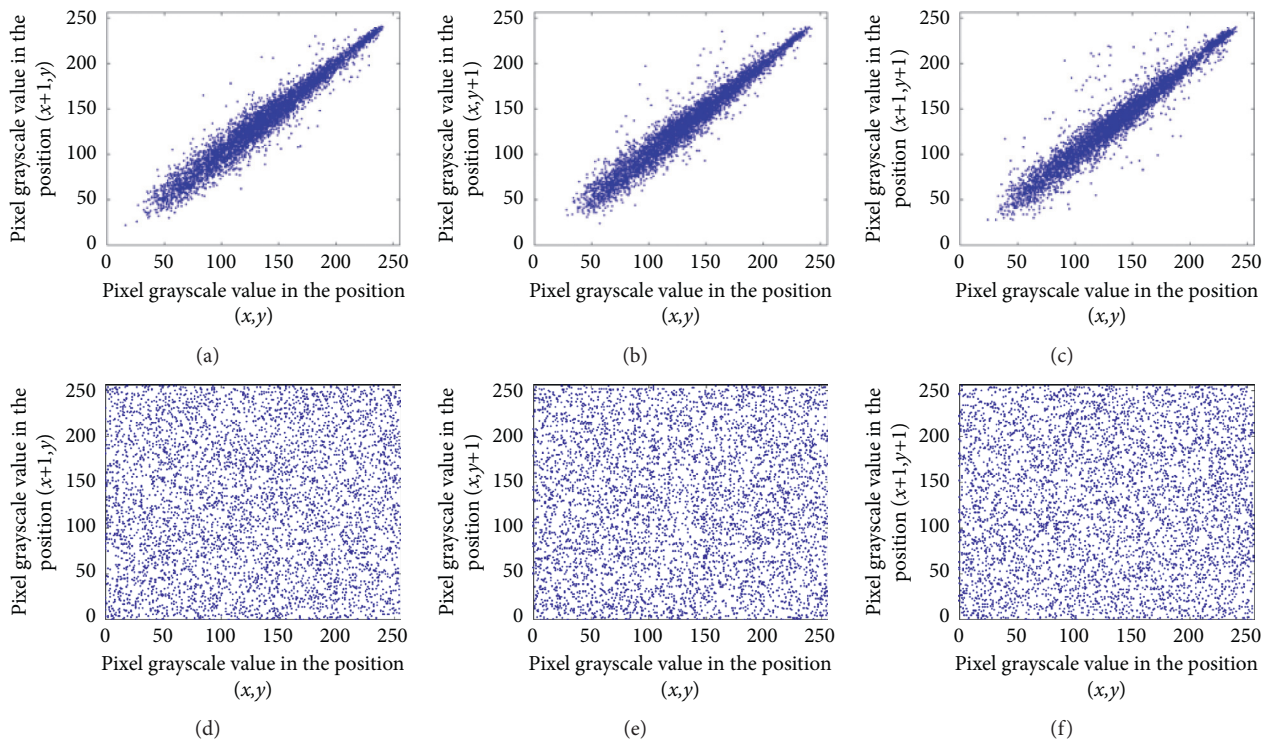


Figure 12: Pixel correlation distribution charts of elaine.512.tiff and its cipher image: (a, d) distribution charts in horizontal direction; (b, e) distribution charts in vertical direction; (c, f) distribution charts in diagonal direction.

transformations on the plain image pixels during the permutation process, so the common attack method that ignores the permutation process by the chosen plain images composed of single-value pixels cannot work. Thirdly, SIT-SR adopts a nonlinear diffusion structure; that is, it adopts the discrete logarithms based on two different generators, which makes the encryption process cannot be simplified by chosen plain images. To sum up, SIT-SR can effectively resist CPAs.

4.2.8. Encryption Efficiency. Improving encryption efficiency is one of the most important motivations to design new image encryption schemes. SIT-SR introduce DPPs and discrete logarithms, but in fact, discrete logarithms can be calculated in advance, and the calculation process of DPP is very simple, so the impact on encryption efficiency is very small. In addition, SIT-SR uses single-pixel diffusion and only performs three iterations in the encryption process. These also help to reduce the total number of primitive operations that need to be executed. Table 12 shows the average time required by SIT-SR and some other recent image encryption schemes to encrypt the 8-bit grayscale image Lena (256×256). As can be seen from Table 12, although the time complexity of each image encryption scheme is $O(M \times N)$, the scheme proposed in [13] requires the least number of primitive operations, so it has the

Table 12: Comparison of test results obtained by different encryption schemes to encrypt Lena image.

Scheme	SIT-SR	[13]	[18]	[19]	[21]	[22]
Time (s)	**0.089**	0.072	0.417	0.683	0.275	0.264
Throughput (Mbps)	**5.6180**	6.9444	1.1990	0.7321	1.8182	1.8939

The bold values here emphasize that SIT-SR has better performance than other schemes.

highest encryption efficiency, whereas for SIT-SR, it adds a certain number of primitive operations to ensure the security, but it still maintains the significant advantage of high encryption efficiency. That is, in terms of encryption efficiency, SIT-SR is significantly better than the remaining four image encryption schemes.

5. Conclusions

In order to improve the efficiency and security of image transmission, an image transmission scheme based on two chaotic maps is proposed in this paper. The proposed scheme divides the image transmission from sensor nodes to receivers into two stages and carries out a targeted design, which can better adapt to heterogeneous application environments. For image transmission between sensor nodes and sink nodes, the proposed scheme reduces the requirements for hardware resources and improves the image reconstruction quality by introducing a lightweight chaotic map. Besides, the design of dynamic perturbation improves the security of image transmission at this stage, whereas for image transmission between sink nodes and receivers, the proposed scheme improves the security and efficiency of image transmission by introducing another chaotic map with better chaotic performance and discrete logarithms. In order to verify and demonstrate the excellent performance of the proposed scheme, extensive simulation tests and theoretical analyses are carried out. These tests and analyses show that, compared with some recent schemes, the proposed scheme has higher feasibility, security, and practicability. In the future, we will extend the proposed scheme to video transmission.

Data Availability

The figure data and table data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Wei Feng conceptualized the study, was responsible for methodology and software, performed formal analysis and investigation, prepared the original draft, and was involved in funding acquisition; Jing Zhang was involved in conceptualization, funding acquisition, and supervision, validated the data, and reviewed and edited the manuscript; Zhentao Qin was responsible for methodology and software, validated the data, and reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was supported by the Science and Technology Development Center Project of Chinese Ministry of Education (no. 2018A0105), the Natural Science Key Project of Education Bureau of Sichuan Province (no. 18ZA0288), the Guiding Science and Technology Plan Project of Panzhihua City (nos. 2019ZD-G-18 and 2020ZD-S-40), and the Doctoral Research Startup Foundation of Panzhihua University (no. 2020DOC019).

References

[1] Y. Zhang, Y. Xiang, L. Y. Zhang, Y. Rong, and S. Guo, "Secure wireless communications based on compressive sensing: a survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1093–1111, 2019.

[2] Y. Zhang, Q. He, Y. Xiang et al., "Low-cost and confidentiality-preserving data acquisition for internet of multimedia things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3442–3451, 2018.

[3] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: a year in review," *Journal of Information Security and Applications*, vol. 48, Article ID 102361, 2019.

[4] Z. Niu, M. Zheng, Y. Zhang, and T. Wang, "A new asymmetrical encryption algorithm based on semitensor compressed sensing in WBANs," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 734–750, 2020.

[5] L. Wang, L. Li, J. Li, J. Li, B. B. Gupta, and X. Liu, "Compressive sensing of medical images with confidentially homomorphic aggregations," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1402–1409, 2019.

[6] L. Li, G. Wen, Z. Wang, and Y. Yang, "Efficient and secure image communication system based on compressed sensing for iot monitoring applications," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 82–95, 2020.

[7] L. Li, L. Liu, H. Peng, Y. Yang, and S. Cheng, "Flexible and secure data transmission system based on semitensor compressive sensing in wireless body area networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3212–3227, 2019.

[8] R. Zhao, Y. Zhang, X. Xiao, X. Ye, and R. Lan, "Tpe2: three-pixel exact thumbnail-preserving image encryption," *Signal Processing*, vol. 183, Article ID 108019, 2021.

[9] Y. Zhang, R. Zhao, X. Xiao, R. Lan, Z. Liu, and X. Zhang, "HF-TPE: high-fidelity thumbnail-preserving encryption," *IEEE Transactions on Circuits and Systems for Video Technology*, p. 1, 2021.

[10] A. S. Unde and P. P. Deepthi, "Design and analysis of compressive sensing-based lightweight encryption scheme for multimedia iot," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 167–171, 2020.

[11] Y. Zhang, Q. He, G. Chen, X. Zhang, and Y. Xiang, "A low-overhead, confidentiality-assured, and authenticated data

acquisition framework for iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7566–7578, 2020.

[12] B. Zhang, D. Xiao, and Y. Xiang, "Robust coding of encrypted images via 2D compressed sensing," *IEEE Transactions on Multimedia*, vol. 23, no. 99, pp. 2656–2671, 2021.

[13] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.

[14] W. Feng, Y. He, H. Li, and C. Li, "A plain-image-related chaotic image encryption algorithm based on DNA sequence operation and discrete logarithm," *IEEE Access*, vol. 7, pp. 181589–181609, 2019.

[15] H. Li, T. Li, W. Feng et al., "A novel image encryption scheme based on non-adjacent parallelable permutation and dynamic dna-level two-way diffusion," *Journal of Information Security and Applications*, vol. 61, Article ID 102844, 2021.

[16] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Processing: Image Communication*, vol. 52, pp. 6–19, 2017.

[17] X. Wang and D. Xu, "A novel image encryption scheme based on Brownian motion and PWLCM chaotic system," *Nonlinear Dynamics*, vol. 75, no. 1-2, pp. 345–353, 2014.

[18] Q. Yin and C. Wang, "A new chaotic image encryption scheme using breadth-first search and dynamic diffusion," *International Journal of Bifurcation and Chaos*, vol. 28, no. 4, Article ID 1850047, 2018.

[19] X. Wu, D. Wang, J. Kurths, and H. Kan, "A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system," *Information Sciences*, vol. 349-350, pp. 349-350, 2016.

[20] R. Zahmoul, R. Ejbali, and M. Zaied, "Image encryption based on new beta chaotic maps," *Optics and Lasers in Engineering*, vol. 96, pp. 39–49, 2017.

[21] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "An image encryption algorithm based on compound homogeneous hyper-chaotic system," *Nonlinear Dynamics*, vol. 89, no. 1, pp. 61–79, 2017.

[22] A.-V. Diaconu, "Circular inter-intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355-356, pp. 314–327, 2016.

[23] S. S. Moafimadani, Y. Chen, and C. Tang, "A new algorithm for medical color images encryption using chaotic systems," *Entropy*, vol. 21, no. 6, Article ID 577, 2019.

[24] P. Ramasamy, V. Ranganathan, S. Kadry, R. Damaševičius, and T. Blažauskas, "An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic-tent map," *Entropy*, vol. 21, no. 7, Article ID 656, 2019.

[25] S. Zhu, G. Wang, and C. Zhu, "A secure and fast image encryption scheme based on double chaotic s-boxes," *Entropy*, vol. 21, no. 8, Article ID 790, 2019.

[26] S. Zhou, P. He, and N. Kasabov, "A dynamic dna color image encryption method based on sha-512," *Entropy*, vol. 22, no. 10, Article ID 1091, 2020.

[27] L. Liu, D. Jiang, X. Wang, X. Rong, and R. Zhang, "2D logistic-adjusted-Chebyshev map for visual color image encryption," *Journal of Information Security and Applications*, vol. 60, Article ID 102854, 2021.

[28] D. Jiang, L. Liu, L. Zhu, X. Wang, X. Rong, and H. Chai, "Adaptive embedding: a novel meaningful image encryption scheme based on parallel compressive sensing and slant transform," *Signal Processing*, vol. 188, Article ID 108220, 2021.

[29] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[30] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[31] Y. Zhang, P. Wang, H. Huang, Y. Zhu, D. Xiao, and Y. Xiang, "Privacy-assured FOGCS: chaotic compressive sensing for secure industrial big image data processing in fog computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3401–3411, 2021.

[32] Y. Zhang, P. Wang, L. Fang, X. He, H. Han, and B. Chen, "Secure transmission of compressed sampling data using edge clouds," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6641–6651, 2020.

[33] L. Y. Zhang, K.-W. Wong, Y. Zhang, and J. Zhou, "Bi-level protected compressive sampling," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1720–1732, 2016.

[34] J. Wang, L. Y. Zhang, J. Chen, G. Hua, Y. Zhang, and Y. Xiang, "Compressed sensing based selective encryption with data hiding capability," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6560–6571, 2019.

[35] H. Peng, Y. Mi, L. Li, H. E. Stanley, and Y. Yang, "P-tensor product in compressed sensing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3492–3511, 2019.

[36] H. Gan, S. Xiao, T. Zhang, and F. Liu, "Bipolar measurement matrix using chaotic sequence," *Communications in Nonlinear Science and Numerical Simulation*, vol. 72, pp. 139–151, 2019.

[37] L. Li, Y. Fang, L. Liu, H. Peng, J. Kurths, and Y. Yang, "Overview of compressed sensing: sensing model, reconstruction algorithm, and its applications," *Applied Sciences*, vol. 10, no. 17, Article ID 5909, 2020.

[38] Y. Yicong Zhou, Z. Zhongyun Hua, C. Chi-Man Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 2001–2012, 2015.

[39] A. Eftekhari, M. Babaie-Zadeh, and H. Abrishami Moghaddam, "Two-dimensional random projection," *Signal Processing*, vol. 91, no. 7, pp. 1589–1603, 2011.

[40] G. Chen, D. Li, and J. Zhang, "Iterative gradient projection algorithm for two-dimensional compressive sensing sparse image reconstruction," *Signal Processing*, vol. 104, pp. 15–26, 2014.

[41] C. Li, D. Lin, B. Feng, J. Lü, and F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834–75842, 2018.

[42] W. Feng, Y. He, H. Li, and C. Li, "Cryptanalysis and improvement of the image encryption scheme based on 2D logistic-adjusted-sine map," *IEEE Access*, vol. 7, pp. 12584–12597, 2019.

[43] W. Feng and Y.-G. He, "Cryptanalysis and improvement of the hyper-chaotic image encryption scheme based on DNA encoding and scrambling," *IEEE Photonics Journal*, vol. 10, no. 6, pp. 1–15, Article ID 7909215, 2018.

[44] L. Y. Zhang, Y. Liu, F. Pareschi et al., "On the security of a class of diffusion mechanisms for image encryption," *IEEE transactions on cybernetics*, vol. 48, no. 4, pp. 1163–1175, 2018.

[45] M. Li, D. Lu, W. Wen, H. Ren, and Y. Zhang, "Cryptanalyzing a color image encryption scheme based on hybrid hyperchaotic system and cellular automata," *IEEE Access*, vol. 6, pp. 47102–47111, 2018.

[46] C. Li, D. Lin, and J. Lü, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE MultiMedia*, vol. 24, no. 3, pp. 64–71, 2017.

[47] C. Li, D. Lin, J. Lü, and F. Hao, "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE MultiMedia*, vol. 25, no. 4, pp. 46–56, 2018.

[48] M. Li, H. Fan, Y. Xiang, Y. Li, and Y. Zhang, "Cryptanalysis and improvement of a chaotic image encryption by first-order time-delay system," *IEEE MultiMedia*, vol. 25, no. 3, pp. 92–101, 2018.

[49] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 235–246, 2016.

[50] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[51] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 305–313, 2018.

[52] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, pp. 323–342, 2013.

[53] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018.

Hindawi
Complexity
Volume 2021, Article ID 2036466, 10 pages
https://doi.org/10.1155/2021/2036466

Research Article

Industrial Printing Image Defect Detection Using Multi-Edge Feature Fusion Algorithm

Bangchao Liu, Youping Chen, Jingming Xie, and Bing Chen

School of Mechanical Science and Engineering, State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Bangchao Liu; d201880275@hust.edu.cn

Received 6 August 2021; Accepted 13 September 2021; Published 4 October 2021

Academic Editor: Padmapriya Praveenkumar

Online defect detection system is a necessary technical measure and important means for large-scale industrial printing production. It is effective to reduce artificial detection fatigue and improve the accuracy and stability of industry printing line. However, the existing defect detection algorithms are mainly developed based on high-quality database and it is difficult to detect the defects on low-quality printing images. In this paper, we propose a new multi-edge feature fusion algorithm which is effective in solving this problem. Firstly, according to the characteristics of sheet-fed printing system, a new printing image database is established; compared with the existing databases, it has larger translation, deformation, and uneven illumination variation. These interferences make defect detection become more challenging. Then, SIFT feature is employed to register the database. In order to reduce the number of false detections which are caused by the position, deformation, and brightness deviation between the detected image and reference image, multi-edge feature fusion algorithm is proposed to overcome the effects of these disturbances. Lastly, the experimental results of mAP (92.65%) and recall (96.29%) verify the effectiveness of the proposed method which can effectively detect defects in low-quality printing database. The proposed research results can improve the adaptability of visual inspection system on a variety of different printing platforms. It is better to control the printing process and further reduce the number of operators.

1. Introduction

The online defect detection system based on machine vision is widely used in the field of industrial automation, such as welding defect detection [1], glass manufacturing industry [2], machine-parts processing [3], printed circuit board industry [4], textile industry [5], and printing industry [6, 7]. In printing industry, manual detection has been far from meeting the quality control requirements of modern large-scale printing production. Online defect detection system is an indispensable link to ensure the quality of printed matter [8]. Figure 1 gives an example of industrial printing line, and Figure 2 shows a typical defect detection system using machine vision for roll-to-roll printing line.

This paper is organized as follows. We present related research in Section 2, and Section 3 gives the methodology, which includes the architecture of the proposed defect detection system, image registration method, and feature extraction method. The experimental results and discussion are given in Section 4. Section 5 presents the conclusion and future works.

2. Related Research

Many defect detection methods have been proposed for roll-to-roll printing process, and these methods can effectively detect a variety of printing defects in real time. Paper [9] detected some common printing defects, such as ink drop, stripe, character loss, and color defect. Compared with the traditional method to extract the gradient edge of gray image, the proposed edge detection algorithm has better detection performance, and it can reduce the information loss of RGB three channels and make the edge extraction more accurate. Paper [10] provided an image fusion method, which used multi-channel image subtraction to segment defects. The method can update the

Figure 1: Industrial printing production equipment.

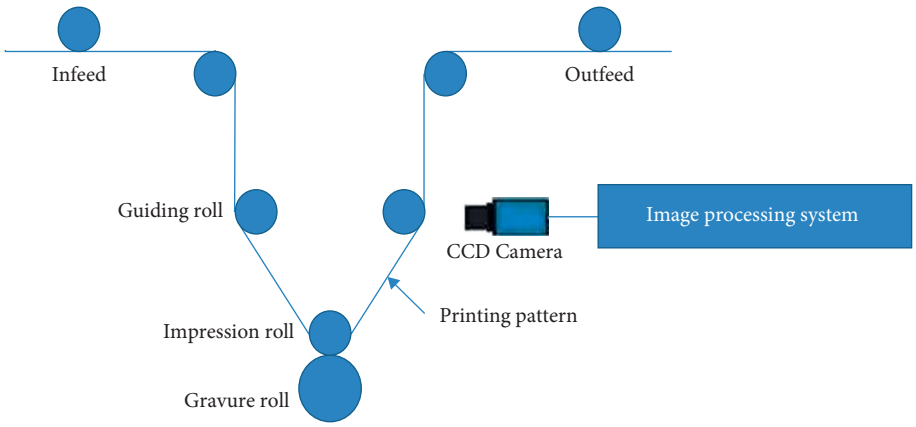


Figure 2: The diagram of defect detection system using machine vision for gravure printing line.

reference image continuously with the printing production. Next, a twice template matching algorithm was established in paper [11], which firstly matched the template and then performed differential operation on matched image to find the location of the defect. In paper [12], the authors designed a bidirectional image difference algorithm to avoid the error detection of contour artifacts. In order to better locate the first row of captured image, a fast-computational algorithm based on image projection was given, which can convert 2D image searching into 1D feature matching. Paper [13] adopted laser scanning technology to detect overfill and underfill defects on upper surface of the deposited parts in the additive manufacturing process by comparing the existing point cloud with the presliced stereolithography (STL) model. In paper [14], Chervyakov et al. utilized two modified adaptive median filtering methods of impulse noise in images. The experiment showed potential application in processing satellite and medical imagery, geophysical data, and in other areas of digital image processing. Paper [15] provided a solution to the problem of distinguishing the defects and their own characteristics in robot 3D printing. The research findings can help to detect the defects online, improve the detection accuracy, and reduce the false detection rate without being affected by its own characteristics.

The abovementioned detection methods all applied image difference to extract defective patterns. Because the defective image and standard image collected in reality often have interferences with size, rotation, deformation, and other factors, whether the two images can be well registered will directly affect the accuracy of defect detection [16]. Image registration needs to be considered from the feature space, search space, interpolation method, search strategy, similarity measurement, and other aspects [6]. In addition, these inspection systems also need to meet some other requirements: (1) high-end and expensive line scan CCD camera; (2) special high-precision mechanical installation structure and lighting mode; (3) stable and reliable feeding platform; and (4) high-quality image data acquisition. These requirements limit the application range of the detection system [17–20]. For example, it is difficult to apply the online quality inspection system in the sheet-fed platform and the low-end printing production line because the vibration and interference of the platform are too large, and it is impossible to collect qualified images. For the research issue of low-quality printing image defect detection, the research results are few.

Currently, more and more researchers pay attention to the defect detection method based on machine learning [21]. Du used the deep learning method to improve the performance of X-ray image defect detection of automotive aluminum castings [22]. In paper [23], the defect detection of railway track fastener was studied by combining image processing and deep learning. However, labeling defect regions was time consuming, and it was difficult to collect enough defect samples for artificial neural network learning, which limited the application and promotion of deep learning in the field of defect detection. Paper [24] proposed an automatic inspection system with five-plane array charge-coupled device (CCD) cameras and four LED light sources

in a closed environment. A support vector machine algorithm was adopted to classify defects based on the extracted features in candidate defect regions. In paper [25], Abul'khanov and Kazanskiy created visual and numerical tools to analyze a rough surface, through characterizing the rough surface by building its information pattern through imaging micro-roughnesses on the controlled surface and using the parameter value. Paper [26] designed a bridge cracks detection algorithm by using a modified active contour model and greedy search-based support vector machine. In paper [27], a novel cascaded autoencoder (CASAE) architecture was designed for segmenting and localizing defects. The defect regions of segmented results are classified into their specific classes via a compact convolutional neural network (CNN). Paper [28] adopted a single convolutional neural network (CNN) model that can extract effective features for defect classification without using additional feature extraction algorithms, and the proposed method can identify defect classes not seen during training by comparing the CNN features of the unseen classes with those of the trained classes. Paper [29] proposed a vision-based method using a deep architecture of convolutional neural networks (CNNs) for detecting concrete cracks without calculating the defect features. In addition, many other methods were also proposed for defect detection using machine learning, e.g., the generative adversarial networks [30] and reinforcement learning [31].

In summary, traditional defect detection methods based on image difference cannot be applied to low-quality images effectively and it is difficult to get enough defect samples to train a machine learning model for defect identification. In order to improve the adaptability of visual inspection system on a variety of different printing platforms (sheet-fed or roll-to-roll), in this paper, a new printed image database is established using a small CCD area array camera to collect images on a sheet-fed machine platform. Then, a new multi-edge feature fusion algorithm is proposed to adapt to defect detection in low-quality dataset.

3. Materials and Methods

The architecture of the proposed detection system consists of image registration, image sub-block, feature extraction, feature fusion, and feature matching, as shown in Figure 3. Initially, some basic image preprocessing methods are introduced and then we describe the detection algorithm using multi-edge feature fusion in detail. Because the variation of uneven illumination and deformation between samples can greatly affect the edge feature extraction, which will lead to false detection and missed detection, we adopt the feature fusion method to eliminate the influence of those interferences. Next, the detection evaluation criteria selected in this paper are described elaborately. Lastly, we present and analyze the experiment results.

3.1. Image Registration Using SIFT Feature. The obtained original image contains many interference factors, such as rotation and deformation [32]. The inconsistency of images needs to be corrected in advance. In this paper, SIFT feature is applied to eliminate these variations in original images, and Figure 4 shows the architecture of image registration using SIFT feature, including key points, matching images, and corrected images [33]. We cut off the redundant boundary with size of 50 pixels directly, and the image resolution is reduced from $800 * 550$ to $750 * 500$. The corrected images are as follows:

$$C_k(i, j) = R_k(i + 25, j + 25), \tag{1}$$

where C is the corrected image, i and j are the coordinates of each pixel, R stands for the registered image, and k is the image sequence number. Then, we divide the whole image into several sub-blocks with resolution of $50 * 50$, as shown in image sub-block step in Figure 3.

3.2. Image Feature Extraction and Feature Matching

3.2.1. Feature Extraction Using Canny Edge Detection Operator. The mainstream edge detection operators include Roberts, Sobel, Prewitt, and so on [34]. Robert operator is sensitive to noise. Prewitt and Sobel operators have better detection performance on the image with gradual gray level and low noise, but for the image with mixed multi-complex noise, the processing effect is not ideal. The detection effect of the Canny operator is better than that of the gradient operator, which can detect the thin edge of the image. There are four processing steps consisting of noise reduction, gradient calculation, non-maximum suppression, and double threshold filtering. We can change the edge with multiple pixel width into a single pixel wide edge and remove the weak edge to retain the strong edge. Therefore, it is necessary to select the appropriate operator to detect the edge feature according to different environmental conditions and requirements. The output edge images are as follows:

$$B_k(i, j) = \begin{cases} 0, & \text{if } Nms_k(i, j) < T_L, \\ 1, & \text{if } Nms_k(i, j) < T_H, \end{cases} \tag{2}$$

where B is the detected edge feature binary image, Nms represent gradient amplitude images after non-maximum suppression, i and j are the coordinates of each pixel, and TL and TH represent high and low thresholds. Figure 5 presents the edge detection effect of the Canny operator, and the defective part is marked with a red circle.

3.2.2. Feature Similarity Matching Using Euclidean Distance. After obtaining the edge features, similarity with feature matching is used to judge the defect. Firstly, the edge feature image is reduced to one-dimensional feature vector. The formula is as follows:

$$F_k(i) = \{B_k(1), B_k(2), B_k(3), \ldots, B_k(m \times n)\}, \tag{3}$$

where F is the one-dimensional eigenvector of edge block feature, B represents every pixel on the binary edge feature image, m and n represent the row number and column

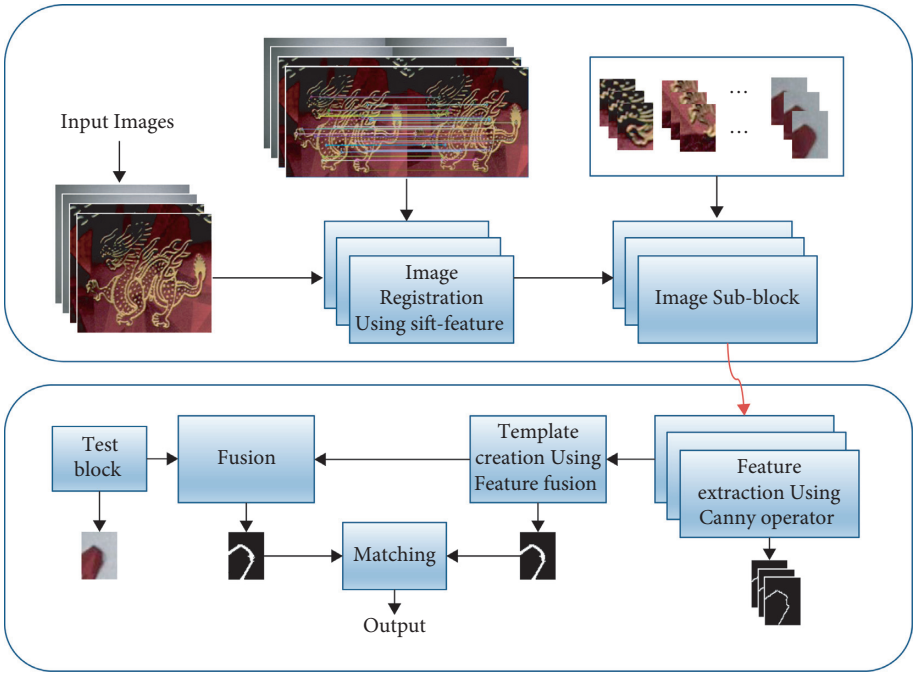


Figure 3: The architecture of the proposed defect detection system.

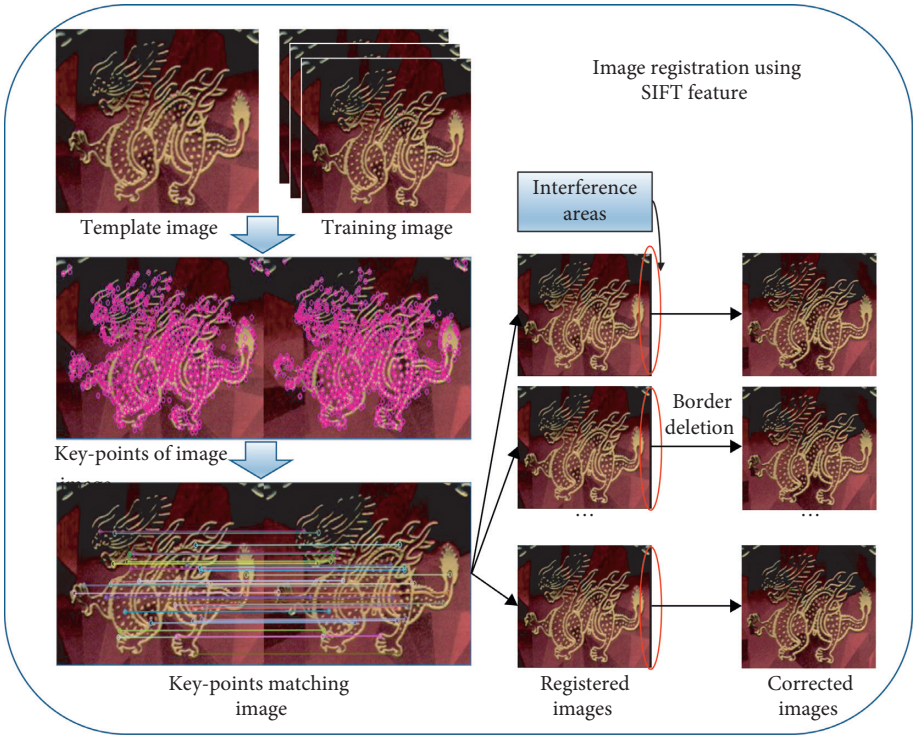
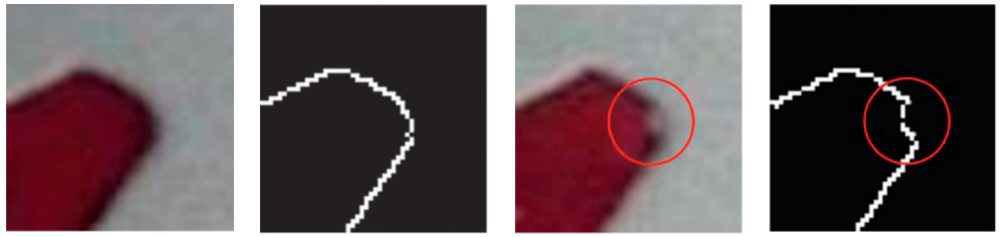


Figure 4: The architecture of image registration using SIFT feature.

number of feature image, and k is the image sequence number.

Then, Euclidean distance is used to match the similarity between two eigenvectors [17]. The formula of Euclidean distance is

$$\mathrm{dist}\left(F_{\mathrm{Template}}, F_{\mathrm{Test}}\right) = \sqrt{\sum_{i=1}^{m\times n}\left(F_{\mathrm{Template}}(i) - F_{\mathrm{Test}}(i)\right)^2}, \quad (4)$$



(a) (b) (c) (d)

Figure 5: Edge detection effect of Canny detection operator: (a) qualified pattern; (b) qualified pattern with Canny edge detection; (c) defect pattern; (d) defect pattern with Canny edge detection.

where dist is the Euclidean distance between two eigenvectors with F_{Template} and F_{Test}, i represents every element in two eigenvectors, m and n represent the row number and column number of feature image, and Template and Test represent template and test eigenvectors, respectively.

According to the definition of Euclidean distance, smaller distance between two eigenvectors means greater similarity. We define the activation function of defect judgment as follows.

$$\text{Output} = \begin{cases} 1, & \text{dist}\left(F_{\text{Template}}, F_{\text{Test}}\right) > \text{threshold}, \\ 0, & \text{other}. \end{cases} \tag{5}$$

When the matching result is greater than the threshold value, the test eigenvector is recognized as defect and 1 is output.

3.2.3. Fluctuation Analysis of Feature Matching Similarity. Firstly, we defined feature similarity matching as four types: (a) single qualified pattern and multiple qualified patterns matching; (b) single qualified pattern and multiple defect patterns matching; (c) single defect pattern and multiple qualified patterns matching; and (d) single defect pattern and multiple defect patterns matching.

Figure 6 shows the corresponding matching results. According to analysis of matching results, the four matching results show uniform fluctuation and there is no obvious similarity difference between them. It is due to reflection of pattern surface, uneven external light, and mechanical vibration. These images with inconsistent fluctuation can lead to false detections. In the next section, we will utilize feature fusion method to eliminate matching similarity fluctuation between qualified patterns.

3.2.4. Template Establishment and Defect Segmentation. In order to eliminate the interferences as much as possible, we propose a multi-template edge feature fusion algorithm to increase the accuracy of defect identification. The architecture of the proposed method model is shown in Figure 7, and the formula of feature fusion is expressed as follows:

$$\begin{aligned} F_{\text{Template}} &= F_1 | F_2 | \ldots | F_n, \\ F_{\text{Test}} &= F_{\text{Defect}} | F_{\text{Template}}, \end{aligned} \tag{6}$$

where F_{Template} represents the template feature vector, F_{Defect} stands for the defect image feature vector, and n is the sequence number of template images. Theoretically, the more the features are fused, the better the robustness of the template will be, and subsequent experiments will test the proposed template detection performance with different fusion sizes.

For visualization of defects, we extract the defect part using feature image difference, as shown in Figure 7. The defect feature is obtained using image difference between test feature and fusion feature. According to the block position number, we mark it in the original image to complete semantic segmentation and display the defect parts.

4. Results

It is difficult to collect enough defect samples in industrial production site. We ended up with 4035 complete qualified images and 135 defective images. Because most of the samples are defect free, we take 135 of them as a genuine class. Thus, the data used for test experiment contain 135 samples for each type and all the images are divided into $50 * 50$ sub-blocks. The dataset and some samples are shown in Table 1 and Figure 8, and the defective parts are marked with a red circle.

To demonstrate the effect of fusion size on the detection accuracy, some experiments under different fusion sizes were carried out, and the results are shown in Table 2. We designed seven different fusion sizes from 1 to 256. The two truth values represent genuine and defective patterns, respectively. Apparently, the detection accuracy increases significantly when the fusion size becomes larger. However, when the fusion size is larger than 160, the detection accuracy decreased. Therefore, the optimal fusion size is 160 for this case with accuracy of 95.18%, precision of 94.20%, and recall of 96.29%. When the feature fusion scale is too large or too small, it cannot achieve the ideal detection effect, such as using a single defect pattern for template feature generation.

In addition, we draw the similarity matching results with fusion size of 160 in Figure 9. The horizontal axis is the image number from 1 to 270, and the vertical axis is the Euclidean distance of feature matching from 0 to 1. From the distribution characteristics of red dots, there are only 5 missing detections with a matching result to 0 in the detection results

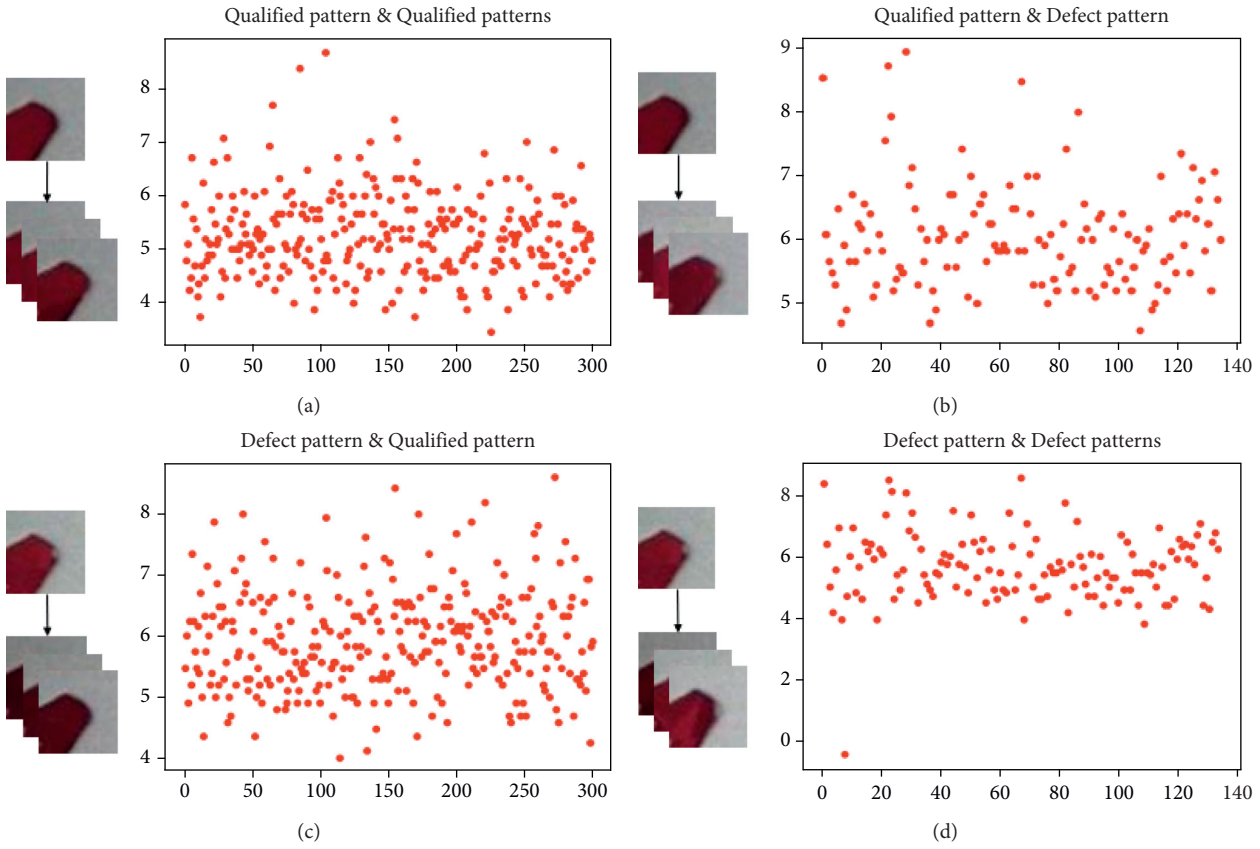


Figure 6: The similarity matching results of different pattern types. (a) The qualified pattern matches the qualified patterns. (b) The qualified pattern matches the defective patterns. (c) The defective pattern matches the qualified patterns. (d) The defective pattern matches the defective patterns. The horizontal axis represents the number of similarity matches. The vertical axis represents the similarity matching results (similarity matching score). In (a), there are 300 red points, and each point represents a matching result. The block pattern without superposition is a qualified pattern. The section superimposed by three block patterns represents the other 300 qualified patterns. In (b), there are 135 red points, and each point represents a matching result. The block pattern without superposition is a qualified pattern. The section superimposed by three block patterns represents the other 135 defective patterns. In (c), there are 300 red points, and each point represents a matching result. The block pattern without superposition is a defective pattern. The section superimposed by three block patterns represents the other 300 qualified patterns. In (d), there are 135 red points, and each point represents a matching result. The block pattern without superposition is a defective pattern. The section superimposed by three block patterns represents the other 135 defective patterns.

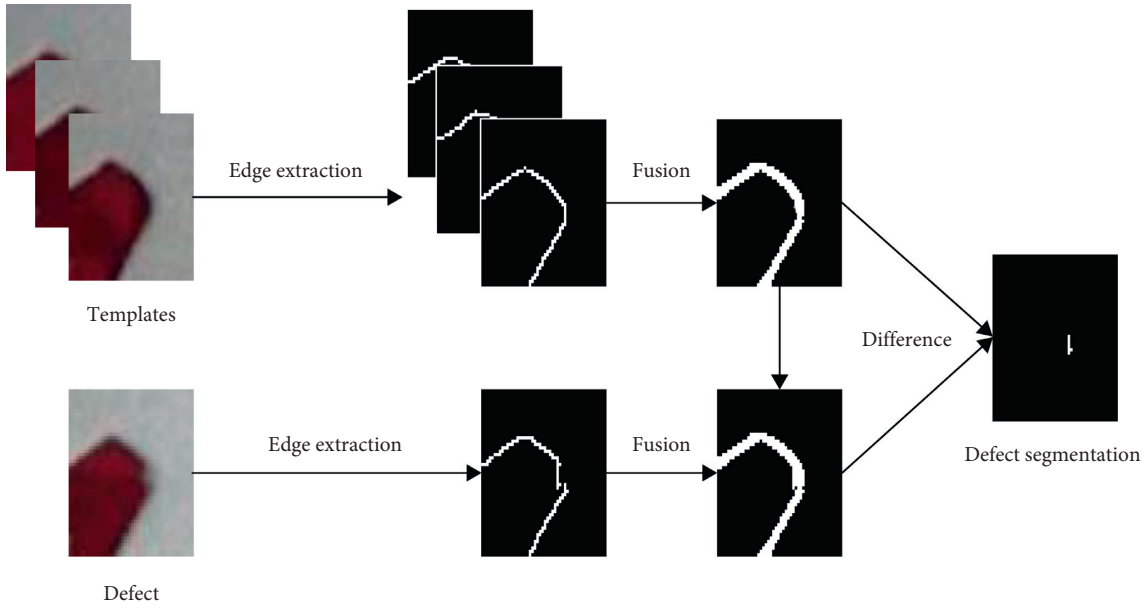


Figure 7: The architecture of the proposed feature fusion and defect segmentation model.

Table 1: The details of dataset for experiment.

Types	Complete	Defect	Total
Testing set	135	135	270
Original set	4035	135	4170

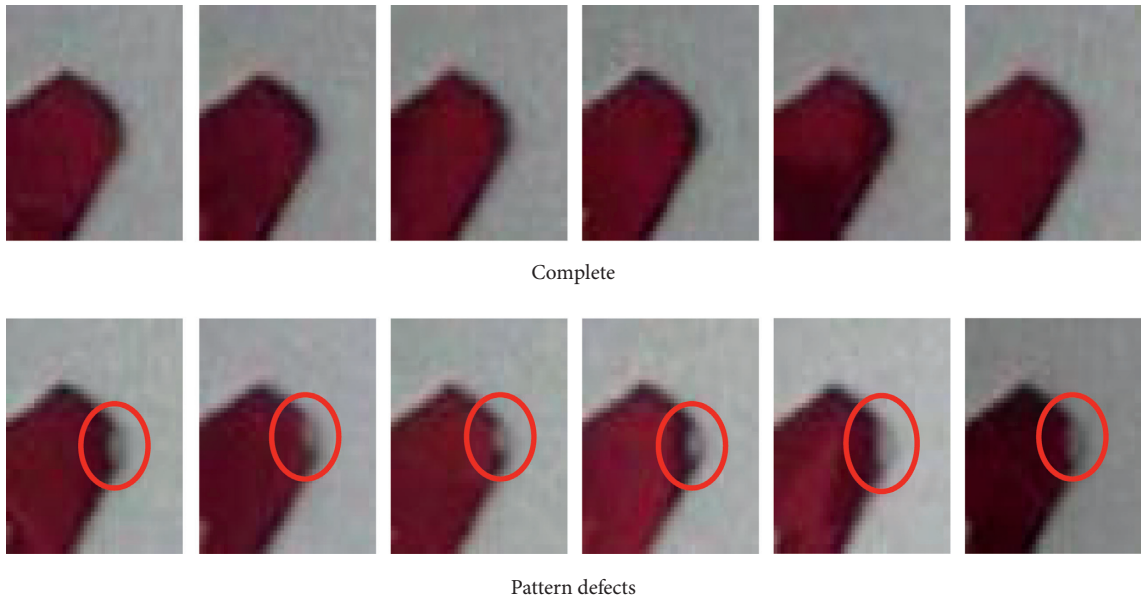


Figure 8: Dataset of some printing images used in this paper.

Table 2: The confusion matrix and evaluation result of defect detection with different fusion sizes.

Confusion matrix, accuracy, precision, and recall with different fusion sizes						
Fusion size	Truth value	Predicted value		Accuracy (%)	Precision	Recall
		Pattern	Defect			
1	Pattern	0	135	50.00	1	0
	Defect	0	135		50.00%	1
16	Pattern	48	87	67.77	1	35.55%
	Defect	0	135		60.81%	1
64	Pattern	110	25	88.88	95.65%	81.48%
	Defect	5	130		83.87%	96.29%
128	Pattern	116	19	91.11	95.86%	85.92%
	Defect	5	130		87.24%	96.29%
160	Pattern	127	8	95.18	**96.21%**	**94.07%**
	Defect	5	130		**94.20%**	**96.29%**
192	Pattern	127	8	84.81	79.37%	94.07%
	Defect	33	102		92.72%	75.55%
256	Pattern	127	8	81.11	74.70%	94.07%
	Defect	43	92		92.00%	68.14%

of the first 135 defects on horizontal axis. On the contrary, 8 out of the last 135 qualified images are detected with matching distance not equal to 0.

Then, we select different thresholds to draw the P-R curve, as shown in Figure 10. The results are the same as shown in Table 2; the detection performance first increases to the green curve with fusion size 160 and then decreases. The front four curves are in a state of underfitting with insufficient amount of feature information learning and the last two curves are in a state of overfitting state with too much redundant information learning. Therefore, too much or too little feature fusion cannot achieve ideal recognition effect. In addition, we employ average precision (AP) and mean average precision (mAP) to evaluate the performance of the defect detection method. When fusion size is 160, the mAP and recall are 92.65% and 96.29%, respectively, as shown in Table 3.

Lastly, we test detection performance of the proposed method in extended dataset with 4035 patterns and 135 defects. Table 4 shows the experimental results in asymmetric dataset. The precision of small-sample dataset

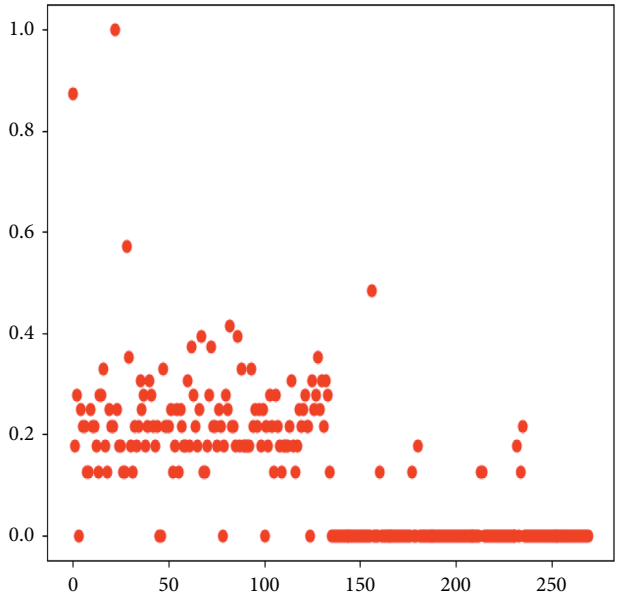


Figure 9: Similarity matching results with fusion size of 160.

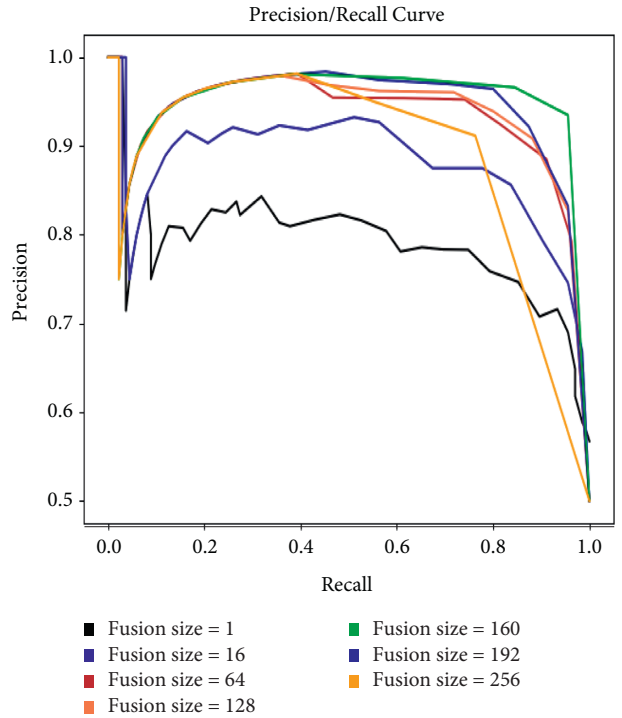


Figure 10: The P-R curve with different fusion sizes.

Table 3: mAP and recall of defect detection.

Fusion size	Type	mAP	Recall
160	Defects	0.9265	0.9629

is only 31.47%, while it is up to 99.86% in large-sample dataset.

To demonstrate the effectiveness of the method proposed in this paper, a comparison is made for the other detection methods including non-fusion registration difference and convolutional neural network (CNN). The result is shown in Table 5. The method proposed in this paper achieves an accuracy of 93.09% which outperforms other methods. The reason for this result is that the low-quality images are not suitable for the traditional differential detection algorithm

Table 4: Defect detection result using extended dataset.

Confusion matrix, accuracy, precision, and recall with extended dataset						
Fusion size	Truth value	Predicted value		Accuracy (%)	Precision (%)	Recall (%)
		Pattern	Defect			
160	Pattern	3752	283	93.09	99.86	92.98
	Defect	5	130		31.47	96.29

Table 5: Detection accuracy of different methods.

Method of detection	Accuracy (%)
Non-fusion registration difference	86.33
CNN	90.07
This paper	93.09

and it is different to provide sufficient defect training data for convolutional neural network, which leads to misclassification. In this paper, a new multi-edge feature fusion algorithm is used to recognize printing defects in low-quality datasets, which achieves a higher precision for the industrial printing image defect detection.

5. Conclusions

In this work, a new defect detection method using multi-edge feature fusion is proposed to improve the detection accuracy of low-quality printing images. The specific contributions are as follows:

(1) We set up a new and more challenging print image dataset which consists of 4170 images and has more rotation, deformation, and uneven illumination changes, compared with the existing printing database.

(2) The proposed multi-edge feature fusion algorithm can effectively distinguish pattern defects and interference changes.

(3) Different feature fusion sizes will greatly affect the detection accuracy, and we also found that for all fusion scales, an optimal value exists for the detection accuracy; too large or too small amount of fusion information will reduce the overall detection performance of the system.

The current detection system mainly solves the problem of fine edge defect detection of low-quality printing image, while in further work, the detection system should be promoted to identify more types of defects. In addition, most samples of industrial printing products are qualified and authentic, which leads to unbalanced data types. In future work, how to collect enough defective images in industrial production field and how to use machine learning method to detect defective patterns need to be further studied.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. Sassi, P. Tripicchio, and C. A. Avizzano, "A smart monitoring system for automatic welding defect detection," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9641–9650, 2019.

[2] D. Jin, S. Xu, L. Tong, L. Wu, and S. Liu, "End image defect detection of float glass based on faster region-based convolutional neural network," *Traitement du Signal*, vol. 37, no. 5, pp. 807–813, 2020.

[3] J. Lin, D. Wang, H. Tian, and Z. Liu, "Surface defect detection of machined parts based on machining texture direction," *Measurement Science and Technology*, vol. 32, no. 2, Article ID 025204, 2020.

[4] A.-A. I. M. Hassanin, F. E. Abd El-Samie, and G. M. El Banby, "A real-time approach for automatic defect detection from PCBs based on SURF features and morphological operations," *Multimedia Tools and Applications*, vol. 78, no. 24, pp. 34437–34457, 2019.

[5] K. Hanbay, M. F. Talu, Ö. F. Özgüven, and mer Faruk, "Fabric defect detection systems and methods-A systematic literature review," *Optik*, vol. 127, no. 24, pp. 11960–11973, 2016.

[6] Y. Chen, P. He, M. Gao, and E. Zhang, "Automatic feature region searching algorithm for image registration in printing defect inspection systems," *Applied Sciences*, vol. 9, no. 22, p. 4838, 2019.

[7] W. Yangping, X. Shaowei, X. Shaowei, Z. Zhengping, S. Yue, and Z. Zhenghai, "Real-time defect detection method for printed images based on grayscale and gradient differences," *Journal of Engineering Science and Technology Review*, vol. 11, no. 1, pp. 180–188, 2018.

[8] E. Zhang, Y. Chen, M. Gao, J. Duan, and C. Jing, "Automatic defect detection for web offset printing based on machine vision," *Applied Sciences*, vol. 9, no. 17, p. 3598, 2019.

[9] N. G. Shankar, N. Ravi, and Z. W. Zhong, "A real-time print-defect detection system for web offset printing," *Measurement*, vol. 42, no. 5, pp. 645–652, 2009.

[10] X. Peng, Y. Chen, and J. Xie, "An intelligent online presswork defect detection method and system," in *Proceedings of the International Conference on Information Technology & Computer Science*, June 2010.

[11] B. Ma, W. Zhu, Y. Wang, H. Wu, and Y. Yang, "The defect detection of personalized print based on template matching," in *Proceedings of the IEEE International Conference on Unmanned Systems (ICUS)*, pp. 266–271, Miami, FL USA, June 2017.

[12] Y. XiM and W. ShuanH, "A rapid defect detecting algorithm for printed matter on the assembly line," in *Proceedings of the International Conference on Systems and Informatics (ICSAI)*, pp. 1842–1845, IEEE, Shandong, China, May 2012.

[13] W. Lin, H. Shen, J. Fu, and S. Wu, "Online quality monitoring in material extrusion additive manufacturing processes based on laser scanning technology," *Precision Engineering*, vol. 60, pp. 76–84, 2019.

[14] N. I. Chervyakov, P. A. Lyakhov, P. A. Lyakhov, and A. R. Orazaev, "New methods of adaptive median filtering of impulse noise in images," *Computer Optics*, vol. 42, no. 4, pp. 667–678, 2018.

[15] H. Shen, W. Du, W. Sun, Y. Xu, and J. Fu, "Visual detection of surface defects based on self-feature comparison in robot 3-D printing," *Applied Sciences*, vol. 10, no. 1, p. 235, 2019.

[16] Y. Y. Guan and Y. C. Ye, "Printing defects detection based on two-times difference image method," *Applied Mechanics and Materials*, vol. 340, pp. 512–516, 2013.

[17] L. Zhang, K. Xie, and T. Li, "Based on line scan CCD print image detection system," in *Proceedings of the Mippr: Pattern Recognition & Computer Vision. International Society for Optics and Photonics*, 2015.

[18] K. Whisler and J. Mauro, "Defect detection, quality control, and efficiency through vision inspection systems," *Gatfworld*, vol. 5, no. 5, pp. 30–32, 2013.

[19] T.-M. Lee, J.-H. Noh, C. H. Kim, J. Jo, and D.-S. Kim, "Development of a gravure offset printing system for the printing electrodes of flat panel display," *Thin Solid Films*, vol. 518, no. 12, pp. 3355–3359, 2010.

[20] C. Englund and A. Verikas, "Ink feed control in a web-fed offset printing press," *International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9, pp. 919–930, 2008.

[21] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, 2020.

[22] W. Du, H. Shen, J. Fu et al., "Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning," *NDT International*, vol. 107, no. Oct., pp. 1–12, 2019.

[23] X. Wei, Z. Yang, Y. Liu, D. Wei, L. Jia, and Y. Li, "Railway track fastener defect detection based on image processing and deep learning techniques: a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 80, no. APR, pp. 66–81, 2019.

[24] Q. Zhou, R. Chen, B. Huang, C. Liu, J. Yu, and X. Yu, "An automatic surface defect inspection system for automobiles using machine vision methods," *Sensors*, vol. 19, no. 3, p. 644, 2019.

[25] S. R. Abul'khanov and N. L. Kazanskiy, "Information pattern in imaging of a rough surface," *IOP Conference Series: Materials Science and Engineering*, vol. 302, Article ID 012068, 2018.

[26] Z. Qu, L. Bai, S.-Q. An, F.-R. Ju, and L. Liu, "Lining seam elimination algorithm and surface crack detection in concrete tunnel lining," *Journal of Electronic Imaging*, vol. 25, no. 6, Article ID 063004, 2016.

[27] T. Xian, D. Zhang, and W. Ma, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Applied Sciences-Basel*, vol. 8, no. 9, 2018.

[28] S. Cheon, H. Lee, and O. K. Chang, "Convolutional neural network for wafer surface defect classification and the detection of unknown defect class," *IEEE Transactions on Semiconductor Manufacturing*, no. 99, p. 1, 2019.

[29] Y. J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, pp. 361–378, 2017.

[30] Y. Gao, L. Gao, and X. Li, "A generative adversarial network based deep learning method for low-quality defect image reconstruction and recognition," *IEEE Transactions on Industrial Informatics*, no. 99, p. 1, 2020.

[31] F. Zeng, X. Cai, and S. S. Ge, "Low-shot wall defect detection for autonomous decoration robots using deep reinforcement learning," *Journal of Robotics*, vol. 2020, Article ID 8866406, 7 pages, 2020.

[32] B. P. Jesper, N. Kamal, and B. M. Thomas, "Quality inspection of printed texts," in *Proceedings of the 23rd International Conference on Systems, Signals and Image Processing*, pp. 1–4, Bratislava, Slovakia, June 2016.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] G. T. Shrivakshan and C. Chandrasekar, "A comparison of various edge detection techniques used in image processing," *International Journal of Computer Science Issues*, vol. 9, no. 5, pp. 269–276, 2012.

Hindawi
Complexity
Volume 2021, Article ID 2594965, 11 pages
https://doi.org/10.1155/2021/2594965

Research Article

A Novel Megastable Oscillator with a Strange Structure of Coexisting Attractors: Design, Analysis, and FPGA Implementation

Kui Zhang,[1] M. D. Vijayakumar,[2] Sajjad Shaukat Jamal,[3] Hayder Natiq,[4] Karthikeyan Rajagopal,[5] Sajad Jafari,[6,7] and Iqtadar Hussain[8]

[1]*School of Electronic Engineering, Changzhou College of Information Technology, Changzhou 213164, China*
[2]*Center for Materials Research, Chennai Institute of Technology, Chennai, India*
[3]*Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia*
[4]*Information Technology Collage, Imam Ja'afar Al-Sadiq University, Baghdad 10001, Iraq*
[5]*Center for Nonlinear Systems, Chennai Institute of Technology, Chennai 600069, Tamilnadu, India*
[6]*Health Technology Research Institute, Amirkabir University of Technology, 424 Hafez Ave., Tehran 15875-4413, Iran*
[7]*Department of Biomedical Engineering, Amirkabir University of Technology, 424 Hafez Ave., Tehran 15875-4413, Iran*
[8]*Department of Mathematics, Statistics and Physics, Qatar University, Doha 2713, Qatar*

Correspondence should be addressed to Karthikeyan Rajagopal; rkarthiekeyan@gmail.com

Received 7 June 2021; Accepted 19 August 2021; Published 27 August 2021

Academic Editor: Ahmed A. Abd El-Latif

Megastable chaotic systems are somehow the newest in the family of special chaotic systems. In this paper, a new megastable two-dimensional system is proposed. In this system, coexisting attractors are in some islands, interestingly covered by megalimit cycles. The introduced two-dimensional system has no defined equilibrium point. However, it seems that the origin plays the role of an unstable equilibrium point. Therefore, the attractors are determined as hidden attractors. Adding a forcing term to the system, we can obtain chaotic solutions and coexisting strange attractors. Moreover, the effect of three different values of the forcing term's amplitude is studied. The dynamical properties of the designed system are investigated using attractor plots, bifurcation diagrams, and Lyapunov Exponents diagram. Phase portraits of the novel megastable oscillator are presented by FPGA design. Xilinx system generator block diagrams of the proposed system and trigonometric functions are also presented.

1. Introduction

Finding new special chaotic systems or, more specifically, new systems with special and unique dynamical characteristics has been an active area of research since about 30 years ago. First, Sprott has introduced some elegant quadratic three-dimensional chaotic systems [1]. Then, people have tried to find the simplest cases of special chaotic systems [2]. For example, chaotic systems with many wings have been designed [3], simplest jerk systems have been introduced [4], elegant hyperchaotic systems have been found [5, 6], circulant chaotic systems have been constructed [7], and symmetric chaotic flows have been investigated [8, 9].

One crucial point about dynamical systems is the role of equilibria in them. It was believed that the strange attractors and unstable equilibrium points have a strong relationship. More particularly, unstable equilibrium points were supposed to be the clue for strange attractors. However, finding dissipative chaotic systems with no equilibria was an exciting discovery which challenged that confidence [10]. Also, chaotic systems with stable equilibria changed many conventional beliefs about the reason for the creation of strange attractors [11]. Systems

with lines, curves, and surfaces of equilibria came one after another and shed more light on many unknown points in the field. Nevertheless, calculating system's equilibrium points is the first basic step of analyzing its dynamics.

Multistability is an important phenomenon in dynamical systems [12], which is a kind of double-edged sword feature. While it can cause unwanted shifts in a system's dynamic, it can provide extra flexibility, e.g., for the control aims. Sometimes the number of coexisting attractors in a multistable system becomes infinite. In such a scenario, if those infinite attractors are uncountable, the system is called extreme multistable [13–15]. Initial conditions play the role of bifurcation parameters in such systems. However, when those infinite attractors are countable, the system is called megastable. The term "megastable" was first used in [16]. Megastable chaotic systems are somehow the newest in the family of special chaotic systems [16]. In summary, the main difference between a megastable system and an extreme multistable system is in the countability of the system's coexisting attractors. In both terms, the number of coexisting attractors is limitless. Many interesting configurations of coexisting attractors have been reported in megastable systems [17].

Hidden and self-excited are types of attractors. Many research studies have focused on categorizing dynamical attractors based on them [18–20]. A self-excited attractor can be detected easily by observing an unstable equilibrium point in the attractor's basin of attraction. However, an attractor with no equilibrium point inside its basin of attraction is called hidden [21].

The analysis of the dynamics of a dynamical system needs some powerful tools to provide primary information about the system behaviors in different conditions. In this way, obtaining the bifurcation diagram is considered as one of the primary steps of analyzing system's dynamics. Another popular tool for analyzing the dynamics of a system is the Lyapunov Exponents spectrum (LE diagram). LE is simply a quantitative measure that can prove the presence of chaos in a dynamical system [2].

Chaotic systems have many engineering applications. They can be used in image encryption [22], communication [23], circuits [24], robots [25], and so on [26]. Field programmable gate array (FPGA) implementation of nonlinear systems plays a vital role in realizing a system using targeted hardware. In fact, FPGAs are a kind of chips or gate arrays that are easy to program. Engineering applications of FPGA are wired and wireless communication, industrial and medical systems, military, and aerospace. FPGAs are cost-effective depending on their families, such as Spartan, Kintex, and Virtex. Many researchers have shown interest in FPGA implementation of chaotic systems. They have performed the software-hardware interface by implementing chaotic systems in FPGA [27].

In this paper, a new two-dimensional megastable system is proposed. The sections of this paper are arranged as follows. The new proposed two-dimensional system is introduced in Section 2. Moreover, the dynamical properties are explained in that section. Next, Section 3 describes the FPGA implementation of the proposed system. The conclusion of the paper is presented in Section 4.

2. A New Megastable Chaotic Oscillator

Consider System (1), which is a two-dimensional nonlinear autonomous oscillator,

$$\begin{aligned} \dot{x} &= -0.1y + x\frac{\cos(r)}{r}, \\ \dot{y} &= \sin(0.1x) + y\frac{\cos(r)}{r}, \\ r &= \sqrt{x^2 + y^2}. \end{aligned} \tag{1}$$

System (1) is symmetric around the origin because the equations are invariant under the transformation $(x, y) \longrightarrow (-x, -y)$.

System (1) has no equilibrium points since no points can be found to solve the equations $-0.1y + x(\cos(r)/r) = 0$ and $\sin(0.1x) + y(\cos(r)/r) = 0$. However, the origin $(0, 0)$ mimics an unstable equilibrium. This system is megastable since it has infinite countable coexisting attractors (here, limit cycles).

Figure 1 is a plot of coexisting limit cycles in System (1) resulted from random initial conditions distributed around the x-axis. The formation of these attractors is noticeable. We can see islands of attractors consisting of 3, 4, and 5 limit cycles. Surprisingly 11 islands are enclosed by a huge limit cycle. Due to the attractors' isolated configuration, the phrase "islands of attractors" can be used for such systems attractors. Figure 1 includes both transient and final states of the trajectories to show the areas of islands more significantly.

By introducing a periodic external force in the first equation of System (1), the following forced oscillator is achieved:

$$\begin{aligned} \dot{x} &= -0.1y + x\frac{\cos(r)}{r} + A\,\sin(\omega t), \\ \dot{y} &= \sin(0.1x) + y\frac{\cos(r)}{r}, \\ r &= \sqrt{x^2 + y^2}. \end{aligned} \tag{2}$$

It is primarily desired to find chaos in System (2). Many sets of (A, ω) may result in chaos. By trial and error, $\omega = 0.6$ is chosen, and A is considered as the bifurcation parameter. However, an infinite number of coexisting attractors are detected. Each attractor can go through different dynamical regimes during the change in the bifurcation parameter. Thus, two attractors (one around the origin and the other around the point $(60, 0)$) are selected, and their occurring bifurcations are tracked to show such a difference.

Figure 2 shows the bifurcation diagram and LEs diagram versus A for the nearest attractor around the origin resulted from the constant initial conditions $(0.1, 0)$. It is seen that the dynamical solution starts from an attracting torus (one negative and two zero LEs). After observing limit cycles (two negative and one zero LEs), chaos occurs (one positive LE). Then, the dynamic alternates between chaos and limit cycles.

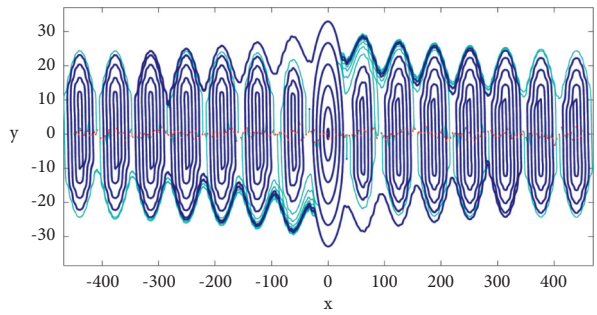


Figure 1: Coexisting limit cycles in System (1) resulted from random initial conditions distributed around the x-axis. Islands of attractors can be observed, consisting of 3, 4, and 5 limit cycles. Surprisingly 11 islands are enclosed by a huge limit cycle. The transients are shown in cyan, and steady states are shown in dark blue.

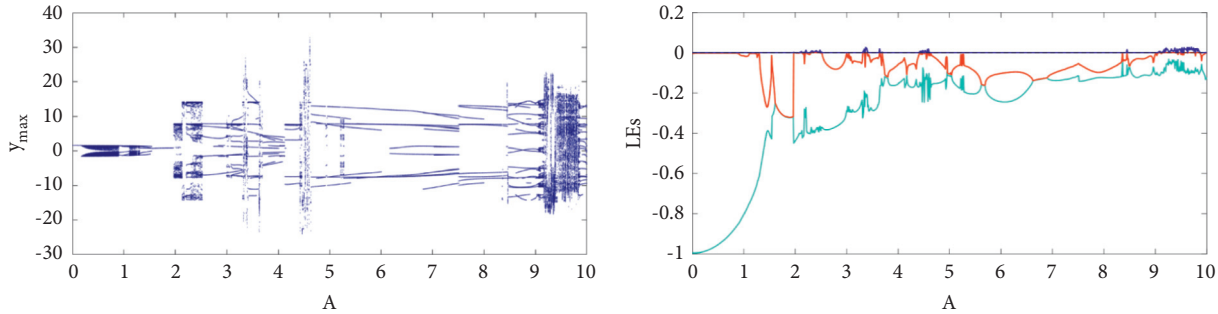


Figure 2: (a) Local maximum values of "y" time-series versus A in System (2) when $\omega = 0.6$. The initial conditions are $(0.1, 0)$ without reinitiating. Thus, the diagram is only related to the first (inner) limit cycle around the origin. (b) Corresponding Lyapunov exponent diagram. The dynamic of the system starts from an attracting torus, then limit cycles observed, and finally strange attractors appear.

Figure 3 shows the bifurcation diagram and LEs diagram versus A for the nearest attractor around the point $(60, 0)$ resulted from the constant initial conditions $(60, 0)$. It is seen that the dynamical solutions are different from the previous attractor. It starts from a limit cycle and continues with it, encountering narrow areas of chaotic attractor. It occasionally has an attracting torus in larger values of the parameter.

It should be mentioned that the LEs represented in Figures 2 and 3 are plotted and calculated using the Wolf algorithm [28] with the run time of 2000. Moreover, these two bifurcation diagrams and LEs diagrams can help comprehend the system behaviors in two different initial conditions. Furthermore, the local maxima of the time-series of variable y (y_{max}, which are the peaks of the time-series of y) are considered for plotting the bifurcation diagrams of the proposed system.

Figure 4 shows coexisting attractors for different values of the amplitude of the forcing term. While the system can have different types of attractors (limit cycle, torus, and strange attractor) simultaneously, increasing the amplitude makes them become closer and even overlap with each other.

3. FPGA Implementation of Novel Megastable Oscillator

FPGAs are gate arrays that are programmable, and they can be designed to meet a special need. FPGAs are also cost-efficient, and they are simple to design, implement, and fast prototyping. Some of the recent pieces of the literature on FPGA design had attracted many researchers, such as variable-order fractional operator [29], hardware implementation of the multistable chaotic jerk system [30], FPGA implementation of self-excited and hidden chaotic systems [31], the discrete memristor chaotic system realized using hardware [32], and digital implementation of the memristive chaotic circuit [33]. Development of the nonlinear system on an FPGA using VHDL or VERILOG hardware description language is very work-intensive. It is easy to design the system using the Xilinx system generator rather than writing test benches for the VHDL or Verilog HDL programming. In a Simulink library browser, a separate Xilinx block set toolbox is readily available to design the system in the Xilinx system generator platform. Simulink diagrams of Systems (1) and (2) are shown in Figures 5 and 6 using Xilinx system generator software. Basic blocks such as adder, subtractor, multiplier, divider, constant multiplier, and square root are used to design the proposed system in FPGA. All Xilinx block sets are different from MATLAB Simulink blocks with the Xilinx logo in them. Additional blocks are created to represent trigonometric functions present in the proposed system. By applying the Taylor series (equations (3) and (4)), trigonometric functions are implemented using the readily available (XSG) Xilinx System Generator block sets, which is shown in Figure 7 (sine function) and Figure 8 (cosine function). All these blocks used to design the proposed

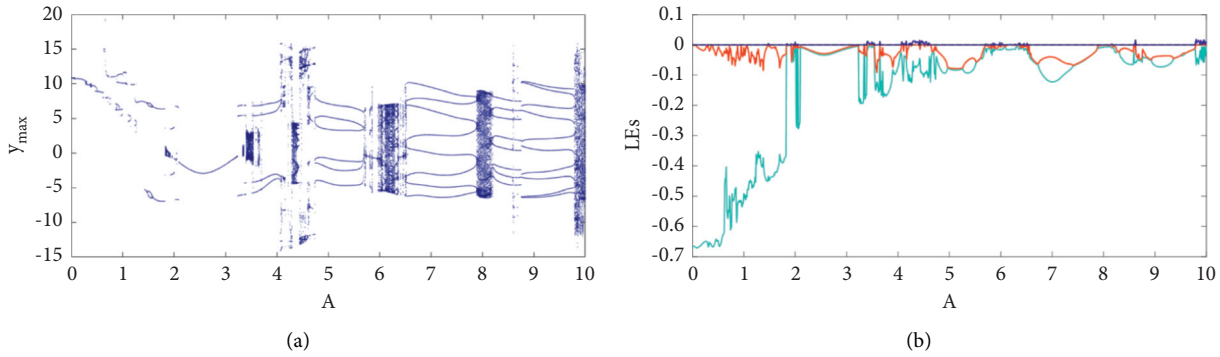


FIGURE 3: (a) Local maximum values of "y" time-series versus A in System (2) when $\omega = 0.6$. The initial conditions are $(60, 0)$ without reinitiating. Thus, the diagram is only related to the inner limit cycle around the point $(60, 0)$. (b) Corresponding Lyapunov exponent diagram. The dynamic of the system starts from a limit cycle, and then narrow areas of chaos are observed.

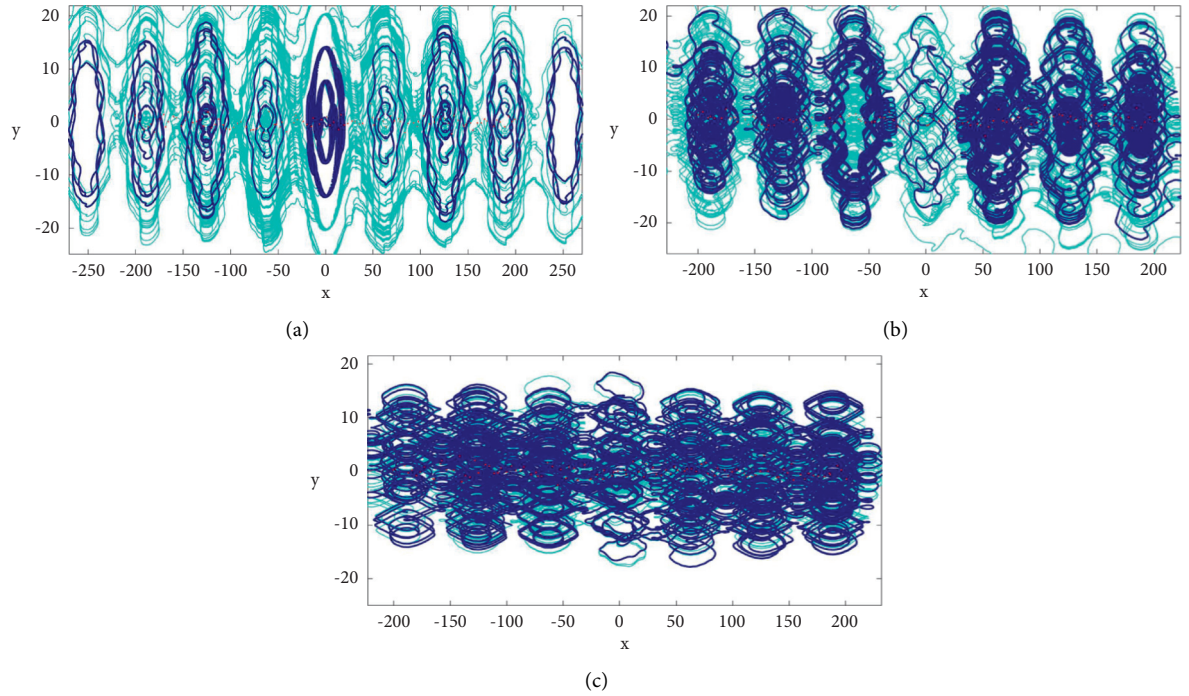


FIGURE 4: Coexisting attractors in System (2) resulted from random initial conditions distributed around the x-axis. The transients are shown in cyan, and steady states are shown in dark blue. The frequency of the forcing term is $\omega = 0.6$, and the amplitude is (a) $A = 2$, (b) $A = 4.3$, and (c) $A = 9.7$. Increasing the amplitude of the forcing term leads to more overlapped islands.

system are configured according to the IEEE754 standard, and the step size $h = 0.01$ is set. Integrator blocks of the state equations are designed using Forward Euler's Method, and the mathematical equation to design integrators is expressed in equation (5). A set of discretized system equations are stated in equations (6) and (7).

$$\sin(r) = -\frac{r^3}{3!} + \frac{r^5}{5!} - \frac{r^7}{7!} + \frac{r^9}{9!} - \frac{r^{11}}{11!} + \frac{r^{13}}{13!} - \cdots, \tag{3}$$

$$\cos(r) = 1 - \frac{r^2}{2!} + \frac{r^4}{4!} - \frac{r^6}{6!} + \frac{r^8}{8!} - \frac{r^{10}}{10!} + \frac{r^{12}}{12!} - \cdots, \tag{4}$$

$$k_{n+1} = k_n + hf(k_{n-1}), \tag{5}$$

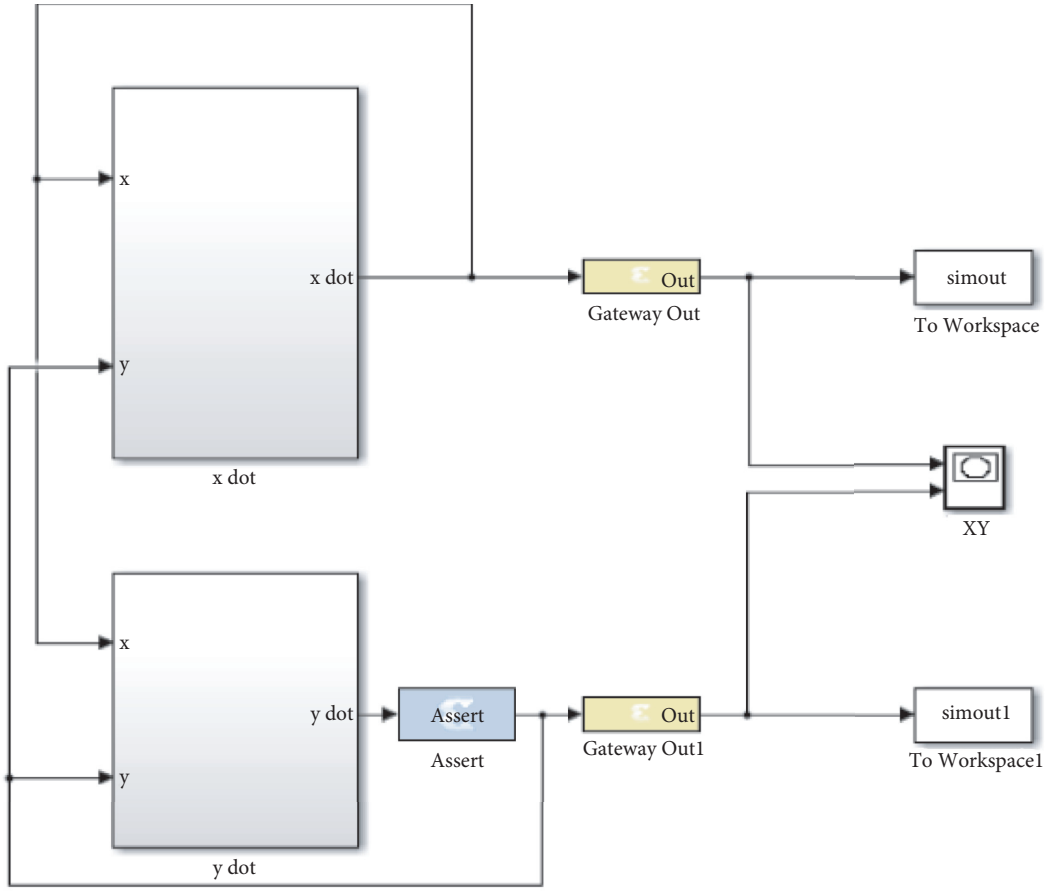


Figure 5: Xilinx system generator (XSG) Simulink diagram of System (1). It is another representation of the proposed system in autonomous form.

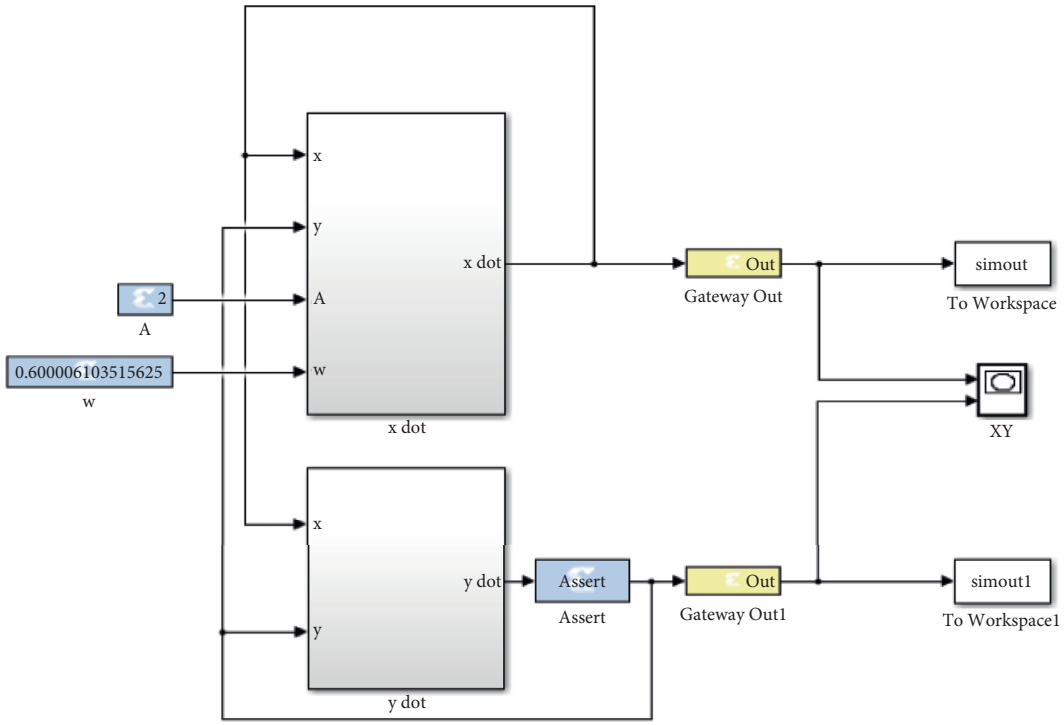


Figure 6: Xilinx system generator (XSG) Simulink diagram of System (2) with $\omega = 0.6$ and amplitude $A = 2$. It is another representation of the proposed system in nonautonomous form.

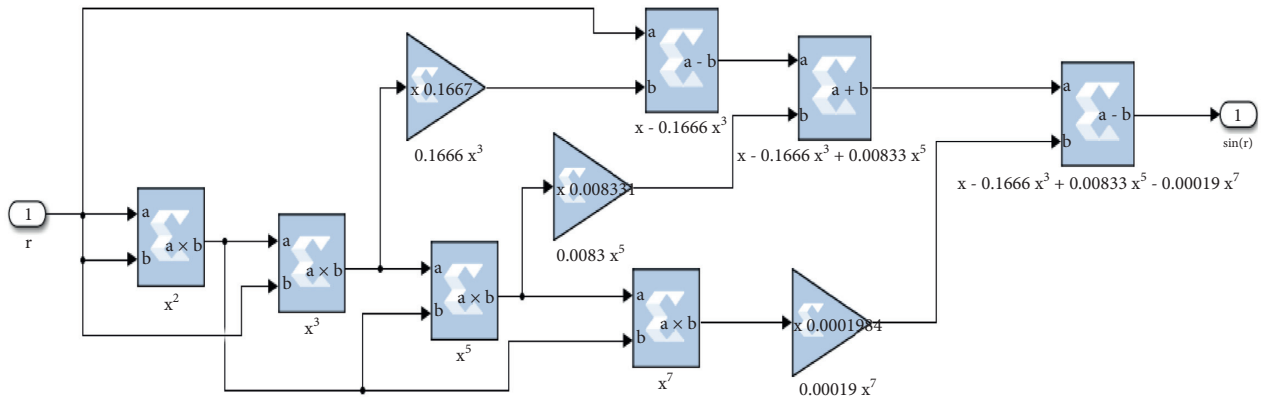


Figure 7: Taylor series of sine function implemented using (XSG) Xilinx system generator toolbox. It is discretized implementation of sine function based on equation (3).

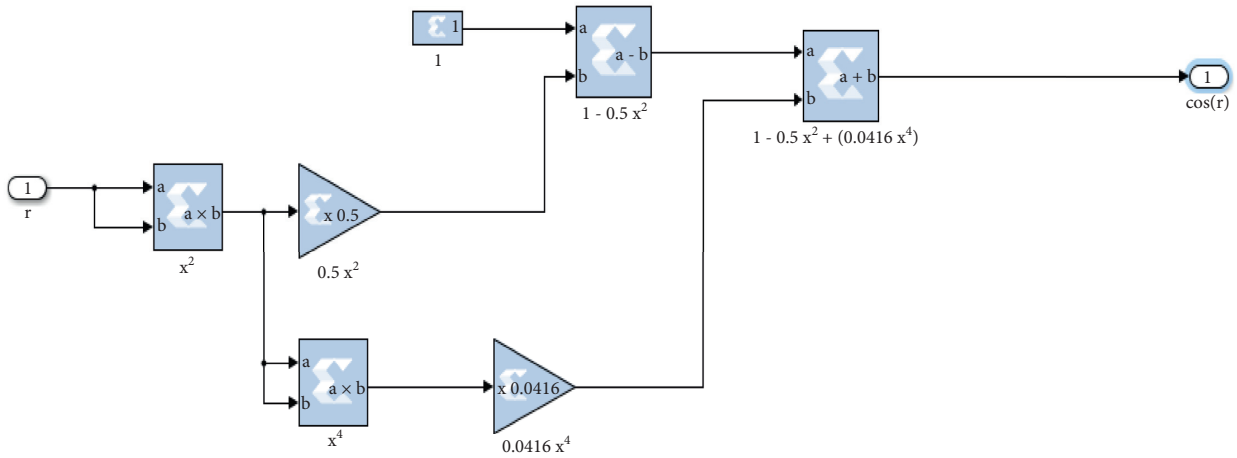


Figure 8: Taylor series of cosine function implemented using (XSG) Xilinx system generator toolbox. It is discretized implementation of cosine function based on equation (4).

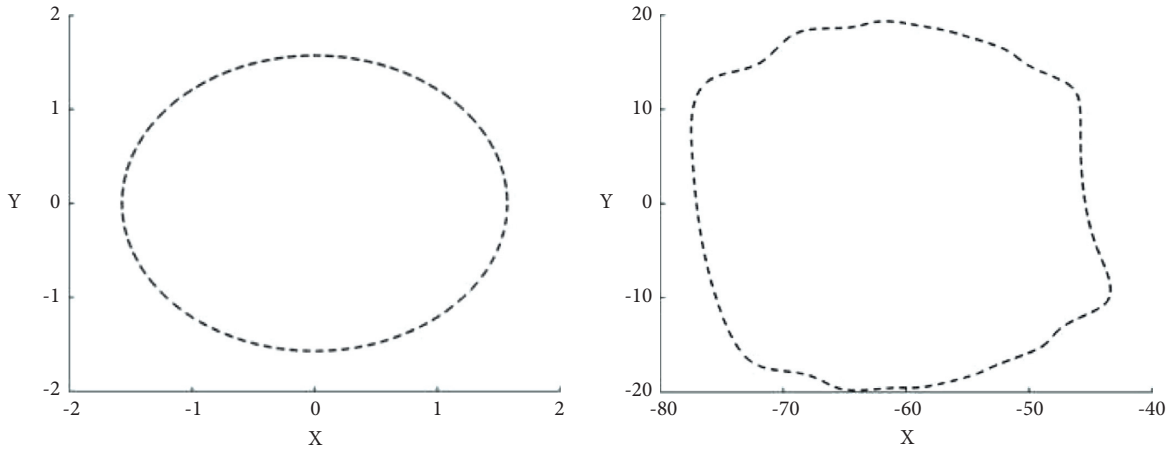


Figure 9: Continued.

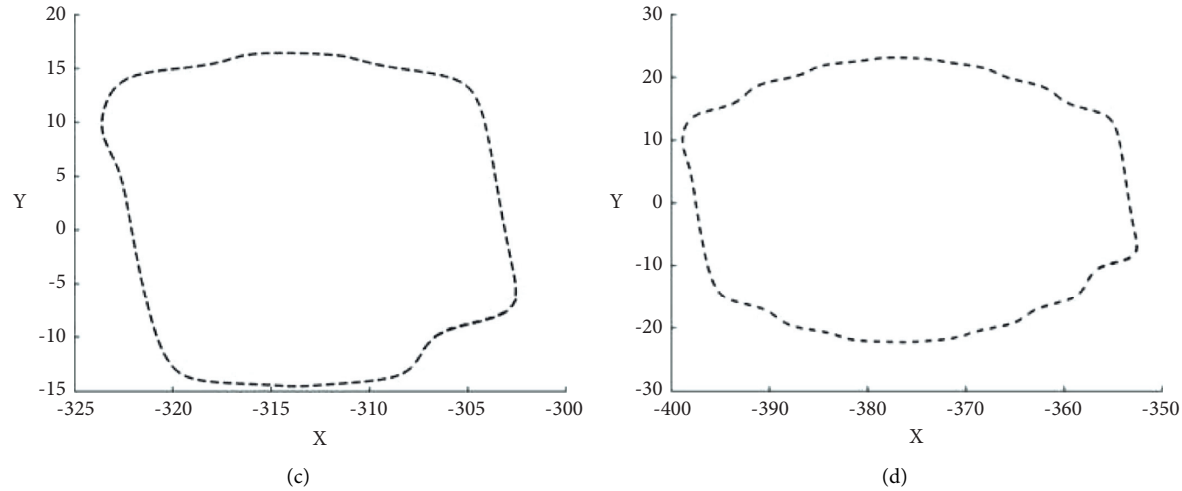


Figure 9: Phase portraits in the plane (x, y) for System (1) illustrating megastability obtained by considering random initial conditions distributed around the x-axis. For all initial conditions, the final state represents a limit cycle.

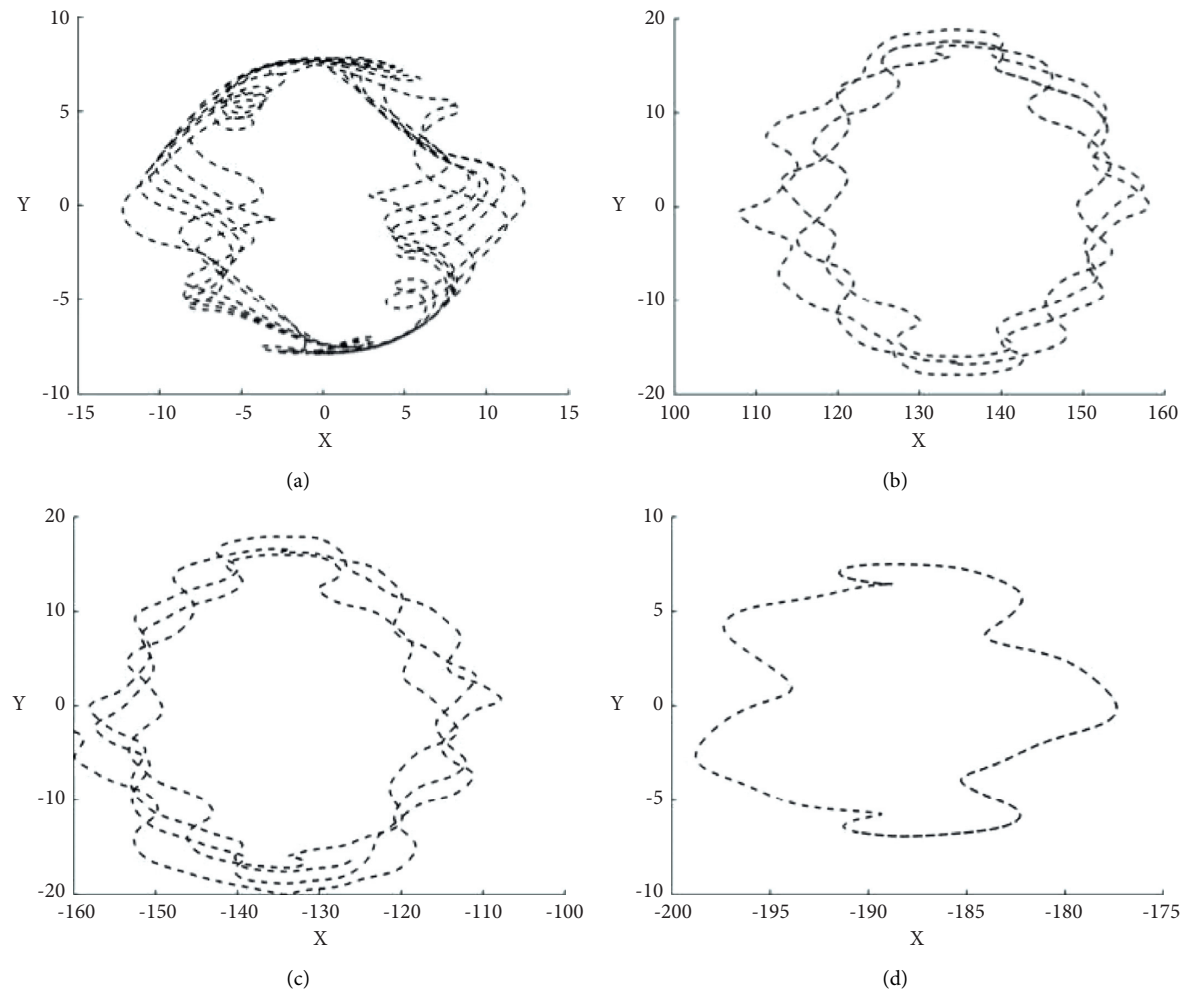


Figure 10: Phase portrait in the plane (x, y) for System (2) illustrating megastability obtained by considering random initial conditions distributed around the x-axis. The frequency of the forcing term is $\omega = 0.6$, and the amplitude $A = 2$. Adding a forcing term can make the system exhibit various dynamics.

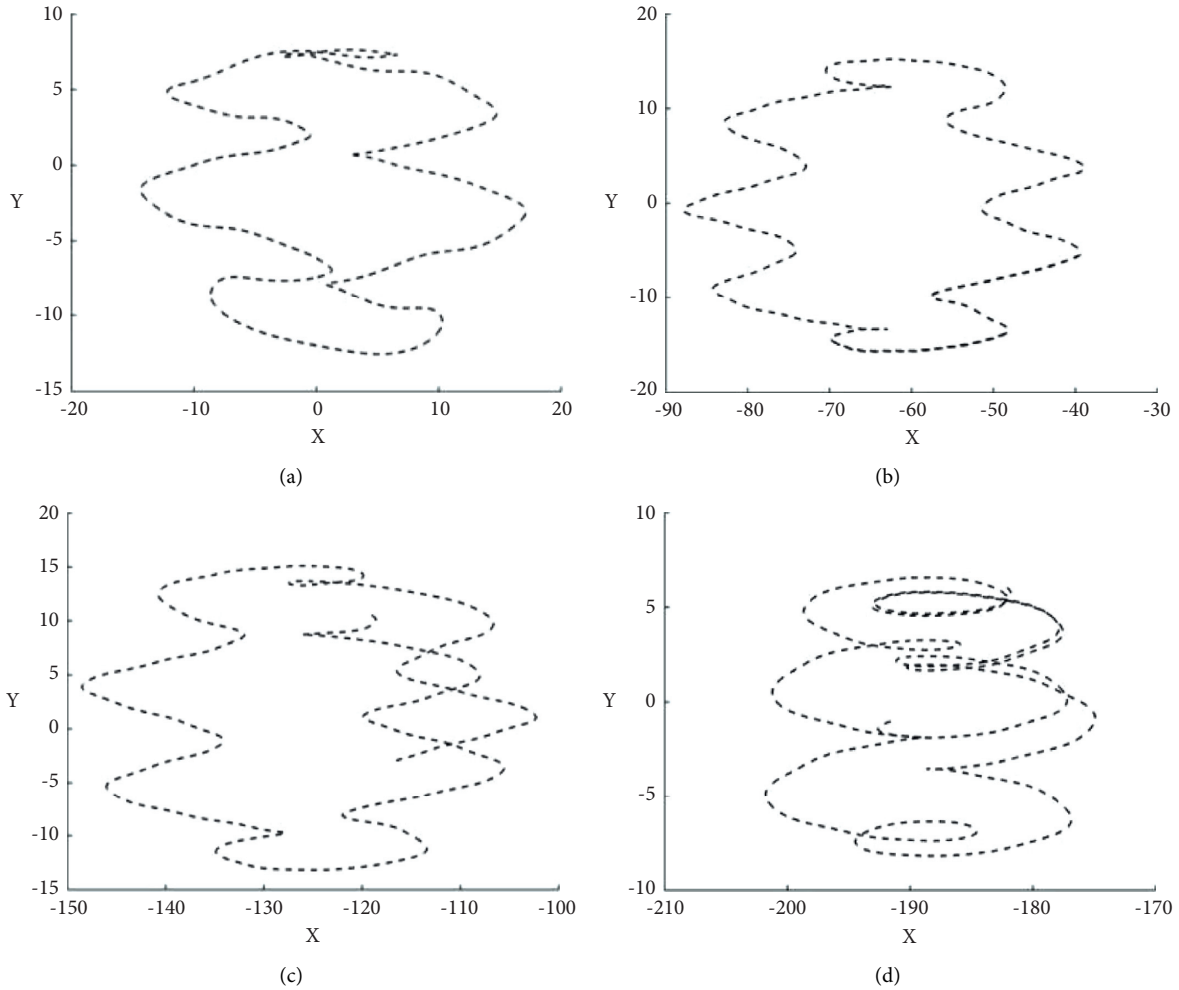


Figure 11: Phase portrait in the plane (x, y) for System (2) illustrating megastability obtained by considering random initial conditions distributed around the x-axis. The frequency of the forcing term is $\omega = 0.6$, and the amplitude $A = 4.3$. Adding a forcing term can make the system exhibit various dynamics.

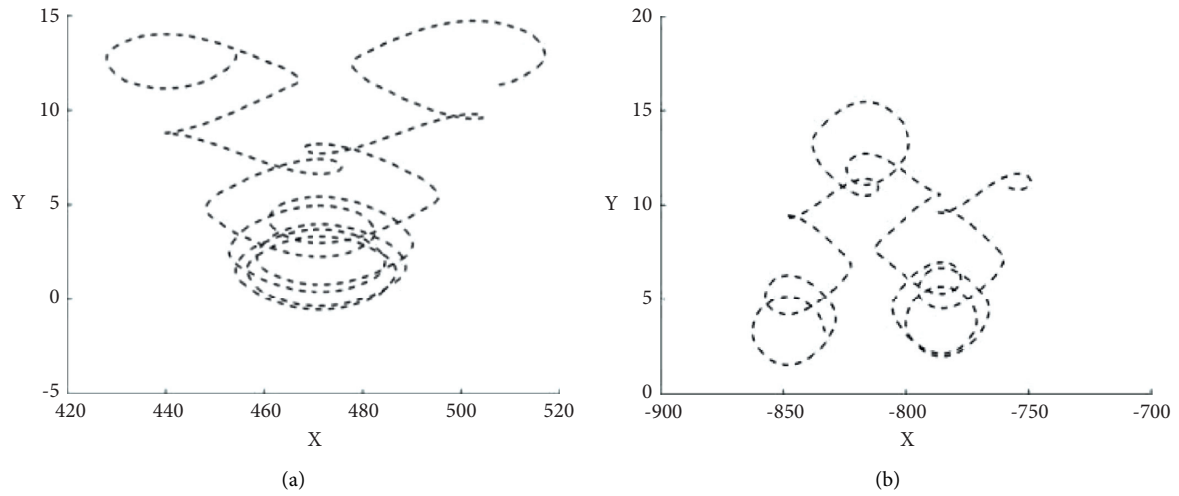


Figure 12: Continued.

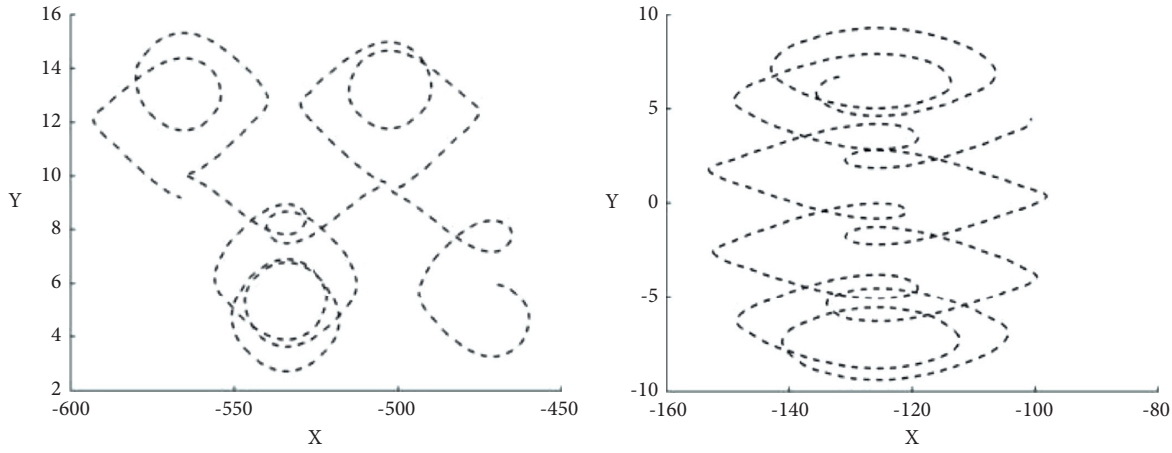


Figure 12: Phase portrait in the plane (x, y) for System (2) illustrating megastability obtained by considering random initial conditions distributed around the x-axis. The frequency of the forcing term is $\omega = 0.6$, and the amplitude $A = 9.7$. Adding a forcing term can make the system exhibit various dynamics.

$$\begin{aligned} x_{n+1} &= x_n + h\left[-0.1y_{n-1} + x_{n-1}\frac{\cos(r)}{r}\right]y_{n+1} \\ &= y_n + h\left[\sin\left(0.1x_{n-1}\right) + y_{n-1}\frac{\cos(r)}{r}\right], \quad \text{where } r = \sqrt{x^2 + y^2}, \end{aligned} \tag{6}$$

$$\begin{aligned} x_{n+1} &= x_n + h\left[-0.1y_{n-1} + x_{n-1}\frac{\cos(r)}{r} + A\,\sin(\omega t)\right]y_{n+1} \\ &= y_n + h\left[\sin\left(0.1x_{n-1}\right) + y_{n-1}\frac{\cos(r)}{r}\right], \quad \text{where } r = \sqrt{x^2 + y^2}. \end{aligned} \tag{7}$$

A system generator token is an important block that is dragged from the Xilinx block set library, which has information about the system generator model, through which it is possible to interface with the Vivado design tool to create an RTL design of the system. The phase planes of the proposed System (1) are shown in Figure 9, and the phase planes of System (2) are shown in Figures 10–12, which are obtained while running the system generator by changing initial conditions distributed around the x-axis.

4. Conclusion

A megastable system, the newest in the family of special chaotic systems, was designed and proposed. It was two-dimensional flow coexisting attractors in some islands, interestingly covered by megalimit cycles. No equilibrium point was found for the proposed two-dimensional system. However, the origin (0, 0) acted like an unstable point. Adding a forcing term to the proposed system, chaotic solutions and coexisting strange attractors were obtained. Different behaviors were observed by altering the amplitude of the forcing term. Since the system was found to have no equilibrium point, the attractors were considered in the category of hidden attractors. The dynamical properties of this new system were investigated utilizing some tools such as attractor plots, bifurcation diagrams, and LEs diagrams. Two bifurcation and LEs diagrams were plotted to show the effect of initial conditions in the system's behaviors and dynamics. Phase portraits of the novel megastable oscillator were presented by FPGA design. Xilinx system generator block diagrams of the proposed system and trigonometric functions were also presented. The proposed system is a low-dimensional system with the ability to exhibit chaos by adding a forcing term. So, it can be used in some applications, such as a random number generator or image encrypting as future works.

Data Availability

All the numerical simulation parameters are mentioned in the respective text part, and there are no additional data requirements for the simulation results.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors extend their gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through the research group program under grant number R. G. P. 2/48/42.

References

[1] J. C. Sprott, "Some simple chaotic flows," *Physical Review E*, vol. 50, no. 2, pp. R647–R650, 1994.

[2] J. C. Sprott, *Elegant Chaos: Algebraically Simple Chaotic Flows*, World Scientific, Singapore, 2010.

[3] Q. Deng, C. Wang, and L. Yang, "Four-wing hidden attractors with one stable equilibrium point," *International Journal of Bifurcation and Chaos*, vol. 30, Article ID 2050086, 2020.

[4] J. C. Sprott, "Some simple chaotic jerk functions," *American Journal of Physics*, vol. 65, no. 6, pp. 537–543, 1997.

[5] C. Xu, J. Sun, and C. Wang, "An image encryption algorithm based on random walk and hyperchaotic systems," *International Journal of Bifurcation and Chaos*, vol. 30, Article ID 2050060, 2020.

[6] Q. Wan, Z. Zhou, W. Ji, C. Wang, and F. Yu, "Dynamic analysis and circuit realization of a novel no-equilibrium 5D memristive hyperchaotic system with hidden extreme multistability," *Complexity*, vol. 2020, Article ID 7106861, 16 pages, 2020.

[7] K. Rajagopal, M. E. Cimen, S. Jafari et al., "A family of circulant megastable chaotic oscillators, its application for the detection of a feeble signal and PID controller for time-delay systems by using chaotic SCA algorithm," *Chaos, Solitons & Fractals*, vol. 148, Article ID 110992, 2021.

[8] T. Lu, C. Li, X. Wang, C. Tao, and Z. Liu, "A memristive chaotic system with offset-boostable conditional symmetry," *The European Physical Journal Special Topics*, vol. 229, no. 6-7, pp. 1059–1069, 2020.

[9] Z. Gu, C. Li, X. Pei, C. Tao, and Z. Liu, "A conditional symmetric memristive system with amplitude and frequency control," *The European Physical Journal Special Topics*, vol. 229, no. 6-7, pp. 1007–1019, 2020.

[10] Z. Wei, "Dynamical behaviors of a chaotic system with no equilibria," *Physics Letters A*, vol. 376, no. 2, pp. 102–108, 2011.

[11] M. Molaie, S. Jafari, J. C. Sprott, and S. M. R. Hashemi Golpayegani, "Simple chaotic flows with one stable equilibrium," *International Journal of Bifurcation and Chaos*, vol. 23, Article ID 1350188, 2013.

[12] A. N. Pisarchik and U. Feudel, "Control of multistability," *Physics Reports*, vol. 540, no. 4, pp. 167–218, 2014.

[13] H. Lin, C. Wang, and Y. Tan, "Hidden extreme multistability with hyperchaos and transient chaos in a hopfield neural network affected by electromagnetic radiation," *Nonlinear Dynamics*, vol. 99, no. 3, pp. 2369–2386, 2020.

[14] M. Chen, Y. Feng, H. Bao et al., "Hybrid state variable incremental integral for reconstructing extreme multistability in memristive jerk system with cubic nonlinearity," *Complexity*, vol. 2019, Article ID 8549472, 16 pages, 2019.

[15] M. Chen, M. Sun, H. Bao, Y. Hu, and B. Bao, "Flux–charge analysis of two-memristor-based chua's circuit: dimensionality decreasing model for detecting extreme multistability," *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 2197–2206, 2019.

[16] J. C. Sprott, S. Jafari, A. J. M. Khalaf, and T. Kapitaniak, "Megastability: coexistence of a countable infinity of nested attractors in a periodically-forced oscillator with spatially-periodic damping," *The European Physical Journal Special Topics*, vol. 226, no. 9, pp. 1979–1985, 2017.

[17] Z. Wang, H. R. Abdolmohammadi, M. Chen et al., "A new megastable chaotic oscillator with singularity," *The European Physical Journal Special Topics*, vol. 229, no. 12-13, pp. 2341–2348, 2020.

[18] M. F. Danca, N. V. Kuznetsov, and G. Chen, "Approximating hidden chaotic attractors via parameter switching," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, Article ID 013127, 2018.

[19] S. Cang, Y. Li, R. Zhang, and Z. Wang, "Hidden and self-excited coexisting attractors in a lorenz-like system with two equilibrium points," *Nonlinear Dynamics*, vol. 95, no. 1, pp. 381–390, 2019.

[20] M. Chen, C. Wang, H. Bao et al., "Reconstitution for interpreting hidden dynamics with stable equilibrium point," *Chaos, Solitons & Fractals*, vol. 140, Article ID 110188, 2020.

[21] G. A. Leonov, N. V. Kuznetsov, and T. N. Mokaev, "Homoclinic orbits, and self-excited and hidden attractors in a lorenz-like system describing convective fluid motion," *The European Physical Journal Special Topics*, vol. 224, no. 8, pp. 1421–1458, 2015.

[22] E. Tlelo-Cuautle, V. H. Carbajal-Gomez, P. J. Obeso-Rodelo, J. J. Rangel-Magdaleno, and J. C. Núñez-Pérez, "FPGA realization of a chaotic communication system applied to image processing," *Nonlinear Dynamics*, vol. 82, no. 4, pp. 1879–1892, 2015.

[23] H. Li, Z. Hua, H. Bao et al., "Two-dimensional memristive hyperchaotic maps and application in secure communication," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, 2020.

[24] B. Bao, M. Peol, H. Bao et al., "No-argument memristive hyper-jerk system and its coexisting chaotic bubbles boosted by initial conditions," *Chaos, Solitons & Fractals*, vol. 144, Article ID 110744, 2021.

[25] Q. Hong, H. Chen, J. Sun, and C. Wang, "Memristive circuit implementation of a self-repairing network based on biological astrocytes in robot application," *IEEE transactions on neural networks and learning systems*, no. 99, 2020.

[26] N. Tsafack, S. Sankar, B. Abd-El-Atty et al., "A new chaotic map with dynamic analysis and encryption application in internet of health things," *IEEE Access*, vol. 8, Article ID 137731, 44 pages, 2020.

[27] E. Tlelo-Cuautle, J. J. Rangel-Magdaleno, A. D. Pano-Azucena, P. J. Obeso-Rodelo, and J. C. Nuñez-Perez, "FPGA realization of multi-scroll chaotic oscillators," *Communications in Nonlinear Science and Numerical Simulation*, vol. 27, no. 1–3, pp. 66–80, 2015.

[28] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[29] L. F. Ávalos-Ruiz, C. J. Zúñiga-Aguilar, J. F. Gómez-Aguilar, R. F. Escobar-Jiménez, and H. M. Romero-Ugalde, "FPGA implementation and control of chaotic systems involving the variable-order fractional operator with mittag-leffler law," *Chaos, Solitons & Fractals*, vol. 115, pp. 177–189, 2018.

[30] H. Chen, S. He, A. D. Pano Azucena et al., "A multistable chaotic jerk system with coexisting and hidden attractors: dynamical and complexity analysis, FPGA-based realization, and chaos stabilization using a robust controller," *Symmetry*, vol. 12, no. 4, p. 569, 2020.

[31] K. Rajagopal, F. Nazarimehr, A. Karthikeyan, A. Srinivasan, and S. Jafari, "CAMO: self-excited and hidden chaotic flows,"

International Journal of Bifurcation and Chaos, vol. 29, no. 11, Article ID 1950143, 2019.

[32] A. Karthikeyan and K. Rajagopal, "FPGA implementation of fractional-order discrete memristor chaotic system and its commensurate and incommensurate synchronisations," *Pramana*, vol. 90, no. 1, p. 14, 2018.

[33] B. Karakaya, A. Gülten, and M. Frasca, "A true random bit generator based on a memristive chaotic circuit: analysis, design and FPGA implementation," *Chaos, Solitons & Fractals*, vol. 119, pp. 143–149, 2019.