PAPER

Medical image encryption system based on a simultaneous permutation and diffusion framework utilizing a new chaotic map

Physica Scripta

PAPER

Medical image encryption system based on a simultaneous permutation and diffusion framework utilizing a new chaotic map

Zhen Le[1,4], **Quanjun Li**[1,4], **Huang Chen**[2], **Shuting Cai**[3], **Xiaoming Xiong**[3] and **Linqing Huang**[1]

[1] School of Advanced Manufacturing, Guangdong University of Technology, Jieyang 522000, People's Republic of China
[2] School of Automation, Guangdong University of Technology, Guangzhou 510006, People's Republic of China
[3] School of Integrated Circuits, Guangdong University of Technology, Guangzhou 510006, People's Republic of China
[4] These authors contributed equally to this work.

E-mail: hlq@gdut.edu.cn

RECEIVED
16 January 2024

REVISED
2 April 2024

ACCEPTED FOR PUBLICATION
8 April 2024

PUBLISHED
18 April 2024

Keywords: chaotic map, Josephus traversing, simultaneous permutation and diffusion, medical image encryption, efficiency

Abstract

In telemedicine, diverse medical images transmitted between doctors and patients contain sensitive personal information. Thus, there is an urgent need for reliable and efficient medical image encryption to protect these medical images during transmission. In this paper, a simultaneous permutation and diffusion framework (SPDF) is introduced for medical image encryption based on a new chaotic map. Firstly, combining the Chebyshev map and the iterative chaotic map with infinite collapse (ICMIC), we propose a one-dimensional chaotic system (1D-CICMIC) which exhibits higher ergodicity and unpredictability compared to other 1D chaotic maps through comprehensive analyses. Secondly, in order to enhance permutation effect, we modify traditional Josephus traversing with a dynamic scrambling method where the scrambling scheme of the current pixel depends on the value of the previous diffused pixel. Thirdly, we develop a simultaneous permutation and diffusion framework, wherein the diffusion is embedded into the modified Josephus traversing to prevent attackers from targeting the scrambling and diffusion phases separately. Finally, based on 1D-CICMIC and SPDF, an encryption system is proposed. It adopts plaintext correlation in the diffusion operation, which strikes a balance between ciphertext sensitivity and plaintext sensitivity, offering resistance against chosen-plaintext attack (CPA), noise attack and data loss. Simulation results show that the proposed algorithm has high encryption efficiency and can withstand various common attacks.

1. Introduction

The progression of telemedicine has facilitated the transmission of diverse images such as MRI, x-ray, and CT between patients and doctors. However, the transmission of unencrypted information poses a substantial threat to privacy. Traditional encryption methods such as the Advanced Encryption Standard (AES), Rivest Shamir Adleman (RSA) algorithm, and Data Encryption Standard (DES), specifically designed for safeguarding textual messages, encounter challenges when applied to the encryption of medical images. This is due to the intrinsic features of medical images, particularly the extensive data volume [1]. Given these image's distinctive features and the high value of privacy, there is a pressing need for specialized image encryption methods tailored to safeguard medical images.

Until now, numerous image encryption methods have been developed to safeguard medical images [1–5]. Among these, chaotic-based image encryption schemes have garnered increasing interest owing to the chaotic system's intrinsic randomness, sensitivity to initial values, and other exceptional characteristics. In 1998, Fridrich [6] presented a new image encryption architecture, which employed the pseudo-random numbers generated by the chaotic system for both scrambling and diffusion stage. In recent years, a plethora of encryption algorithms have been proposed, which utilize a similar encryption framework [1–5, 7–26]. In these work, one or multiple pseudo-random sequences generated by 1D or high-dimensional chaotic maps are needed for scrambling or diffusion. However, most of the common 1D chaotic maps have a small chaotic range and lack

sufficiently complex dynamical behaviors. In [22], Pak *et al* proposed a new image encryption method based on a new 1D chaotic map that combines classic chaotic maps. However, Wang *et al* discovered an algebraic weakness in [22] and attacked its equivalent cryptographic method using chosen plaintext attack in [27]. To address the limitations of 1D chaotic maps, scholars have proposed alternative approaches such as the 1D robust chaotic maps [15, 28] and the 1D chaotic maps without fixed points [24].

For an encryption system, the strength of encryption is a vital factor. In certain encryption schemes, the encryption process or key stream generation is independent of the original image which is defined as a non-plaintext-related (NPR) cryptosystem. In [7], Kumar *et al* developed an algorithm that uses DNA addition and circular shifting to encrypt RGB images. The method in [8] employs orthogonal Latin cubes to scramble a 3D bit matrix and achieves resistance to the statistical analysis, benefiting from the fact that the Latin cube has uniform histogram. However, NPR encryption schemes [7, 8] face the risk of being deciphered because of the weak relationship between plaintext and encryption operation or key stream. For example, an analysis conducted by Akhavan *et al* [29] showed that the encryption algorithm presented by Kumar *et al* [7] is being susceptible to CPA, as images encrypted with the same key could be easily recovered with as few as two chosen plain images. To counter CPA, plaintext-related (PR) encryption algorithms have been developed by scholars. In PR encryption mechanisms, the key stream generation or scrambling and diffusion operations are impacted by the plaintext information. For instance, in [10], the hash function SHA-256 is employed to generate plaintext-related internal keys for both the scramble and diffusion phases. Chai *et al* [11] introduced an approach for increasing sensitivity to plaintext by applying plaintext-related Latin-square-based block permutation, along with a diffusion operation that relies on the plaintext. In spite of the introduction of plaintext-related encryption aimed at improving plaintext sensitivity, several cryptanalysis have successfully attacked PR algorithms by using CPA [2, 30]. This vulnerability arises from various factors, including traditional scrambling-diffusion structure that can be broken separately. Some works [31–34] performed the scrambling and diffusion operation simultaneously and inseparably to solve this problem. As an illustration, a novel mechanism [31] integrates permutation and diffusion into one encryption stage to pursue high security.

In recent years, the Josephus traversing has been utilized in image encryption as a scrambling technique due to its simple principle and and effectiveness. However, traditional Josephus traversing has two drawbacks. One is the fixed step size, which most scholars [9, 13, 14] try to solve by using a dynamic step size. The dynamic step size is determined by a pseudo-random sequence generated with a chaotic system. The other issue is that in each iteration, the target element needs to be taking out from the sequence, and the position of all remaining elements should be updating. As a consequence, this process leads to an extended scrambling time. Guan *et al* [9] utilized an element swapping technique in lieu of removing an element to improve efficiency.

To address the above problems, we introduce a simultaneous permutation and diffusion framework (SPDF) for medical image encryption. The main contributions of this paper are as follows: (1) We propose a new 1D-Chebyshev-ICMIC map (1D-CICMIC) and comprehensive analyses show that the map has higher ergodicity and unpredictability than existing 1D chaotic maps. (2) We improve the Josephus traversing by employing two different scrambling schemes for each element for the purpose of increasing the permutation effect, distinguishing it from the traditional method that relies on a single scheme for scrambling. (3) Unlike previous algorithms that only use the Josephus traversing for scrambling, the proposed SPDF embeds diffusion operation into the Josephus traversing and performs the scrambling and diffusion simultaneously. This prevents attackers from targeting the scrambling and diffusion phases separately and enhances the security of the algorithm. (4) The SPDF employs plaintext correlation for diffusion operations, which magnifies small variations in plaintext, leading to high plaintext sensitivity and resistance to CPA.

In this article, section 2 provides an introduction to conventional Josephus traversing and section 3 presents 1D-CICMIC chaotic system. Pseudo-random sequence generation and the SPDF algorithm are presented in section 4, followed by the decryption process in section 5. Section 6 assesses the security performance of the proposed algorithm, and section 7 gives a conclusion.

2. Preliminaries

The classic Josephus Problem, which is a well-known problem in the field of mathematics and computer science, involves removing elements from a circular queue according to a specific rule. Within the realm of image encryption, it is commonly utilized to scramble the plaintext sequence. Typically, in Josephus traversing, a 2D plaintext image is converted into a 1D vector, and elements are extracted by using a fixed interval step to generate a scrambled vector. Equation (1) defines the process of Josephus traversing, where n is the length of the sequence to be scrambled, m is the initial position, and $step$ is the interval step.

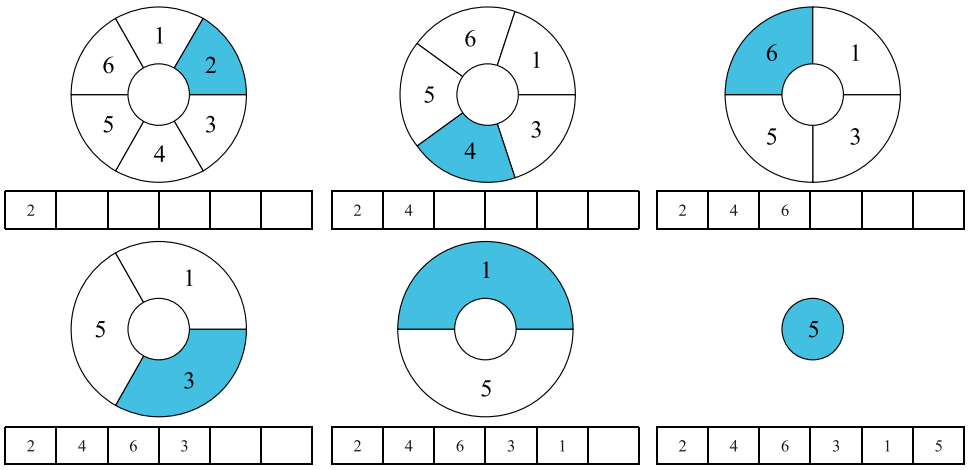


Figure 1. Conventional Josephus traversing process with *step* = 2.

Table 1. Tested maps.

Name	Mathematical expressions	Parameter settings	State variables
1D-CICMIC	$x_{n+1} = \sin\left(\frac{a}{\cos(a \times \arccos(x_n))}\right)$	$a \in (0, \infty)$	x∈(−1, 1)
1D-ICMIC	$x_{n+1} = \sin\left(\frac{\gamma}{x_n}\right)$	$\gamma \in (0, \infty)$	x∈(−1, 1)
sine	$x_{n+1} = b \times \sin(\pi x_n)$	$b \in (0, \infty)$	x∈(−∞, ∞)
1-DFCS	$x_{n+1} = \frac{\cos((\eta x_n + 1)^2 + 1)}{\sin((\eta x_n + 1)^2 + 1) + 2}$	$\eta \in, (0, \infty)$	x∈$(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$
1D-Chebyshev	$x_{n+1} = \cos(k \times \arccos(x_n))$	$k \in (0, \infty)$	x∈(−1, 1)

$$J(n,\ m,\ step) \tag{1}$$

The steps of Josephus traversing are as follows: firstly, starting from the m-th position of the 1D vector, we scan the vector in a clockwise direction. Secondly, remove the *step*-th pixel from the queue and start a new scan from the next pixel. Thirdly, repeat the previous step until all pixels have been removed from the queue. To prevent the *step* from exceeding n, it will be calculated by equation (2).

$$step = mod(step,\ n) + 1 \tag{2}$$

This process effectively scrambles the vector and generates a new sequence. For example, set $n = 6$, $m = 0$, $step = 2$ and a vector is [1, 2, 3, 4, 5, 6], the process is shown in figure 1. The new vector generated after scrambling is [2, 4, 6, 3, 1, 5].

3. The proposed new chaotic system

The classic ICMIC and Chebyshev map are defined as equation (3) and equation (4), respectively. However, these maps are not sensitive enough to the initial values, and their chaotic ranges are narrow, which will be analyzed later. In this section, based on the ICMIC map and Chebyshev map, we propose a new chaotic map with a wide chaotic range named 1D-CICMIC map, defined by equation (5).

$$x_{n+1} = \sin\left(\frac{\gamma}{x_n}\right) \tag{3}$$

$$x_{n+1} = \cos(k \times \arccos(x_n)) \tag{4}$$

$$x_{n+1} = \sin\left(\frac{a}{\cos(a \times \arccos(x_n))}\right) \tag{5}$$

where γ, k and a are the system parameters. To demonstrate the superiority of the proposed chaotic map, we will analyze the chaotic behaviors of different maps which are illustrated in table 1.

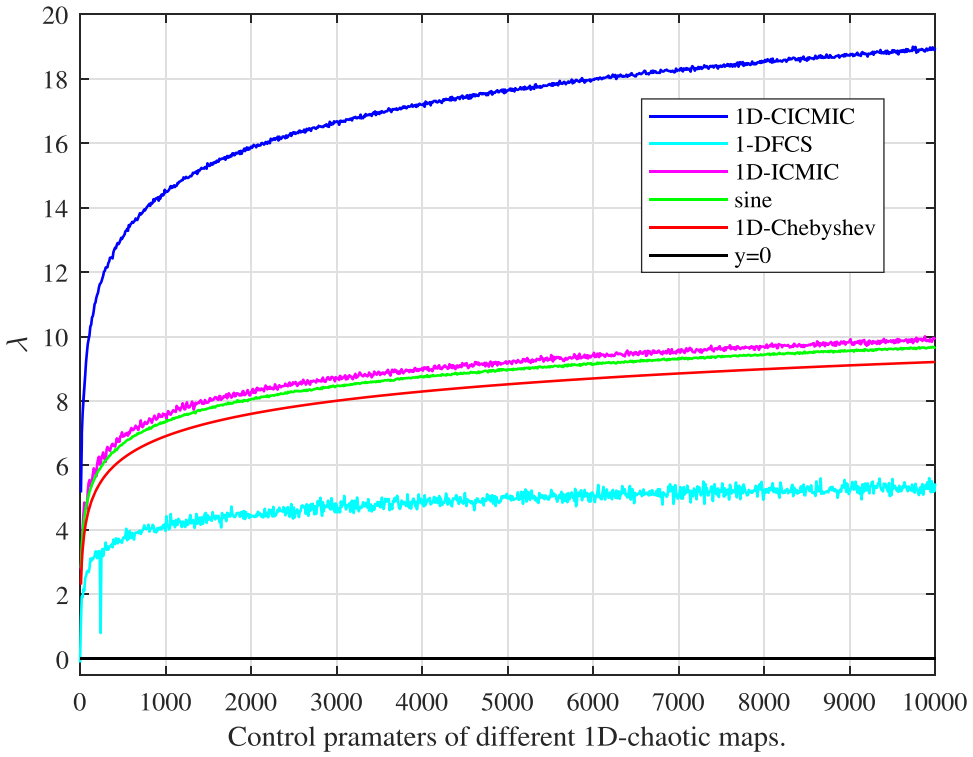


Figure 2. The Lyapunov exponent of 1D-CICMIC, 1-DFCS, 1D-ICMIC, Sine and 1D-Chebyshev map.

3.1. Lyapunov exponent test

The Lyapunov exponent test is an effective method to analyze whether a non-linear dynamic system is sensitive to changes in initial conditions [35], which can be calculated using equation (6).

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{n\ =\ 0}^{N-1} \ln|f'(x_n)| \tag{6}$$

where n is an integer and $f'(x_n)$ is the value of derivative of the mapping function at x_n. The Lyapunov diagram of different maps is plotted in figure 2. One can see that when the control parameters of different 1D-chaotic maps $\in$(0, 10 000), the Lyapunov exponent of 1D-CICMIC is positive and larger than the ones of other chaotic maps.

3.2. Bifurcation analysis

The bifurcation diagram can visually show a dynamical system's long-term behavior under different control parameter values. The bifurcation diagram of the 1D-CICMIC, 1D-Chebyshev, Sine map, 1D-ICMIC [36], and 1-DFCS(novel one-dimensional chaotic system based on the fraction of cosine over sine) [21] are shown in figure 3, which indicates that the 1D-CICMIC map features a larger chaotic region.

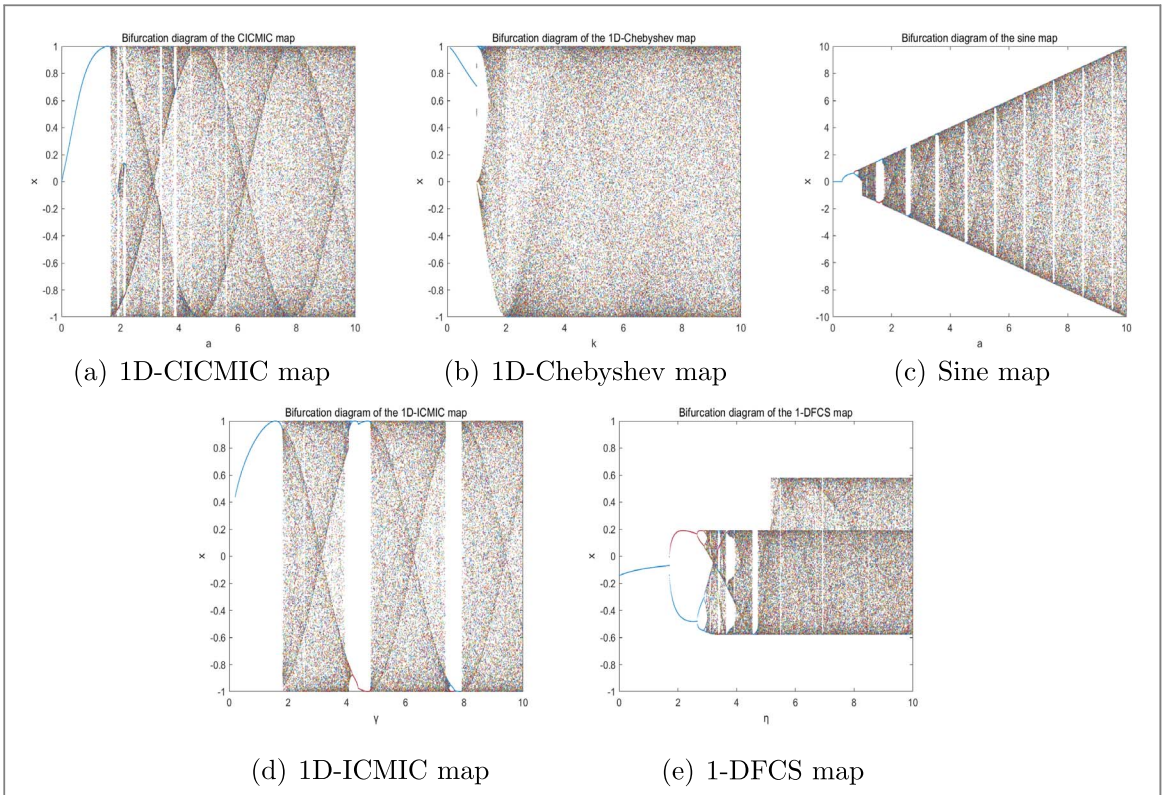
3.3. Sample entropy

The self-similarity of a time series generated by dynamical systems can be measured using Sample entropy (SE) [37]. A larger SE value indicates lesser regularity of the time series and higher complexity of the dynamical system. The value of SE of the time series X can be implemented as equation (7) and equation (8).

$$D_m(i) = [X(i),\ X(i+1),\ldots,X(i+m-1)] \tag{7}$$

$$SE(m,\ r,\ n) = -\log\frac{A}{B} \tag{8}$$

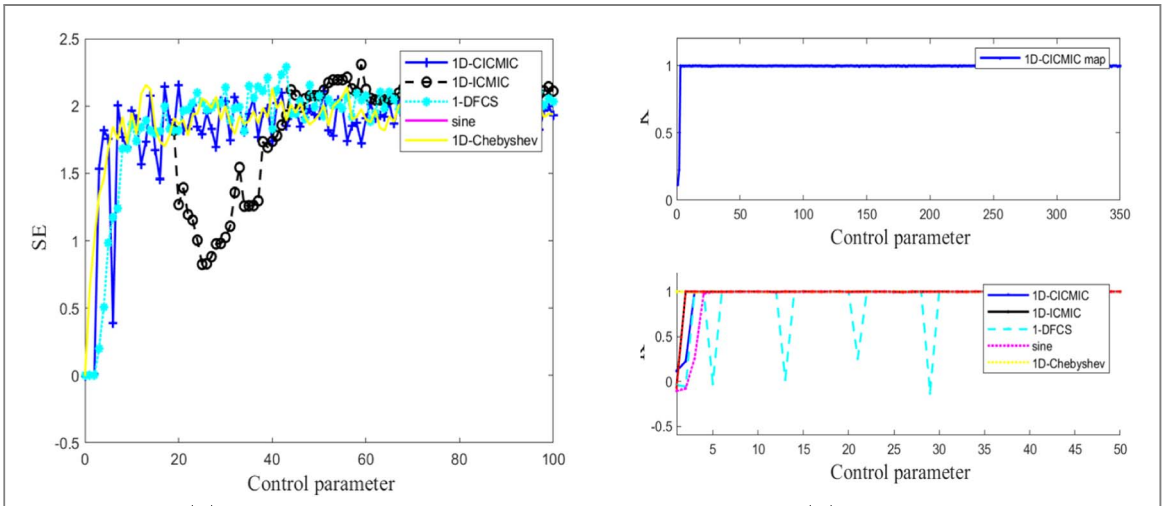
Here, template vector $D_m(i)$ defined in equation (7) with the size of m is set to be taken from time series, A and B in equation (8) are the number vectors that satisfy $d[X_{m+1}(i), X_{m+1}(j)] < r$ and $d[X_m(i), X_m(j)] < r$ respectively. $d[X(i), X(j)]$ denotes the Chebyshev distance between $X(i)$ and $X(j)$, and r is the given distance. In our experiment, m and r are chosen as 2 and 0.2std, respectively. A comparative experiment of SE is given, and its results are shown in figure 4(a). One can see that the time series produced by the proposed 1D-CICMIC has better performance in SE analysis.



(a) 1D-CICMIC map (b) 1D-Chebyshev map (c) Sine map

(d) 1D-ICMIC map (e) 1-DFCS map

Figure 3. The Bifurcation analysis of different 1D-chaotic maps.



(a) Sample Entropy (b) The 0-1 test

Figure 4. The Sample Entropy and 0-1 test of different 1D-chaotic maps.

3.4. 0-1 test

The 0-1 test is a chaos indicator that estimates the rate of growth of a nonlinear dynamic system's series [38]. And the $p(n)$ and $s(n)$ can be defined as equation (9), where the stant $c \in (0, 2\pi)$ and $\phi(n)$ is a 1D time series.

$$\begin{cases} p(n+1) = p(n) + \phi(n)\cos(cn) \\ s(n+1) = s(n) + \phi(n)\sin(cn) \end{cases}, \; n = 1, 2, 3, \ldots \tag{9}$$

The time-averaged mean square displacement $M(n)$ and the asynchronous growth rate K can be calculated by equation (10) and equation (11), respectively. Figure 4(b) shows that the K is close to 1 which means the system is in a chaotic state.

Table 2. NIST test results of 1D-CICMIC.

Test index	P−value	Result
Frequency (Monobit) Test	0.816537	PASS
Frequency Test within a Block	0.224912	PASS
Runs Test	0.383827	PASS
Longest Run of Ones in a Block Test	0.202268	PASS
Binary Matrix Rank Test	0.202268	PASS
Discrete Fourier Transform (Spectral) Test	0.350485	PASS
Non-overlapping Template Matching Test	0.541081	PASS
Overlapping Template Matching Test	0.883171	PASS
Maurers universal Statistical Test	0.595549	PASS
Linear Complexity Test	0.946308	PASS
Serial Test	0.719747	PASS
Approximate Entropy Test	0.013569	PASS
Cumulative Sums (Cusums) Test	0.967102	PASS
Random Excursions Test	0.503500	PASS
Random Excursions Variant Test	0.377877	PASS

$$M(n) = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}([p(i+n) - p(i)]^2 + [s(i+n) - s(i)]^2), \qquad n = 1, 2, 3 \tag{10}$$

$$K = \lim_{n\to\infty} \frac{\log M(n)}{\log n}. \tag{11}$$

3.5. NIST SP 800-22 test

The National Institute of Standards and Technology (NIST) SP800-22 is an effective tool to test the randomness of the output sequences of a chaotic system [38]. Firstly, a time series with 12,500,000 elements is generated by 1D-CICMIC map. Then, each element is mapped into 0-255. Finally, the integer part of each value in the mapped sequence is converted to 8 bits to obtain a binary number of 1 megabyte in size. Table 2 shows that all the p-values fall within the range (0.001, 1), which indicates that 1D-CICMIC can produce a random enough sequence.

4. The proposed encryption scheme

4.1. Secret key formulation

The proposed scheme is designed with four secret keys, consisting of two system parameters a_1 and a_2, as well as two initial values y_1 and z_1 for the chaotic maps. These keys, a_1, a_2, y_1, and z_1, are respectively denoted as K_1, K_2, K_3, and K_4. The ranges of keys are defined as follows: $K_1 \in (2, 12)$, $K_2 \in (2, 12)$, $K_3 \in (0, 1)$, and $K_4 \in (0, 1)$.

4.2. Pseudo-random sequence generation

The detailed steps of pseudo-random sequence generation are as follows:

Step 1: Convert a plaintext image with dimensions $M \times N$ into a vector A with a size of $L = M \times N$.

Step 2: We set K_1 and K_3 as the control parameter and the initial value of the 1D-CICMIC chaotic system, respectively, and then iterate it M times to obtain the pseudo-random vector Y. Similarly, we acquire the vector Z with the length of N using K_2 and K_4. The vectors Y and Z are shown in equation (12).

$$\begin{cases} Y = \{y_1, y_2, y_3, \cdots, y_M\} \\ Z = \{z_1, z_2, z_3, \cdots, z_N\} \end{cases} \tag{12}$$

Step 3: Obtain an $M \times N$ matrix X_{2D} and transform it to the sequence X with a length of L by equation (13):

$$\begin{cases} X_{2D}(i, j) = floor(mod(Y(i) \times Z(j) \times 10^9, 256)) \\ X = reshape(X_{2D}', [1, L]); \end{cases} \tag{13}$$

where $i = 1, 2, 3 \cdots M, j = 1, 2, 3 \cdots N$.

4.3. SPDF

In this section, we introduce the SPDF in detail. The encryption process is illustrated in figure 5, which consists of two rounds of simultaneous permutation and diffusion.

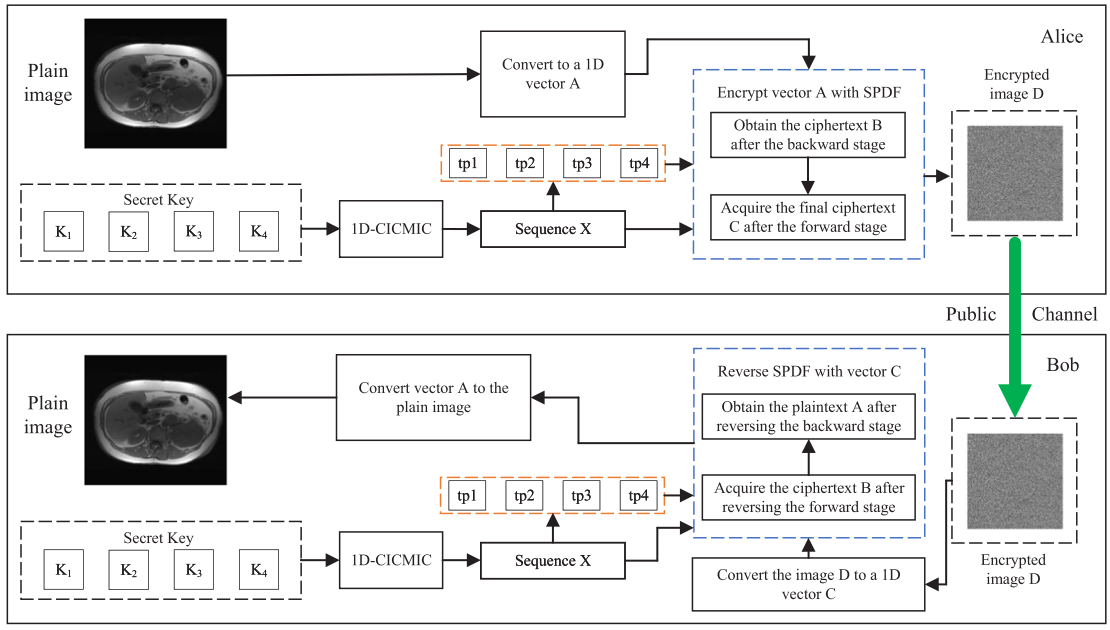


Figure 5. The proposed cryptosystem.

The detailed steps of SPDF are divided into two parts based on the processing order: backward and forward. In the first stage, the operation proceeds from the element $A(L)$ to $A(1)$, and in the second stage, the order of the operation is reversed.

Step 1: Diffuse the element $A(L)$ using equation (14).

$$B(L) = mod(A(L) + X(1),\ 256) \tag{14}$$

Step 2: For the remaining elements in vector A, their target positions tp are determined based on the value of the previous diffused pixel B_{pre}. Two target position vectors, $tp1$ and $tp2$, are generated using equation (15) and equation (16).

$$tp1(i) = mod(X(i) + 1,\ i) + 1 \tag{15}$$

$$tp2(i) = mod(tp2(i + 1) + tp1(i),\ i) + 1 \tag{16}$$

where i is the current position ranging from $L - 1$ to 1. If B_{pre} is greater than 128, the element at the current position is then exchanged with the element at $tp1$; otherwise, it is exchanged with the element at $tp2$. Additionally, $tp2(L)$ is 1.

Step 3: Diffuse $A(i)$ using equation (17) and equation (18) with the element in X and previous diffused value in ciphertext vector B.

$$B(i) = mod(B(i + 1) + A(i) + X(q1),\ 256) \tag{17}$$

$$q1 = tp2(B(i + 1) + 1) + 1 \tag{18}$$

where i is the current position ranging from $L - 1$ to 1.

Step 4: Iterate Step 2 and 3 until all elements of the plaintext vector A have been processed, resulting in the ciphertext vector B with a size of L.

Step 5: In the forward stage, diffuse the element $A(1)$ using equation (19) with the element at position 1 of the pseudo-random sequence X.

$$C(1) = mod(B(1) + X(1),\ 256) \tag{19}$$

For the remaining elements in vector B, the exchange rule is similar to the backward stage. But the generation of their target positions are different.

Step 6: Two target position vectors, $tp3$ and $tp4$, are generated using equation (20) and equation (21).

$$tp3(i) = mod(X(i) + 1,\ L - i + 1) + i - 1 \tag{20}$$

$$tp4(i) = mod(tp4(i + 1) + tp3(i),\ L - i + 1) + i - 1 \tag{21}$$

where i is the current position ranging from 2 to L. If B_{pre} is greater than 128, the element at the current position is then exchanged with the element at $tp3$; otherwise, it is exchanged with the element at $tp4$ and $tp4(1)$ is 1.

Step 7: Diffuse $B(i)$ using equation (22) and equation (23) with the element in X and the previous diffused value in ciphertext vector C.

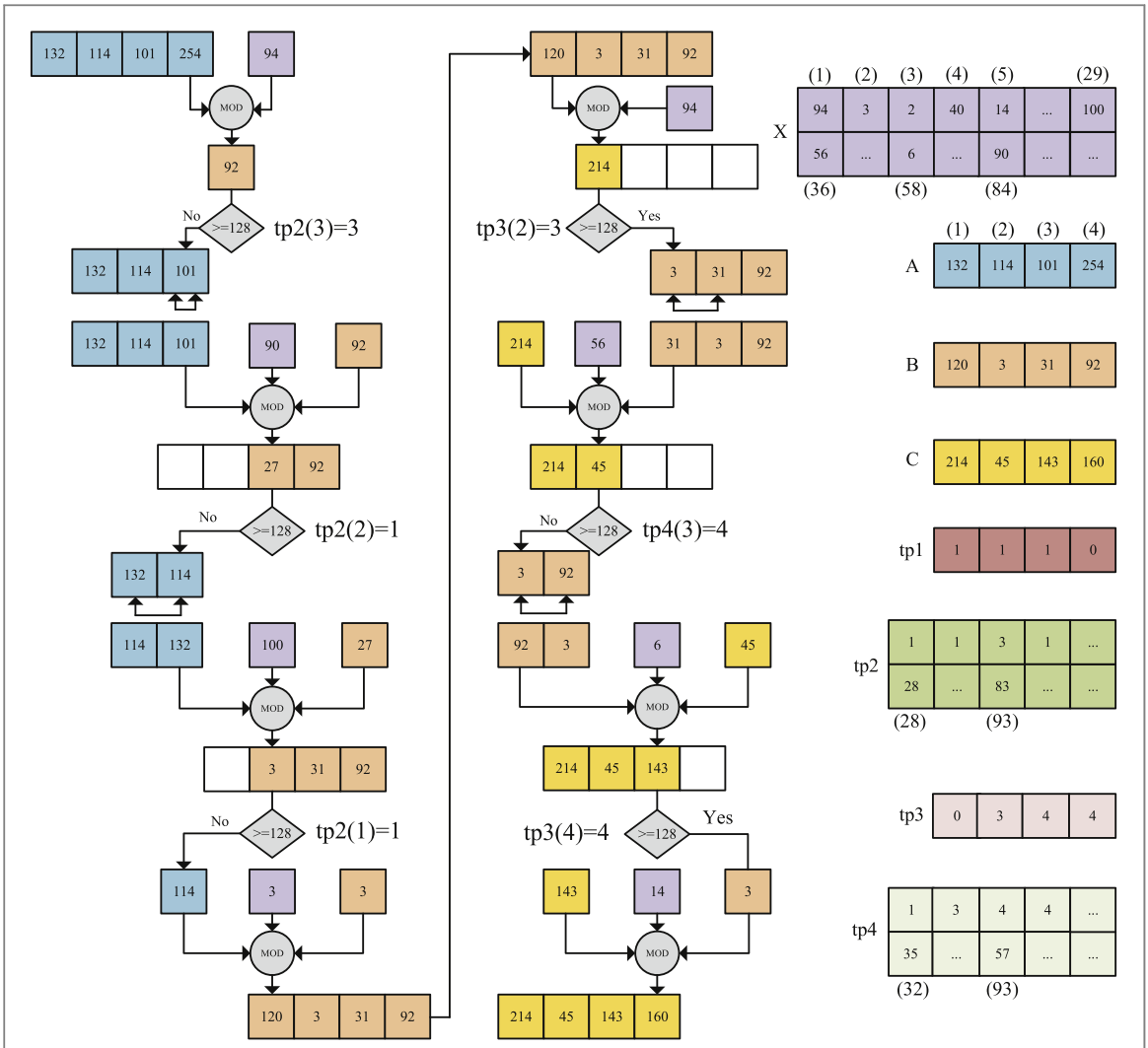


Figure 6. An illustration of SPDF.

$$C(i) = mod(C(i-1) + B(i) + X(q2), 256) \tag{22}$$

$$q2 = tp4(C(i-1) + 1) + 1 \tag{23}$$

where i is the current position ranging from 2 to L.

Step 8: Iterate Step 6 and 7 until all elements of the plaintext vector B have been processed, resulting in the ciphertext vector C with a size of L.

It's worth noting that when the length of the plaintext A is less than 256, the generation length of X, $tp2$ and $tp4$ will extend to 258.

SPDF, with the parameter $L = 4$, is illustrated in figure 6. The pseudocode of the SPDF encryption algorithm is presented in algorithm 1.

Algorithm 1. Encryption algorithm with SPDF.

Require: The plaintext vector A and its size L, the pseudo-random sequence X, four generated vectors $tp1$, $tp2$, $tp3$ and $tp4$.
Ensure: Encrypted sequence C
1: $B(L) = mod(A(L) + X(1), 256)$
2: **for** $i = L - 1: -1: 1$ **do**
3: $cp = A(i)$
4: **if** $B(i+1) > =128$ **then**
5: $A(i) = A(tp1(i))$
6: $A(tp1(i)) = cp$
7: **else**
8: $A(i) = A(tp2(i))$
9: $A(tp2(i)) = cp$
10: **end if**

(Continued.)

$B(i) = mod(B(i+1) + A(i) + X(tp2(B(i+1)+1)+1), 256)$
end for
$C(1) = mod(B(1) + X(1), 256)$
for $i = 2: L$ **do**
$cp = B(i)$
if $C(i-1) >= 128$ **then**
$B(i) = B(tp3(i))$
$B(tp3(i)) = cp$
else
$B(i) = B(tp4(i))$
$B(tp4(i)) = cp$
end if
$C(i) = mod(C(i-1) + B(i) + X(tp4(C(i-1)+1)+1), 256)$
end for

5. The decryption alogorithm

Referring to figure 5, the decryption process is the inverse process of encryption. Initially, four keys are used to generate the pseudo-random sequences X. Then the encrypted image D is converted into a 1D vector C. After performing the reverse process of SPDF, which is detailed in algorithm 2, the plaintext sequence A is produced. Finally, the vector A is converted back into the original image.

Algorithm 2. Decryption algorithm of SPDF.

Require: The ciphertext vector C and its size L, the pseudo-random sequence X, four generated vectors $tp1$, $tp2$, $tp3$ and $tp4$, two vectors $node1$ and $node2$ from 1 and incrementing by 1 up to L.
Ensure: Plaintext sequence A.
$B(node1(1)) = mod(256*2 + C(1) - X(1), 256)$
for $i = 2: L$ **do**
$cp = node1(i)$
if $C(i-1) >= 128$ **then**
$node1(i) = node1(tp3(i))$
$node1(tp3(i)) = cp$
else
$node1(i) = node1(tp4(i))$
$node1(tp4(i)) = cp$
end if
$B(node1(i)) = mod(256*2 + C(i) - C(i-1) - X(tp4(C(i-1)+1)+1), 256)$
end for
$A(node2(L)) = mod(256*2 + B(L) - X(1), 256)$
for $i = L-1: -1: 1$ **do**
$cp = node2(i)$
if $B(i+1) >= 128$ **then**
$node2(i) = node2(tp1(i))$
$node2(tp1(i)) = cp$
else
$node2(i) = node2(tp2(i))$
$node2(tp2(i)) = cp$
end if
$A(node2(i)) = mod(256*2 + B(i) - B(i+1) - X(tp2(B(i+1)+1)+1), 256)$
end for

6. Simulation results and security analyses

In this section, the simulation results and security analyses of the proposed encryption scheme are presented. The simulations are conducted on a computer equipped with an AMD Ryzen 5 4600H CPU@ 3.00 GHz, 16.0 GB of RAM, Windows 10 OS, MATLAB R2022b. The secret keys are selected arbitrarily as $K_1 = 5$, $K_2 = 10$, $K_3 = 0.5$ and $K_4 = 0.7$. The medical images utilized in the experimental process are sourced from the Pseudo-PHI-DICOM-Database (cancerimagingarchive.net). Specifically, we select four CT images each with a size of 512512,

Table 3. Key space between different algorithms.

Schemes	Proposed	[9]	[23]	[25]	[20]
Key space size	10^{58}	10^{51}	10^{45}	10^{92}	10^{55}

from distinct body regions. These CT images are respectively labeled as '001', '002', '003' and '004' to facilitate clear identification in subsequent analyses.

6.1. Security key space

For an encryption algorithm with high security, its key space should be larger than 2^{100}. In the proposed cryptosystem, four secret keys are utilized, each within the range of $K_1 \in (2, 12)$, $K_2 \in (2, 12)$, $K_3 \in (0, 1)$, and $K_4 \in (0, 1)$, with a computational precision of 10^{14}. As a result, the total key space is approximately $10^{15} \times 10^{15} \times 10^{14} \times 10^{14} \approx 2^{192}$, which is significantly larger than 2^{100}. This ensures a great defense against all forms of brute-force attacks. Table 3 illustrates the comparison of key spaces between our algorithm and other encryption methods, and our method exhibits a large key space.

6.2. Histogram analysis

An effective encryption system should conceal the pixel distribution characteristics of the original image to prevent some attacks using statistical analysis. Figure 7 illustrates the histograms of both the plain and encrypted images. As shown in figure 7, the histogram of the cipher image is uniform, indicating its better performance in resisting statistical attack.

6.3. Correlation analysis

An ideal encrypted image should display a weak correlation between the adjacent pixels to prevent attackers from exploiting the correlations for attacking. Correlation analysis serves as a commonly used technique to quantify the correlation in different directions, and the correlation coefficient is defined as follows:

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)} \times \sqrt{D(y)}} \tag{24}$$

$$\mathrm{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{25}$$

where x and y are the pixel values of adjacent pixels and N is the total number of the image pixels. $E(x)$ and $E(y)$ are the expected values of x and y, respectively. $D(x)$ and $D(y)$ are the variances of x and y, respectively. Tables 4 and 5 present the correlation coefficient results of our algorithm and other algorithms, and our algorithm demonstrates a lower average correlation coefficient compared to other techniques. Also, the correlation plots of the 001 and 002 images, along with their corresponding cipher images, are presented in figure 8. The consequences of correlation analyses indicate that the strong correlation in the plain images is significantly weakened, approaching negligible correlation in the encrypted images.

6.4. Secret key sensitivity and plaintext sensitivity analysis

Secret key sensitivity and plaintext sensitivity analysis are essential parts of evaluating the security of image encryption algorithms. The sensitivity analysis aims to assess the impact of small changes in the secret key or plaintext on the cipher image. One common method to test sensitivity is Number of Pixel Change Rate (NPCR) and Unified Average Change Intensity (UACI). NPCR and UACI are defined as follows:

$$\begin{cases} NPCR = \sum_{i=1}^{M}\sum_{j=1}^{N} D(i, j) \times 100\% \\ UACI = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|C1(i, j) - C2(i, j)|}{255} \times 100\% \end{cases} \tag{26}$$

where $C1$, $C2$ are two cipher-images and $D(i, j) = \begin{cases} 0, & if\ C1(i, j) = C2(i, j) \\ 1, & if\ C1(i, j) \neq C2(i, j) \end{cases}$. M and N are the height and the width of the image, respectively.

Table 6 illustrates the key sensitivity results in the encryption process using the NPCR and UACI indicators. Meanwhile, figure 9 depicts the key sensitivity results in the decryption process. As observed in figure 9, the encrypted images decrypted by keys with slightly changed bear no resemblance to the original images, highlighting a high key sensitivity during decryption. Additionally, table 7 presents the results of the plaintext

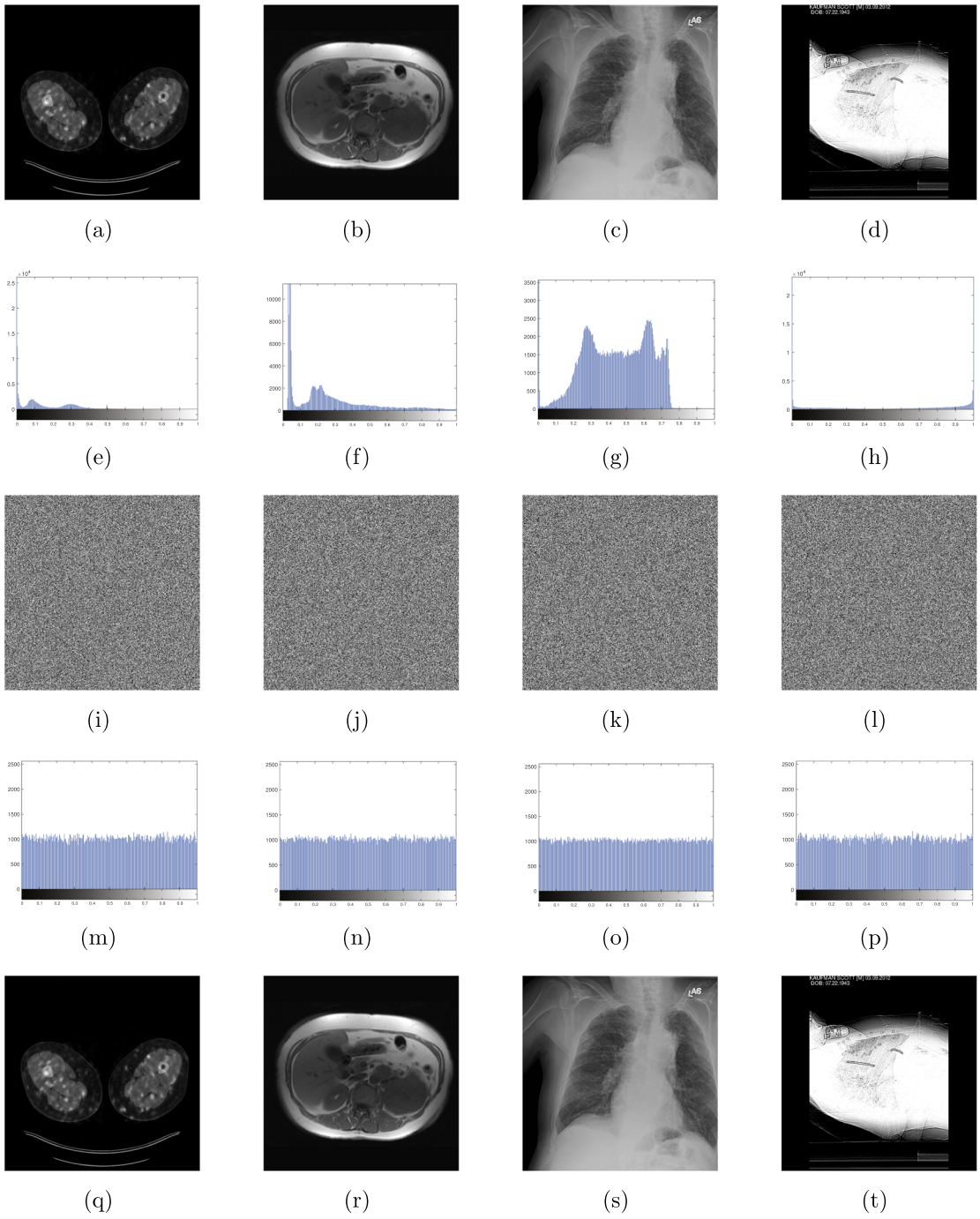


Figure 7. Simulation results. (a)–(d) original images; (e)–(h) the histograms of the original images; (i)–(l) the cipher images of (a)–(d), respectively; (m)–(p) the histograms of (i)–(l), respectively; (q)–(t) the decrypted images of (i)–(l), respectively.

Table 4. Correlation coefficients of original images and encrypted images.

Image	Original image			Cipher image		
	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal
001	0.9744	0.9932	0.9702	−0.0014	−0.0030	0.0036
002	0.9955	0.9981	0.9538	−0.0001	0.0033	0.0009
003	0.9956	0.9937	0.9926	0.0013	0.0010	0.0043
004	0.9768	0.9758	0.9650	0.0004	0.0025	−0.0011

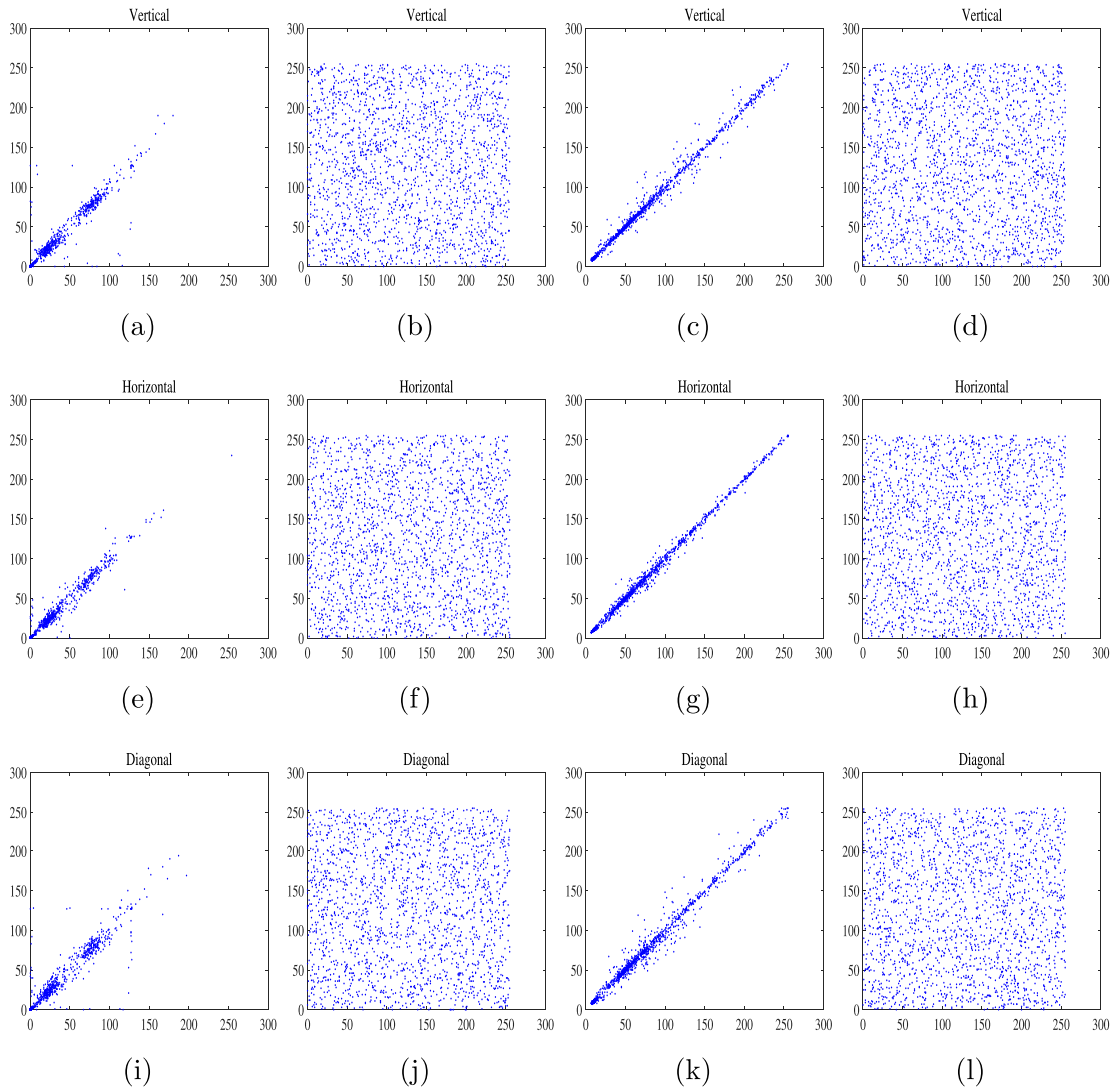


Figure 8. Simulation results. (a) , (e) , (i) are the correlation plots of the original image of 001; (b) , (f) , (j) are the correlation plots of the encrypted image of 001; (c) , (g) , (k) are the correlation plots of the original image of 002 ; (d) , (h) , (l) are the correlation plots of the encrypted image of 002.

Table 5. Correlation coefficients of encrypted image 001 obtained by different algorithms.

Direction	Proposed	[9]	[23]	[25]	[20]
Vertical	−0.0014	0.0008	0.0001	0.0054	0.0091
Horizontal	0.0030	0.0055	0.0037	0.0022	0.0032
Diagonal	0.0036	−0.0036	0.0087	−0.0069	−0.0024
Average	0.0026	0.0033	0.0041	0.0048	0.0049

sensitivity of different algorithms. Our algorithm's results are closely aligned with the theoretical values of NPCR and UACI, indicating a high plaintext sensitivity.

6.5. Information entropy analysis

Information entropy $H(m)$ of an image reflects the randomness of its pixel values, and a higher entropy value indicates a more secure encryption.

$$H(m) = \sum_{i=1}^{256} p(m_i) \log \frac{1}{p(m_i)} \tag{27}$$

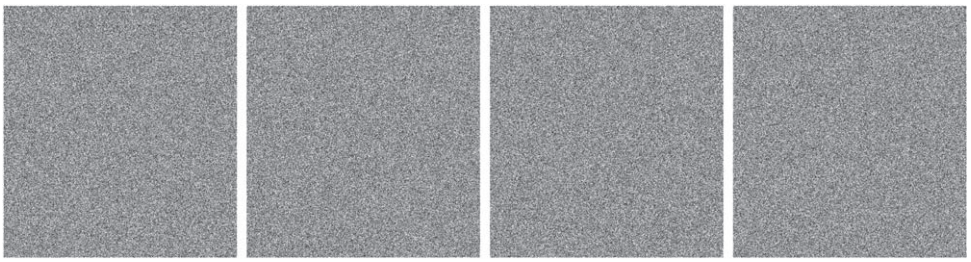


Figure 9. Simulation results. (a) decrypted image with the key ($K_1 + 10^{-14}, K_2, K_3, K_4$); (b) decrypted image with the key (K_1, $K_2 + 10^{-14}, K_3, K_4$); (c) decrypted image with the key ($K_1, K_2, K_3 + 10^{-14}, K_4$); (d) decrypted image with the key (K_1, K_2, K_3, $K_4 + 10^{-14}$).

Table 6. Key sensitivity analysis results of the NPCR and UACI indicators in encryption process.

Image	Index	$K_1 + 10^{-14}$	$K_2 + 10^{-14}$	$K_3 + 10^{-14}$	$K_4 + 10^{-14}$
001	NPCR	99.6094	99.6102	99.6114	99.6086
	UACI	33.4627	33.4698	33.4592	33.4609
002	NPCR	99.6083	99.6098	99.6091	99.6106
	UACI	33.4686	33.4601	33.4698	33.4638
003	NPCR	99.6057	99.6090	99.6115	99.6095
	UACI	33.4623	33.4552	33.4585	33.4702
004	NPCR	99.6075	99.6098	99.6101	99.6106
	UACI	33.4616	33.4620	33.4568	33.4679

Table 7. Plaintext sensitivity analysis results of the NPCR and UACI indicators.

Image	Size	NPCR(99.6094)	UACI(33.4635)
001	512 × 512	99.5789	33.4695
002	512 × 512	99.5795	33.4618
003	512 × 512	99.5826	33.4782
004	512 × 512	99.5740	33.4672
001	256 × 256	99.5385	33.4248
002	256 × 256	99.5467	33.4334
001 in [9]	512 × 512	99.6194	33.4656
001 in [23]	512 × 512	99.7393	33.4064
001 in [25]	512 × 512	99.6293	33.5078
001 in [20]	512 × 512	99.6014	33.4771
001 in [9]	256 × 256	99.6293	33.4356
001 in [23]	256 × 256	99.7504	33.4135
001 in [25]	256 × 256	99.6470	33.4824
001 in [20]	256 × 256	99.5813	33.4827

In equation (27), $m(i)$ represents the probability of different gray-level values. For an encrypted image with 256 gray levels, the theoretical value of $H(m)$ is expected to be 8. Table 8 presents the information entropy results of different plain images and the cipher images encrypted with the proposed scheme and other algorithms. The information entropy of the images encrypted by our proposed scheme closely approaches the theoretical value of 8, indicating the cryptosystem is resistant to entropy attack.

6.6. Chosen-plaintext attack analysis

Chosen-plaintext attack (CPA) is a common method used by attackers in cryptanalysis. In this method, attackers can obtain any plaintexts and its corresponding ciphertexts. They deliberately choose or craft particular plaintexts, such as images entirely in black or white, in an attempt to analyze the relationship between the plaintexts and ciphertexts or deduce the secret key. In SPDF, the target positions of pixels in permutation stage

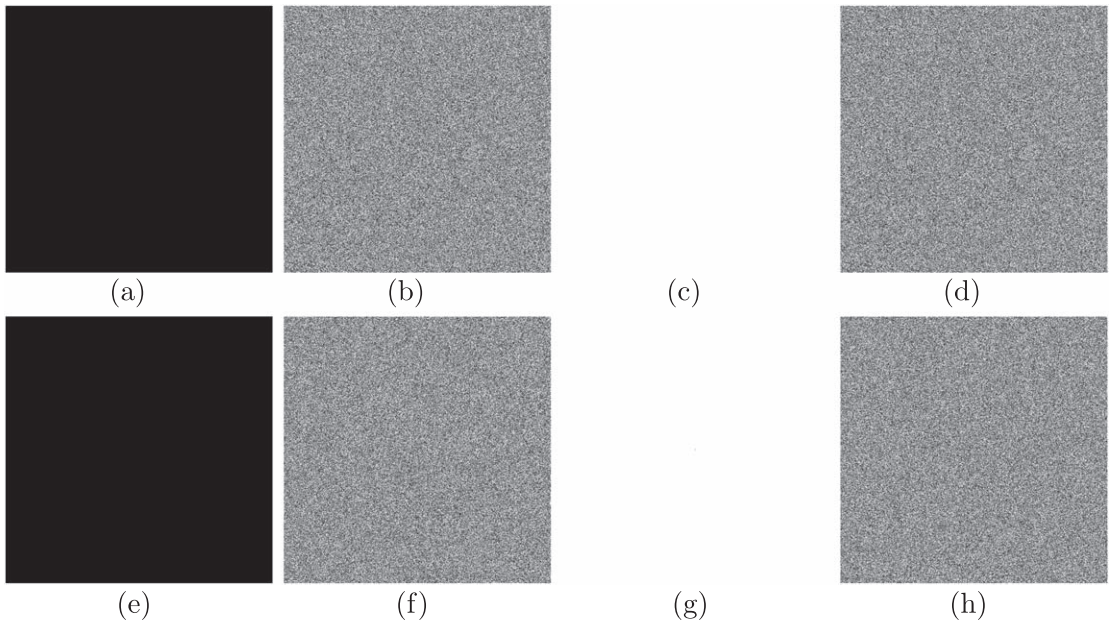


Figure 10. Simulation results of special images. (a) P1; (b) encrypted image of (a); (c) P2; (d) encrypted image of (c); (e) P3; (f) encrypted image of (e); (g) P4; (h) encrypted image of (g).

Table 8. Information entropy analysis results.

Image	Plain image	Cipher image
001	3.2696	7.9992
002	5.8698	7.9993
003	7.2993	7.9993
004	3.6129	7.9992
001 in [9]	3.2696	7.9992
001 in [23]	3.2696	7.9991
001 in [25]	3.2696	7.9993
001 in [20]	3.2696	7.9995

Table 9. The NPCR and UACI results of special images.

Image	NPCR(99.6094)	UACI(33.4635)
Encrypted image of P1	99.5922	33.4529
Encrypted image of P2	99.5876	33.4639
Encrypted image of P3	99.5944	33.4641
Encrypted image of P4	99.6073	33.4646

and diffusion results relies on the previous diffused pixel, plaintext vector and the chaotic sequence. Consequently, even a minor alteration in the plaintext will exert a substantial influence on the encryption result. This feature makes the proposed encryption method difficult for attackers to use specific plaintexts to extract useful information.

Four particular images are designed for this trial, labeled as P1, P2, P3 and P4 with the same size of 512×512. P1 and P2 represent the images entirely in black and white, respectively. P3 is an almost all-black image, in which only one pixel value of 1 in its center location (256,256), and the other pixel values are 0. Similarly, P4 is an almost all-white image with only one pixel value of 0 in center location. The encrypted images are depicted in figure 10, and their corresponding NPCR and UACI results are presented in table 9. The experimental results indicate that the encrypted images of the specific images do not reveal any valuable information for analysis to resist CPA.

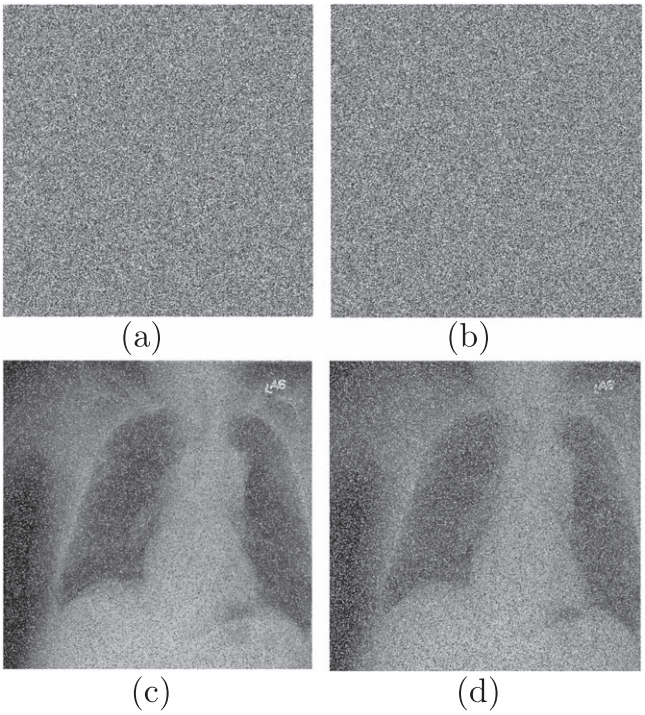


Figure 11. Results of noise resistance: (a) the encrypted image 003 contaminated by salt and pepper noise 0.05 density; (b) the encrypted image 003 contaminated by salt and pepper noise 0.1 density; (c)–(d) the decrypted images of (a)–(b) respectively.

6.7. Noise attack and data loss analysis

Noise attack and data loss are well-known concerns during the transmission of cipher data across networks. To measure the robustness of our proposed encryption algorithm against these challenges, we select image 003 with a size of 512 × 512. In the noise attack trial, the ciphertext images are polluted by salt-and-pepper noise with a noise intensity level of 0.05 and 0.1, respectively. The resulting noisy images and decrypted images are shown in figure 11. In the data loss trial, four cipher images are intentionally obscured with different degrees and regions, and the decrypted images are depicted in figure 12.

Furthermore, noise attack and data loss analysis of the cryptosystem can be quantified by peak-signal-to-noise (*PSNR*), structural similarity (*SSIM*) and 2D correlation coefficients (*CC*), and their mathematical representations are provided in equation (28), equation (29), and equation (30), respectively.

$$PSNR = 10\lg\frac{255 \times 255}{\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(A(i,j) - B(i,j))^2} \tag{28}$$

$$SSIM(A, B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)} \tag{29}$$

$$CC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(A(i,j) - \mu_A)(B(i,j) - \mu_B)}{\sqrt{\left(\sum_{i=1}^{M}\sum_{j=1}^{N}(A(i,j) - \mu_A)^2\right)\left(\sum_{i=1}^{M}\sum_{j=1}^{N}(B(i,j) - \mu_B)^2\right)}} \tag{30}$$

where A is the plain-image and B is the cipher-image, M and N are the size of A. μ_A and μ_B are the mean values of A and B, respectively. σ_A^2 is the variance of A, σ_B^2 is the variance of B and σ_{AB} is the covariance of A and B. C_1 and C_2 are small constants. Table 10 presents the *PSNR*, *SSIM* and *CC* analysis results between different algorithms. From figures 10, 12 and table 10, most of the vital information in the original images can still retain from the decrypted images. It demonstrates that the encryption system has better recovery capability for the polluted information and resists noise attacks and data loss effectively.

6.8. Encryption time analysis

An excellent cryptosystem should have low computational complexity and short processing time. The encryption speed simulation of our scheme and other algorithms is performed in the same environment. The simulation results of encryption speed are shown in table 11. Notably, for our proposed algorithm, the time consumption for encrypting the image 001 with a size of 512 × 512 is 0.0612 s, which is significantly lower than the ones of other algorithms in table 12.

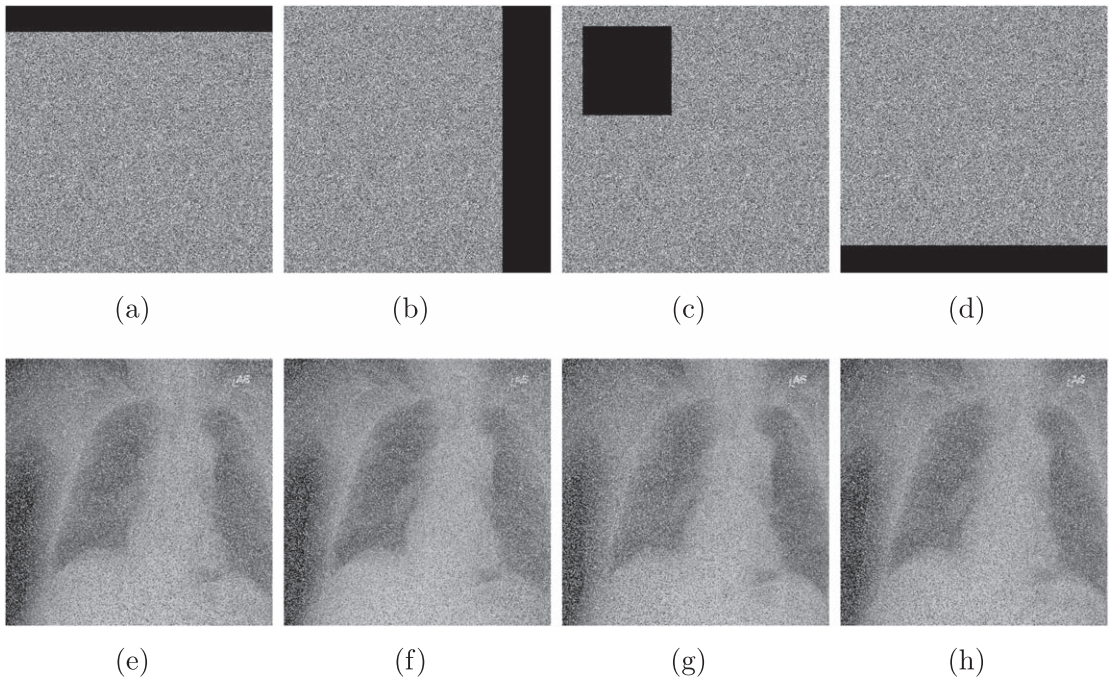


Figure 12. Simulation results of data loss: (a)–(d) are the encrypted images 003 with different degrees of data loss ; (e)–(h) are the corresponding decrypted images respectively.

Table 10. Noise attack and data loss analysis using *PSNR*, *SSIM* and *CC* indicator.

Noise attacks or data loss	Type	Proposed	[23]	[25]	[9]	[20]
Salt and pepper noise(0.05)	*PSNR*	15.0253	12.8016	14.7574	19.8983	16.0863
Salt and pepper noise(0.05)	*SSIM*	0.0308	0.0264	0.0289	0.2099	0.0300
Salt and pepper noise(0.05)	*CC*	0.6277	0.3898	0.5277	0.6188	0.6918
Salt and pepper noise(0.1)	*PSNR*	13.6476	11.5770	13.4095	17.0824	13.5180
Salt and pepper noise(0.1)	*SSIM*	0.0210	0.0152	0.0122	0.1035	0.0189
Salt and pepper noise(0.1)	*CC*	0.4864	0.3321	0.4353	0.5452	0.4776
Data loss(1:50,1:512)	*PSNR*	14.0441	12.0949	12.7425	13.1952	7.9183
Data loss(1:50,1:512)	*SSIM*	0.1790	0.0177	0.1046	0.1156	0.0001
Data loss(1:50,1:512)	*CC*	0.7938	0.3833	0.5010	0.6495	0.0154
Data loss(1:512,420:512)	*PSNR*	13.9627	10.6747	13.8106	13.4483	13.4777
Data loss(1:512,420:512)	*SSIM*	0.1752	0.0062	0.0958	0.0520	0.1533
Data loss(1:512,420:512)	*CC*	0.7143	0.2379	0.5745	0.6319	0.7319
Data loss(40:210,40:210)	*PSNR*	14.1047	12.1108	13.9063	9.0518	15.7865
Data loss(40:210,40:210)	*SSIM*	0.1835	0.0215	0.1236	0.0000	0.1494
Data loss(40:210,40:210)	*CC*	0.7714	0.3981	0.7540	0.0014	0.8375
Data loss(460:512,1:512)	*PSNR*	14.1560	12.0562	12.2368	13.8779	13.7664
Data loss(460:512,1:512)	*SSIM*	0.1692	0.0228	0.1065	0.1104	0.1316
Data loss(460:512,1:512)	*CC*	0.8348	0.3657	0.6311	0.7378	0.7218

Table 11. Encryption time (seconds).

Image	001	002	003	004
Time	0.0612	0.0625	0.0640	0.0636

Table 12. The comparison of encryption time (seconds).

Image	Proposed	[9]	[23]	[25]	[20]
001	0.0612	0.1744	0.3735	0.2438	2.6365

7. Conclusion

In this paper, a new 1D chaotic map 1D-CICMIC is presented to possess better ergodicity and unpredictability. Then, we improve the traditional Josephus traversing by a dynamic scrambling method, aiming to enhance the permutation effect. What's more, we introduce a simultaneous permutation and diffusion framework called SPDF to improve the security of the conventional scrambling-diffusion structure. Finally, based on the 1D-CICMIC and SPDF, we develop a medical image encryption system that adopts the plaintext correlation in the diffusion phases to strengthen the plaintext sensitivity. Extensive experiments have been conducted to evaluate the proposed encryption scheme, demonstrating its high security and high computational efficiency in resisting various attacks. Despite the notable performance in most security experiments, the robustness of the proposed scheme falls short, particularly in the face of the noise attack. Therefore, future work will focus on enhancing the robustness of the encryption system and exploring its potential application in other areas such as the video encryption in telemedicine.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

Ethical approval

Not applicable.

Competing interests

The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, nor in the decision to publish the results.

Authors' contributions

Conceptualization, Zhen Le and Quanjun Li; methodology, Zhen Le and Linqing Huang; software, Zhen Le and Quanjun Li; validation, Huang Chen, Shuting Cai and Xiaoming Xiong; investigation, Linqing Huang; writing original draft preparation, Zhen Le and Quanjun Li; writing review and editing, Linqing Huang; funding acquisition Shuting Cai. All authors have read and agreed to the published version of the manuscript.

Funding

This work was funded by the Key Area R & D Program of Guangdong Province under Grant 2022B0701180001, the National Natural Science Foundation of China (61801127), the Science Technology Planning Project of Guangdong Province, China (No.2019B010140002, No.2020B111110002), and the Guangdong-Hong Kong-Macao Joint Innovation Field Project (No.2021A0505080006).

ORCID iDs

Linqing Huang https://orcid.org/0000-0001-9636-499X

References

[1] Chen J, Chen L, Zhang L Y and Zhu Z l 2019 *Nonlinear Dyn.* **96** 301–22
[2] Chen Y, Tang C and Ye R 2020 *Signal Process.* **167** 107286
[3] Kumar M and Gupta P 2021 *Multimedia Tools Appl.* **80** 18941–67
[4] Yasser I, Khalil A T, Mohamed M A, Samra A S and Khalifa F 2021 *IEEE Access* **10** 244–57
[5] Castro F, Impedovo D and Pirlo G 2023 *Applied Sciences* **13** 6099
[6] Fridrich J 1998 *Int. J. Bifurcation Chaos* **8** 1259–84
[7] Kumar M, Iqbal A and Kumar P 2016 *Signal Process.* **125** 187–202

[8] Xu M and Tian Z 2019 *Inf. Sci.* **478** 1–14
[9] Guan Z, Li J, Huang L, Xiong X, Liu Y and Cai S 2022 *Entropy* **24** 384
[10] Li Z, Peng C, Tan W and Li L 2021 *Sensors* **21** 758
[11] Chai X, Zheng X, Gan Z and Chen Y 2020 *Neural Computing and Applications* **32** 8065–88
[12] Sambas A *et al* 2020 *IEEE Access* **8** 137116137116–32
[13] Xie H w, Gao Y j and Zhang H 2022 *Multimedia Tools Appl.* **82** 16431–53
[14] Alkhayyat A, Ahmad M, Tsafack N, Tanveer M, Jiang D and Abd El-Latif A A 2022 *Journal of Signal Processing Systems* **94** 315–28
[15] Zhu S, Deng X, Zhang W and Zhu C 2023 *Math. Comput. Simul.* **207** 322–46
[16] Zhang Y 2020 *Inf. Sci.* **520** 177–94
[17] Chai X, Gan Z, Yuan K, Chen Y and Liu X 2019 *Neural Computing and Applications* **31** 219–37
[18] Kang Y, Huang L, He Y, Xiong X, Cai S and Zhang H 2020 *Symmetry* **12** 1393
[19] Huang L, Li W, Xiong X, Yu R, Wang Q and Cai S 2022 *Opt. Commun.* **517** 128365
[20] Huang L, Cai S, Xiao M and Xiong X 2018 *Entropy* **20** 535
[21] Midoun M A, Wang X and Talhaoui M Z 2021 *Opt. Lasers Eng.* **139** 106485
[22] Pak C and Huang L 2017 *Signal Process.* **138** 129–37
[23] Chen B, Huang L, Cai S, Xiongb X and Zhang H 2023 *Chin. Phys.* B **33** 030501
[24] Lawnik M, Moysis L, Baptista M S and Volos C 2024 *Nonlinear Dyn.* **112** 6679–93
[25] Huang S, Jiang D, Wang Q, Guo M, Huang L, Li W and Cai S 2022 *Chaos, Solitons Fractals* **163** 112584
[26] Tong X J, Zhang M, Wang Z, Liu Y, Xu H and Ma J 2015 *J. Visual Commun. Image Represent.* **33** 219–34
[27] Wang H, Xiao D, Chen X and Huang H 2018 *Signal Process.* **144** 444–52
[28] Huang Y B, Xie P W, Gao J B and Zhang Q Y 2023 *Int. J. Bifurcation Chaos* **33** 2350096
[29] Akhavan A, Samsudin A and Akhshani A 2017 *Opt. Laser Technol.* **95** 94–9
[30] Lin C Y and Wu J L 2020 *Entropy 22 589* **22** 589
[31] Huang L, Cai S, Xiong X and Xiao M 2019 *Opt. Lasers Eng.* **115** 7–20
[32] Bezerra J I M, Machado G, Molter A, Soares R I and Camargo V 2023 *Chaos, Solitons Fractals* **168** 113160
[33] Teng L, Wang X and Xian Y 2022 *Inf. Sci.* **605** 71–85
[34] Alawida M, Teh J S, Mehmood A, Shoufan A *et al* 2022 *Journal of King Saud University-Computer and Information Sciences* **34** 8136–51
[35] Cencini M and Ginelli F 2013 *J. Phys. A: Math. Theor.* **46** 250301
[36] He D, He C, Jiang L G, Zhu H w and Hu G r 2001 *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* **48** 900–6
[37] Richman J S and Moorman J R 2000 *American Journal of Physiology-Heart and Circulatory Physiology* **278** H2039–49
[38] Gottwald G A and Melbourne I 2004 *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **460** 603–11