Research Article

An Image Encryption Scheme Based on Lorenz Hyperchaotic System and RSA Algorithm

Riguang Lin and Sheng Li

Faculty of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang 524088, China

Correspondence should be addressed to Sheng Li; lish_ls@sina.com

Received 27 January 2021; Revised 31 March 2021; Accepted 7 April 2021; Published 21 April 2021

Academic Editor: Zhiyuan Tan

Copyright ©2021 Riguang Lin and Sheng Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This research proposes a new image encryption scheme based on Lorenz hyperchaotic system and Rivest–Shamir–Adleman (RSA) algorithm. Firstly, the initial values of the Lorenz hyperchaotic system are generated by RSA algorithm, and the key stream is produced iteratively. In order to change the position and gray value of the pixel, the image data are hidden by additive mode diffusion. Secondly, the diffusion image matrix is reshaped into a one-dimensional image matrix, which is confused without repetition to hide the image data again. Then, the finite field diffusion algorithm is executed to realize the third hiding of the image information. In order to diffuse the pixel information into the entire cipher image, the additive mode diffusion algorithm needs to be looped twice. Finally, the cipher image can be obtained. The experimental results prove that the image encryption scheme proposed in this research is effective and has strong antiattack and key sensitivity. Moreover, the security of this encryption scheme relies on the RSA algorithm, which has high security.

1. Introduction

With the rapid development and popularization of Internet technology, multimedia has become an important means of communication for people. Digital images, as a multimedia resource, are widely used in information communication because they can carry a large amount of information and express the information content intuitively and vividly. In the current big data era, digital images are widely used in various fields, such as business, education, medical research, aerospace, military, and politics. In order to share image information, we can easily transmit it on the Internet by computer or mobile equipment. However, people who are not authorized can also easily obtain the images, and the unauthorized cryptanalysis is a great threat to the information communication of images [1, 2]. More importantly, some images may involve national security and personal privacy, such as satellite reconnaissance or biometric passports. Therefore, the question of how to efficiently protect the security of image communication has attracted great attention from scholars and experts all over the world [2, 3]. Image data are generally different from text data. Classical cryptography provides good encryption algorithms and decryption algorithms for one-dimensional text data, such as DES (Data Encryption Standard) and AES (Advanced Encryption Standard). However, due to the characteristics of digital images, such as large data volume and strong correlation between adjacent pixels, traditional ciphers are not suitable for encrypting images [2].

Accordingly, image encryption algorithms based on technologies such as DNA random coding [4], double random phase coding [5, 6], Arnold transform [7], and chaotic systems [8–10] have been proposed. The chaotic system has superior performance in the field of digital image encryption, which may be attributed to the basic characteristics of the chaotic system, such as sensitivity to initial conditions, pseudorandomness, nonlinearity, and non-periodicity. As early as 1998, Fridrich [11] proposed a symmetric encryption scheme based on a two-dimensional chaotic map. In this scheme, permutation-diffusion architecture was suggested to encrypt image content, where the permutation operation is performed to alter the positions of
the image pixels, while the diffusion operation is performed to change gray values. Permutation-diffusion mechanism has been widely studied and used in chaotic encryption systems. For instance, a hyperchaos-based image encryption algorithm using pixel-level permutation and bit-level permutation was presented by Li et al. [12], which adopted a 5D multiwing hyperchaotic system, and the key stream generated by hyperchaotic system is related to the original. Ye et al. [2] proposed an efficient pixel-level image encryption algorithm that enhanced the connection between position shuffling for pixels and the change to gray values as compared to the traditional permutation-diffusion architecture. In [13], a novel approach that uses a hyperchaotic system, pixel-level filtering with kernels of variable shapes and parameters, and DNA-level diffusion was designed for image encryption. First, a hyperchaotic system is applied to generating hyperchaotic sequences. Second, dynamic filtering is performed on pixels to change the pixel values. Third, a global bit-level scrambling is conducted to change the values and positions of pixels. Finally, a novel DNA-level diffusion scheme is proposed to further change the image values simultaneously. A new permutation based on bit [14] was implemented into an image encryption algorithm. It is noted that the algorithm [14] employs a "Rubik's cube for bit permutation strategy" to replace the traditional permutation operation. In the diffusion stage, the chaotic map is iterated and bit streams are generated which are then used for diffusion and again chaotic map is used for confusion at pixel level. Unlike the above methods that mainly encrypt grayscale image, a new color image encryption algorithm was proposed in [15] with a new revised one-dimensional chaotic map. Compared with the traditional one-dimensional chaotic map, the revised one-dimensional chaotic map exhibits better chaotic performances and larger chaotic ranges. Firstly, the method [15] reshapes a color image matrix of size $M \times N$ into an image vector $P$, with length $3MN$. Then, it produces a permutation position matrix, $X'$, from chaotic sequence $X$ to shuffle pixel positions for $P$ and obtain a permuted image, $P'$. After that, a diffusion operation for $P'$, using a diffusion matrix $D'$ from $X$, is taken to achieve $C$. A rotating function is applied to $C$ in order to get $C'$. Finally, the cipher image is formed after reshaping $C'$ into a $R$, $G$, and $B$ color image.

There are also many other image encryption algorithms [16–23] that have been proposed to protect image information. For example, a parallel image encryption method based on compressive sensing was proposed in [16]. Memristive chaotic system, elementary cellular automata (ECA), and compressive sensing (CS) were designed for image encryption in [17]. To resist the chosen/plaintext attacks, plaintext-related shuffling was designed for image encryption in [19]; the algorithm mainly includes two plaintext-unrelated diffusion operations and one plaintext-related shuffling. Moreover, generalized Arnold transform and double random phase encoding were introduced for quantum image encryption [20]. In [21], dynamic DNA encryption was used for color image cryptosystem. In order to improve security needs of image content, the method [23] of encryption scheme combines the techniques of chaotic image encryption and DNA (deoxyribonucleic acid) sequence operations. In [24, 25], matrix semitensor product and Boolean network are used in the encryption scheme. First, the pixels of the initial plaintext image are randomly divided into four blocks. The pixels in each block are then subjected to different numbers of rounds of Arnold transformation, and the four blocks are combined to generate a scrambled image. Then, a set of pseudosecret keys is given and filtered through a synchronously updating Boolean network to generate the real secret key, which is used as the initial value to generate a chaotic sequence. Finally, the matrix semitensor product (STP) operation is applied to the chaotic sequences and the scrambled image to generate an encrypted image. Compared with other encryption algorithms, the algorithm is more secure and effective, and it is also suitable for color image encryption. In [26], the fractal sorting matrix is irregular, self-similar, and infinitely iterative. And the scrambling images or information based on this new cluster of matrices can effectively improve encryption algorithm security. In addition, the data in the antiferral attack test are closer to the theoretical values and smaller in data fluctuation. Therefore, the proposed algorithm shows better security and resistance to various attacks. An encrypted coverless information hiding method that transfers secret images between two different image domains using generative models was proposed in [27]. In the encryption stage, firstly, a secret image was embedded into a public image (one domain) to obtain a synthetic image; then, the image was utilized as the input to the first generative model $F$ to obtain an encrypted image (another domain). Adversarial loss and an extraction module are added to improve the quality of the encrypted images generated in this stage. In the decryption stage, a second generative model $G$ was designed to reconstruct the synthetic images from the encrypted images. Finally, the secret image is separated from the reconstructed synthetic image.

However, the methods mentioned above are types of symmetric cryptosystems, where encryption and decryption use the same key. This may cause problems related to key management [28] and image information leakage [1]. In order to overcome the shortcomings of symmetric cryptography in key management, many asymmetric encryption algorithms have been proposed [29–37]. For example, single-channel color image encryption based on asymmetric cryptosystem was proposed in [33]. Firstly, the color components, respectively, multiplied with three random phase encryption keys were combined into one gray image using convolution. Then, the gray image was encoded into a real-value gray ciphertext using the asymmetric cryptosystem. Moreover, the decryption key is generated during encryption process and is different from the encryption key. In [34], Hartley transform and gyrator transform were implemented into single-channel color image encryption with asymmetric cryptosystem. Due to the nonlinear operation of phase truncation, a one-way encryption scheme could be achieved and thus high robustness against existing attacks could be obtained. In addition, transformation angle of $G$ offers remarkably sensitive key, and thus the security of the system is greatly enhanced. A double-image
encryption method based on an asymmetric algorithm is proposed in [35]. The encryption process of the method [35] was different from the decryption, and the encrypting keys were also different from the decrypting keys. During the nonlinear encryption process, the images are encoded as amplitude ciphertext, and two phase-only masks generated based on phase truncation are retained as the decryption key. Chen et al. [36] proposed an enhanced asymmetric cryptosystem for color image, which uses equal modulus decomposition (EMD) in the gyrator transform domains, and created an effective one-way trapdoor function through EMD. To improve the security of the cryptosystem, the red-green-blue (RGB) components of color images were confused by using a Baker map. Rakheja et al. [37] proposed an asymmetric hybrid cipher scheme using a four-dimensional hyperchaotic structure by means of coherent superposition and random decomposition in hybrid multiresolution wavelet domain. The four-dimensional hyperchaotic framework’s parameters and preliminary conditions together with the fractional order expand the key space and consequently give additional strength to the system.

The RSA encryption algorithm is a type of public-key cryptography. It has two different keys, one of which is a public key and the other a private key. The security of RSA is based on the difficulty of decomposing large integers into two prime factors. Consequently, the RSA algorithm is widely studied and applied in the field of image encryption [38, 39]. For example, in order to enhance the strength of the cryptosystem and provide higher security, a chaotic synchronization cryptosystem combined with RSA encryption algorithm was proposed in [40]. The scheme [40] uses the RSA algorithm to encrypt the plain image to produce a cipher image. To achieve double encryption, the cipher image is reencrypted by using chaotic synchronization.

2. Related Works

In recent years, digital image encryption algorithms based on asymmetric cryptosystems have attracted the attention of experts and scholars. More and more image encryption algorithms based on asymmetric cryptosystems have been proposed. For example, Liu et al. [41] proposed a digital image watermarking model based on the scrambling algorithm logistic and RSA asymmetric encryption algorithm to ensure the security of hidden data based on a large embedding amount, strong robustness, and high computing efficiency. The system [41] involved applying the encryption algorithms of logistic and RSA to the watermark image and performing the hybrid decomposition of Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) on the host image, and the watermark was embedded into the low-frequency subband of the host. In [42], elliptic curve pseudorandom and Advanced Encryption System were introduced for image encryption scheme. The proposed scheme uses elliptic curve random generator to produce a sequence of arbitrary numbers based on curves; then, the Advanced Encryption System is applied to the sequence to acquire arbitrary keys for encrypting image. Jiao et al. [43] proposed an image encryption scheme based on a generalized Arnold map and RSA algorithm. In the scheme, RSA algorithm is used to generate the parameters of the generalized Arnold map. Then, the generalized Arnold map together with permutation and diffusion algorithms is performed on original image to obtain the cipher image. To reduce the burden of transmission, a scheme based on compressive sensing and RSA algorithm was proposed in [44], where the optical compressive image system is used to sample original image. During the process of encryption, the Walsh–Hadamard transform and a measurement matrix are designed for measuring the original image in order to reduce the redundant information in the original image. Then, a pseudorandom sequence generated by one-dimensional cascade chaotic system and deoxyribonucleic acid (DNA) sequence operations are utilized to change the pixel values. Finally, the cipher image can be obtained.

In order to efficiently and safely handle large volumes of encrypted data, chirp z-transform (CZT) was implemented into the asymmetric multi-image encryption system proposed in [45]. Since the system is asymmetric, decryption keys are different from encryption keys, which improves the system’s ability to resist cryptanalysis attacks. To resist chosen-plaintext and chosen-ciphertext attacks, Henon Map, Dynamic S-Boxes, and Elliptic Curve Cryptography were introduced for an image encryption scheme [46], where encryption keys are produced by elliptic curve cryptography (ECC). Moreover, a hash verification step was utilized in decryption process to resist chosen-plaintext attacks. In order to improve the security of image transmission, RSA cryptosystem, reality preserving fractional discrete cosine transform, and Arnold transform were implemented into a novel approach [47] for security of multiple-color images. In the process of encryption, firstly, color images are divided into red, green, and blue components. Then, each component is encrypted by RSA cryptosystem. Moreover, the fractional discrete cosine transform is performed on the partially encrypted image. Finally, the cipher image can be obtained after Arnold transform dislocating the processed image. In [48], the scheme combined the advantages of hyperchaotic map and RSA algorithm. Firstly, encrypt the plain image into a cipher image by utilizing the row and column encryption algorithms. Second, the secret key of image is converted into ciphertext by using the RSA algorithm; then the ciphertext is transformed into an image and permuted by using the Arnold map to obtain the key cipher image. Finally, to obtain a visually secure image, the key cipher image and the cipher image are embedded into a carrier image.

Based on the analysis above, this paper combines Lorenz hyperchaotic system and RSA algorithm to solve the problem of key exchange by using permutation and finite field diffusion algorithms.

3. Preliminaries

3.1. RSA Algorithm. RSA is an asymmetric encryption algorithm. In 1978, three mathematicians, Rivest, Shamir, and Adleman, proposed the RSA encryption algorithm for the first time. Unlike the symmetric cryptosystem, RSA
algorithm has two different keys, one of which is a public key and the other is a private key. If the data are encrypted with a public key, only the corresponding private key can be used for decryption; if the data are encrypted with a private key, then only the corresponding public key can be used for decryption. Because encryption and decryption use two different keys, this algorithm is called an asymmetric encryption algorithm. Currently, RSA is the most widely used asymmetric encryption algorithm. It is based on Euler theorem in number theory, and its security depends on the difficulty of factoring large integers. RSA can be used to encrypt information and can also be used as a digital signature. The encryption and decryption process of the RSA algorithm is shown in Algorithm 1.

This algorithm can encrypt plaintext data by using the recipient’s public key. After sending the ciphertext to the receiver, the receiver uses its private key to decrypt the ciphertext to obtain the plaintext data. Only the receiver knows the private key, which reduces the transmission of the key in the channel and improves the security of communication.

3.2. Lorenz Hyperchaotic System. The Lorenz hyperchaotic system is a classic chaotic system, which is defined as follows:

\[
\begin{align*}
\frac{dx}{dt} &= a(y - x) + w, \\
\frac{dy}{dt} &= cx - y - xz, \\
\frac{dz}{dt} &= xy - bz, \\
\frac{dw}{dt} &= -yz + rw.
\end{align*}
\] (1)

A hyperchaotic system needs to satisfy the following conditions. Firstly, it has a phase space of dimension at least four. Secondly, it possesses at least two positive Lyapunov exponents [49].

In equation (1), \( r \) is the control parameter. Let \( a = 10, b = 8/3, c = 28 \); when \(-1.52 \leq r \leq -0.06\), the system exhibits hyperchaotic behavior. According to the method proposed by Ramasubramanian et al. [50], when \( r = -1 \), the four Lyapunov exponents [51] can be obtained as follows: \( \lambda_1 = 0.3381, \lambda_2 = 0.1586, \lambda_3 = 0, \lambda_4 = -15.1752 \); obviously, the Lorenz hyperchaotic system exhibits hyperchaotic behavior. The fourth-order Runge–Kutta method is usually used to discretize continuous chaotic systems such as Lorenz hyperchaotic systems. For example, the first-order differential equation is defined as follows:

\[
\frac{dy}{dx} = f(x, y), \quad x \in [a, b].
\] (2)

The discrete form of Runge–Kutta method for equation (2) is defined as:

\[
\begin{align*}
y_{i+1} &= y_i + \frac{h}{6} \left( K_1 + 2K_2 + 2K_3 + K_4 \right), \\
K_1 &= f(x_i, y_i), \\
K_2 &= f \left( x_i + \frac{h}{2}, y_i + \frac{h}{2}K_1 \right), \\
K_3 &= f \left( x_i + \frac{h}{2}, y_i + \frac{h}{2}K_2 \right), \\
K_4 &= f \left( x_i + h, y_i + hK_3 \right).
\end{align*}
\] (3)

The projections of Lorenz hyperchaotic system attractor drawn by equations (1) and (3) are shown in Figure 1. And the initial value range of Lorenz hyperchaotic system is as follows: \( x_0 \in (-40, 40), y_0 \in (-40, 40), z_0 \in (1, 81) \), and \( w_0 \in (-250, 250) \). Figure 1 shows the six strange attractors of the Lorenz hyperchaotic system, where the phase diagram of \( x_n - z_n \) is butterfly-like and is known as the butterfly attractor.

3.3. Chaotic Sequence Generator. The state values of chaotic systems are floating-point numbers, and the sequences composed of the state values of chaotic systems cannot be directly applied to image cryptosystems. In general, for images with a gray level of \( L \), the chaotic state value needs to be converted to an integer of \( 0 \sim L - 1 \). The transformed sequence can be applied to image cryptosystem, which is called chaotic pseudorandom sequence. Two methods of converting chaotic state values into integers are used in this study, as shown in the following equations:

\[
d_i = \text{floor} \left( x_i \times 10^m \right) \text{mod } L, \quad d_i = \text{floor} \left( x_i \times 2^m \right) \text{mod } L.
\] (4)

3.4. Permutation without Repetition. Reshape the two-dimensional image matrix called \( P \) into a one-dimensional vector in rows or columns, which is denoted as \( A \). Generate a pseudorandom sequence, \( X_i \) with the length of \( M \times N \) by using the Lorenz hyperchaotic system, where \( i = 1, 2, \ldots, MN \). Only the first one of the repeated pseudorandom numbers in \( X \) is preserved. Then, add the values in the set \( \{1, 2, \ldots, MN\} \) that do not appear in \( X \) to the end of \( X \) in ascending order. Finally, swap \( A(X_i) \) and \( A(X_{MN-i+1}) \).

3.5. Permutation Associated with Plaintext

**Step 1.** For a given pixel coordinate \((x, y)\) in image \( A \), we can use equations (6) and (7) to calculate a new coordinate of the pixel \((m, n)\) in image \( B \), as shown in the following equation:
Figure 1: The projections of Lorenz attractor. (a) x-y plane. (b) x-z plane. (c) x-w plane. (d) y-z plane. (e) y-w plane. (f) z-w plane.
algorithm, in equations (9) and (10), where the matrix and dimensional vector in rows or columns, which is denoted as eliminated first, and then the remaining 255 elements are numbered. In the multiplication operation, in order to reduce C

\[ C_i = C_{i+1} \times S_i \times P_i, \]  
\[ P_i = C_i + C_{i+1} \div S_i. \]  

where \( k = 1, 2, \ldots, M, \ l = 1, 2, \ldots, N, \ \{ x_i \}, \{ y_i \}, \{ z_i \}, \) and \( \{ w_i \} \) \( i = 1, 2, \ldots, MN \) are the four pseudorandom sequences generated by the Lorenz hyperchaotic system.

As for equations (6) and (7), if \( (m = i \) or \( m = Z(i, j) \)), \( (n = j \) or \( n = W(i, j) \)), or \( (Z(i, j) = i \) or \( W(i, j) = j) \), then the position of \( A(i, j) \) remains unchanged; otherwise, \( A(i, j) \) and \( A(m, n) \) swap positions.

Step 2. When coordinate \( (i, j) \) traverses all the pixels of image \( A \) in the scanning order from left to right and top to bottom, repeat step 1; we can convert image \( A \) into image \( B \).

3.6. Diffusion Based on Finite Fields GF (257). Diffusion operation hides the information of any original image pixels in as many cipher image pixels as possible without changing the position of pixels. In cryptography, the finite field called GF \( (p) \) is an important field, where \( p \) is a prime number. It is a finite set of integers and uses mode operations to perform basic mathematical operations [52]. This study uses a finite field, \( GF(257) = \{ 0, 1, \ldots, 256 \} \), where 257 is a prime number. In the multiplication operation, in order to reduce the information loss caused by element 0, element 0 can be eliminated first, and then the remaining 255 elements are used for arithmetic operations.

Reshape the two-dimensional image matrix into a one-dimensional vector in rows or columns, which is denoted as \( P \). Let \( C \) and \( S \) be the cipher vector; then the forward diffusion algorithm, \( i \) from 1 to \( MN \), and its inverse operation are shown in equations (9) and (10), where \( M \) is the number of rows in the matrix and \( N \) is the number of columns in the matrix.

\[ C_i = C_{i+1} \times S_i \times P_i, \]  
\[ P_i = C_i + C_{i+1} \div S_i. \]  

The reverse diffusion algorithm, \( i \) from \( MN \) to 1, and its inverse operation are shown in equations (11) and (12), respectively:

where \( Z, W, U, \) and \( V \) are four pseudorandom matrices generated by the hyperchaotic Lorenz system; the sizes of them are all \( M \times N \), as shown in the following equation:

\[ Z(k, l) = \left( \text{floor}(z_{(k-1)\times N+l} \times 10^{13}) \mod M \right) + 1, \]  
\[ W(k, l) = \left( \text{floor}(w_{(k-1)\times N+l} + 500 \mod 1) \times 10^{12} \right) \mod N + 1, \]  
\[ U(k, l) = \left( \text{floor}(x_{(k-1)\times N+l} + y_{(k-1)\times N+l} + 500 \mod 1) \times 10^{12} \right) \mod M + 1, \]  
\[ V(k, l) = \left( \text{floor}(z_{(k-1)\times N+l} + w_{(k-1)\times N+l} + 500 \mod 1) \times 10^{12} \right) \mod N + 1. \]  

The multiplication and division used in the above four equations are arithmetic operations that satisfy the finite field of GF (257).

4. The Proposed Image Encryption Scheme

4.1. Image Encryption Process. This research uses Lorenz hyperchaotic system, RSA algorithm, nonrepetitive permutation, and finite field diffusion algorithm to realize a new asymmetric image encryption scheme. The flowchart of image encryption is shown in Figure 2, where the encryption process is described as follows:

Step 1: select prime numbers \( p \) and \( q \); calculate \( n = p \times q \) and \( \varphi(n) = (p - 1)(q - 1) \).

Step 2: generate public key \( (e, n) \) and private key \( (d, n) \) by using RSA algorithm.

Step 3: four large positive integers \( (m_1, m_2, m_3, m_4) \) are randomly selected as confidential information; then the public key \( (e, n) \) is used to calculate \( c_i = m_i \mod n, i = 1, 2, 3, 4 \), which is sent to the receiver.

Step 4: equation (13) is utilized to calculate the parameters \( x_0, y_0, z_0, w_0 \) of the Lorenz hyperchaotic system.

\[ \begin{align*}
  x_0 &= \sqrt{\text{log}(c_1 + m_1)}, \\
  y_0 &= \sqrt{\text{log}(c_2 + m_2)}, \\
  z_0 &= \sqrt{\text{log}(c_3 + m_3)}, \\
  w_0 &= \sqrt{\text{log}(c_4 + m_4)}.
\end{align*} \]  

Step 5: substitute parameters \( x_0, y_0, z_0, w_0 \) into equations (1) and (3) to generate pseudorandom sequence \( S, X, \) and \( R \), and convert the generated values into the range of 0 to 255:
Security and Communication Networks

Finite field diffusion
Pseudorandom sequence
Lorenz hyperchaotic system

\[
\begin{align*}
S &= \text{mod}(\text{floor}((s+100) \times 10^{10}), 256), \\
X &= \text{mod}(\text{floor}((s+100) \times 10^{14}), 256), \\
R &= \text{mod}(\text{floor}((s+100) \times 2^{16}), 256).
\end{align*}
\]

Step 6: record the plain image as P and perform the following additive mode diffusion operation twice with the generated key stream S to obtain image A: \[
\begin{align*}
A_i &= (A_{i-1} + S_i + P_i) \text{mod} \ 256, \\
A_i &= (A_{i+1} + S_i + P_i) \text{mod} \ 256,
\end{align*}
\]

where \(A_i, P_i,\) and \(S_i\) represent the elements of \(A, P,\) and \(S.\)

Step 7: use the key stream X and conduct nonrepetitive permutation of image A to obtain image B. Perform permutation associated with plaintext.

Step 8: according to equations (9) and (11), use the key stream \(R\) and conduct finite field diffusion of image B to obtain image C:

\[
\begin{align*}
C_i &= C_{i-1} + R_i + B_i, \\
C_i &= C_{i+1} + R_i + B_i,
\end{align*}
\]

where \(C_i, R_i,\) and \(B_i\) represent the elements of \(C, R,\) and \(B.\)

Step 9: finally, transform C to get an encrypted image matrix E.

As to the color image, it can be treated as three gray images by three channels \(R, G,\) and \(B.\) So, the encryption is the same for each channel.

4.2. Image Decryption Process. Image decryption is the reverse process of image encryption, as follows:

Step 1: the receiver decrypts the original image with a private key \((d, n)\) based on the received ciphertext information \(c_i, i = 1, 2, 3, 4,\) calculate \(m_i = c_i^d \mod n, i = 1, 2, 3, 4,\) and then equation (13) is used to calculate the parameters \(x_0, y_0, z_0, w_0\) of the Lorenz hyperchaotic system.

Step 2: substitute parameters \(x_0, y_0, z_0, w_0\) into equations (1) and (3) to generate pseudorandom sequence \(S', X',\) and \(R',\) and convert the generated values into the range of 0 to 255:

\[
\begin{align*}
S' &= \text{mod} \left( \text{floor}((s' + 100) \times 10^{10}), 256 \right), \\
X' &= \text{mod} \left( \text{floor}((s' + 100) \times 10^{14}), 256 \right), \\
R' &= \text{mod} \left( \text{floor}((s' + 100) \times 2^{16}), 256 \right).
\end{align*}
\]

Step 3: according to equations (10) and (12), use the key stream \(R'\) and conduct finite field diffusion of image E to obtain image C':

\[
\begin{align*}
C_i' &= E_i + E_{i-1} + R_{i}', \\
C_i' &= E_i + E_{i+1} + R_{i}.
\end{align*}
\]

where \(C_i', E_i,\) and \(R_i'\) represent the elements of \(C', E',\) and \(R'.\)

Step 4: use the key stream \(X'\) and conduct nonrepetitive permutation of image C' to obtain image B'. Perform permutation associated with plaintext.

Step 5: perform the following additive mode diffusion operation twice on B' with the generated key stream S' to obtain image A':

\[
\begin{align*}
A_i' &= (2 \times 256 + B_{i-1}' - S_i') \text{mod} \ 256, \\
A_i' &= (2 \times 256 + B_{i+1}' - S_{i+1}') \text{mod} \ 256,
\end{align*}
\]

where \(A_i', B_i',\) and \(S_i'\) represent the elements of \(A', B',\) and \(S'.\)

Step 6: transform \(A_i'\) to get the original image matrix \(P'.\)

5. Analysis of Experimental Results

5.1. Random Tests for Chaotic Sequence. We used NIST SP800-22 to test the randomness of the Lorenz hyperchaotic system’s sequences (see Table 1). NIST SP800-22 test gives a total of 15 test methods to test the random characteristics of the sequence. Each test produces a P value in \([0, 1]\). If the P value is higher than a preset threshold \(\alpha\), it means that the chaotic sequence passes the test [53]. In our tests, we set \(\alpha = 0.01,\) and the length of the chaotic sequence is \(10^6.\)
Four positive integers were separately selected as $c$.

The image usually has a uniform frequency distribution and will have the encryption time of our scheme is shorter than that of the other schemes. (%_herefore, the proposed scheme is efficient.

The binary matrix rank test 0.1220 Pass

Tests for the longest-run-of-ones in a block 0.0798 Pass

The discrete Fourier transform test 0.8164 Pass

Maurer’s “universal statistical” test 0.4561 Pass

The nonoverlapping template matching test (2) 0.10 Pass

The overlapping template matching test (1) 0.9104 Pass

The random excursions test (1 and 2) 0.1395 Pass

The serial test (test 1 and test 2) 0.6031 Pass

The runs test 0.6031 Pass

Frequency test 0.8823 Pass

The frequency test 0.8823 Pass

Frequency test within a block (block = 10000) 0.7208 Pass

The serial test (test 1 and test 2) 0.1395 Pass

The cumulative sums test (forward and reverse test) 0.8008 Pass

The random excursions variant test (1) 0.9104 Pass

The random excursions variant test (x = 1) 0.6031 Pass

Table 1: Randomness test by NIST SP800-22 for the Lorenz hyperchaotic system’s sequences.

<table>
<thead>
<tr>
<th>Test name</th>
<th>$P$ value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>The frequency test</td>
<td>0.8823</td>
<td>Pass</td>
</tr>
<tr>
<td>Frequency test within a block (block = 10000)</td>
<td>0.7208</td>
<td>Pass</td>
</tr>
<tr>
<td>The runs test</td>
<td>0.6031</td>
<td>Pass</td>
</tr>
<tr>
<td>Tests for the longest-run-of-ones in a block</td>
<td>0.1220</td>
<td>Pass</td>
</tr>
<tr>
<td>The binary matrix rank test</td>
<td>0.0798</td>
<td>Pass</td>
</tr>
<tr>
<td>The discrete Fourier transform test</td>
<td>0.8164</td>
<td>Pass</td>
</tr>
<tr>
<td>The nonoverlapping template matching test (m = 10)</td>
<td>0.2990</td>
<td>Pass</td>
</tr>
<tr>
<td>The overlapping template matching test (m = 9, template = 111111111)</td>
<td>0.4298</td>
<td>Pass</td>
</tr>
<tr>
<td>Maurer’s “universal statistical” test</td>
<td>0.4561</td>
<td>Pass</td>
</tr>
<tr>
<td>The linear complexity test (block = 1000)</td>
<td>0.6040 [0.6013, 0.3395]</td>
<td>Pass</td>
</tr>
<tr>
<td>The serial test (test 1 and test 2)</td>
<td>0.1395</td>
<td>Pass</td>
</tr>
<tr>
<td>The approximate entropy test</td>
<td>0.4992, 0.5407</td>
<td>Pass</td>
</tr>
<tr>
<td>The cumulative sums test (forward and reverse test)</td>
<td>0.8008</td>
<td>Pass</td>
</tr>
<tr>
<td>The random excursions test (x = -1)</td>
<td>0.9104</td>
<td>Pass</td>
</tr>
<tr>
<td>The random excursions variant test (x = 1)</td>
<td>0.6031</td>
<td>Pass</td>
</tr>
</tbody>
</table>

5.2. Experimental Results. This paper selects some different images from the USC-SIPI database. And Windows 10 operating system was used with the MATLAB R2020a software. An Intel (R) Core (TM) i5-1035G processor was used, and 16 GB of RAM was required to simulate the work. For the experimental procedure, the private key consisted of large prime numbers $p = 3259, q = 3821, and d = 3385223$. Four positive integers were separately selected as $m_1 = 178334, m_2 = 38628, m_3 = 9282, and m_4 = 829809$. And the public key consisted of $e = 1288367, c_1 = 11587151, c_2 = 1799483, c_3 = 12452638, and c_4 = 4198591$. The obtained results were as follows: $x_0 = 4.0349, y_0 = 3.7979, z_0 = 4.2980, and w_0 = 3.9282$. As it can be seen, the results of encryption and decryption for the test image are shown in Figure 3. The encrypted image in Figures 3(i)–3(p) reveals that no information could be obtained from it. The decrypted image (Figures 3(q)–3(x)) also shows that the plain image information could be correctly restored. This proves that the proposed scheme is effective for image encryption. Table 2 shows the time cost of encryption, which uses different size images. As it can be seen, the encryption time of our scheme is shorter than that of the other schemes. Therefore, the proposed scheme is efficient.

6. Analysis of System Performance

6.1. Histogram Analysis. A histogram is utilized to show the distribution of image pixel intensity. An ideal encrypted image usually has a uniform frequency distribution and will not provide any useful statistical information to the attacker. Figure 4 shows the histogram distribution of some images, showing the uniformity of gray values in the results. Simultaneously, a chi-squared test can be used to evaluate the uniformity of the histogram, which is calculated as

$$\chi^2 = \sum_{i=0}^{255} \frac{(O - E)^2}{E}$$

where $L$ is the intensity level and $O$ and $E$ represent the observed occurrence frequency and the expected occurrence frequency for each gray value, respectively. The uniformity of histogram is assessed with the help of the chi-squared test. Table 3 shows the histogram uniformity results of the chi-squared test for gray images by Lena, Baboon, and Peppers. In the table, at 255 degrees of freedom and 5% and 1% significance level, the chi-square values are $\chi^2_{0.05}(255) = 310.457$ and $\chi^2_{0.01}(255) = 293.2478$, respectively [56]. It can be seen that, at both 5% and 1% significance levels, cipher images have fairly uniform distribution for gray values, different from that of their respective plain images. This means that it is difficult to apply histogram attacks.

For quantity analyses of each key, variances of histograms are employed to evaluate uniformity of ciphered images. The lower value of variances indicates the higher uniformity of ciphered images. We also calculate the two variances of ciphered images which are encrypted by different secret keys on the same plaintext image. The closer of
the two values of variances indicates the higher uniformity of ciphered images when the secret keys are varying. The variance of histograms is presented as follows:

\[ \text{var}(Z) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2}(x_i - x_j)^2, \]

Figure 3: Plain image: (a) Lena; (b) Baboon; (c) Peppers; (d) Boat; (e) Clock; (f) House; (g) Male; (h) Tree. Encrypted: (i) Lena; (j) Baboon; (k) Peppers; (l) Boat; (m) Clock; (n) House; (o) Male; (p) Tree. Decrypted: (q) Lena; (r) Baboon; (s) Peppers; (t) Boat; (u) Clock; (v) House; (w) Male; (x) Tree.

Table 2: Time cost analysis.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Boat (512*512)</th>
<th>Lena (256*256)</th>
<th>Lena (256*256)</th>
<th>Lena (256*256)</th>
<th>Lena (256*256)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption time (s)</td>
<td>0.269177</td>
<td>0.079054</td>
<td>0.132</td>
<td>0.417</td>
<td>0.580</td>
</tr>
</tbody>
</table>
where $Z$ is the vector of the histogram values and $Z = \{z_1, z_2, \ldots, z_{256}\}$ and $z_i$ and $z_j$ are the numbers of pixels where gray values are equal to $i$ and $j$, respectively. In simulating experiments, we calculate two variances of histograms of two ciphered images by equation (21) from the same plaintext image with different secret keys. Only one parameter of secret keys is changed in such different secret keys. Table 4 lists the variances of histograms of ciphered Lena, BARB, and Boat images. In Table 4, the variances in the first column are obtained by the secret key Key1 \((x_0 = 4.0349, y_0 = 3.7979, z_0 = 4.2980, w_0 = 3.9282)\); the variances in next columns are obtained by only changing one parameter of $x_0, y_0, z_0,$ and $w_0$, respectively, compared with the secret key Key1. The variance values are about 1000, which indicates that the average fluctuation of the number of pixels in each gray value is about 114 pixels. However, the variance value is 634576.2901 for histogram of the plaintext image Lena. And the variance value is 5335.8309 for histogram of ciphered image of Lena in Zhang’s paper [57], which is greater than any of the variances in Table 4. Therefore, the proposed algorithm is efficient.

6.2. Correlation Coefficient. The correlation coefficient is a linear correlation between adjacent image pixels. In general, plain images have a strong correlation between adjacent pixels in horizontal, vertical, and diagonal directions, but there should be no correlation between adjacent pixels in cipher images. In the experiment, we randomly selected 2000 pairs of adjacent pixels from the original images and the encrypted images and analyzed the correlations at horizontal, vertical, and diagonal directions. The calculation formula is defined as
Table 4: Variances of histograms compared among all secret keys in the proposed algorithm.

<table>
<thead>
<tr>
<th>Cipher image</th>
<th>Key 1</th>
<th>$x_0$</th>
<th>$y_0$</th>
<th>$z_0$</th>
<th>$w_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>1047.6549</td>
<td>1038.0390</td>
<td>1044.2187</td>
<td>1030.9570</td>
<td>1094.3476</td>
</tr>
<tr>
<td>BARB</td>
<td>941.1686</td>
<td>970.0156</td>
<td>924.8242</td>
<td>961.7343</td>
<td>912.7245</td>
</tr>
<tr>
<td>Boat</td>
<td>990.0784</td>
<td>1005.9726</td>
<td>984.6914</td>
<td>1056.5585</td>
<td>971.8085</td>
</tr>
<tr>
<td>Average</td>
<td>992.9672</td>
<td>1004.6757</td>
<td>984.5781</td>
<td>1016.0976</td>
<td>992.9602</td>
</tr>
</tbody>
</table>

\[ r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \]

\[ \text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)), \]

\[ E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i, \]

\[ D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \]

where $r_{xy}$ is the correlation coefficient, $x$ and $y$ are the gray values of two adjacent pixels, and $N$ is the total quantities of $(x_i, y_i)$. The correlation coefficient value is between $-1$ and $+1$. And the correlation coefficient of two independent and uncorrelated random sequences (adjacent pixels) is more close to zero [58]; the effect of encryption is better. The correlation coefficient between the original image (Male) and the encrypted image is shown in Figure 5. Table 5 calculates the correlation coefficients of some images from horizontal, vertical, and diagonal directions. Table 6 takes the color Lake image as an example to analyze the correlation coefficient between the plain image and the cipher image. Table 7 compares the correlation coefficient results of gray Boat image by different encryption algorithms. The results show that the proposed encryption algorithm breaks the strong correlation in the original image and can effectively resist statistical attacks.

6.3. Information Entropy. Information entropy reflects the uncertainty of image information; it is generally believed that the greater the value of information entropy, the greater the uncertainty (the more information). It can be used to measure the randomness of data series. It is mathematically defined as

\[ H = -\sum_{i=0}^{L} p(i)\log_2 p(i), \]

where $L$ is the grayscale of the image and $p(i)$ is the probability that the gray value $i$ will appear. For digital images with 256 grayscale levels, the theoretical value of entropy $H$ is 8. The higher the entropy value of the cipher image, the more uniform the distribution of pixel value. In addition, local Shannon entropy measures the randomness of image by calculating the sample mean of Shannon entropy on randomly selected image blocks and multiple nonoverlapping, so it can overcome the disadvantages of global information entropy, such as low efficiency, inconsistency, and inaccuracy [59]. Table 8 shows the local entropy of the cipher image and the information entropy of the image. And the average information entropy value of the three channels is calculated as the information entropy value of color image. It can be seen from the results that the entropy value of cipher image is very close to the theoretical value of 8, while the information entropy of plain image is significantly different from the theoretical value.

We make the comparison with some color images by information entropy [60, 61]. Table 9 shows the results. It can be seen that the proposed algorithm has good performance to resist differential attacks.

6.4. Differential Attack Analysis. In general, two encrypted images (the original cipher image and the new cipher image by changing one pixel in the original image) are compared to analyze the relationship between the original image and the corresponding cipher image in differential attack. The Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are two common indicators of attack resistance [62–66]. NPCR measures the rate of change of pixel values in a cipher image by changing a pixel value in the original image, while UACI measures the average changing intensity between the original image and the corresponding cipher image. They can be defined as

\[ \text{NPCR}(C_1, C_2) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j)}{M \times N} \times 100\%, \]

\[ D(i, j) = \begin{cases} 0, & \text{if } C_1(i, j) = C_2(i, j), \\ 1, & \text{if } C_1(i, j) \neq C_2(i, j), \end{cases} \]

\[ \text{UACI}(C_1, C_2) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |C_1(i, j) - C_2(i, j)|}{M \times N \times T} \times 100\%, \]

where $M$ is the width of an image, $N$ is the height of an image, and $T$ is the grayscale of an image. Moreover, $C_1$ is the cipher image and $C_2$ is the modified cipher image after a pixel value of the original image is changed. Table 10 shows the NPCR and UACI values of some images. Table 11 lists the results of NPCR and UACI when the pixel values at different positions of Lena image are changed. All values of NPCR and UACI of our method are close to ideal values $99.6094\%$ and $33.4635\%$, respectively.

6.5. Key Space Analysis. A key space is a collection of all valid keys. For the digital image cryptosystem proposed in this paper, the keys, $K = \{x_0, y_0, z_0, w_0\}$, are the initial value...
Table 6: Correlation coefficients of color Lake.

<table>
<thead>
<tr>
<th></th>
<th>Plain image</th>
<th>Cipher image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R (red)</td>
<td>G (green)</td>
</tr>
<tr>
<td>Horizontal</td>
<td>0.9533</td>
<td>0.9648</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9594</td>
<td>0.9680</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9440</td>
<td>0.9543</td>
</tr>
</tbody>
</table>

Table 7: Comparison of correlation coefficients of gray Boat image.

<table>
<thead>
<tr>
<th>Encryption algorithm</th>
<th>Plain image</th>
<th>Cipher image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Horizontal</td>
<td>Vertical</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.9709</td>
<td>0.9353</td>
</tr>
<tr>
<td>Ref. [9]</td>
<td>0.9629</td>
<td>0.9672</td>
</tr>
<tr>
<td>Ref. [43]</td>
<td>0.9379</td>
<td>0.9383</td>
</tr>
</tbody>
</table>
of Lorenz hyperchaotic system, where \( x_0 \in (-40, 40), \) \( y_0 \in (-40, 40), \) \( z_0 \in (1.81), \) and \( w_0 \in (-250, 250). \) The steps of \( x_0, y_0, \) and \( z_0 \) are all \( 10^{-13}, \) and the steps of \( w_0 \) are \( 10^{-12}. \) Therefore, the space size of the key is about \( S \approx 2.56 \times 10^{35}, \) which is equal to the length of the key \( L = \log_2 S = 197 \text{bit}. \)
If the eavesdropper uses an exhaustive search of the key to crack the encryption or decryption of the cryptosystem, it only needs to attempt half of the keys in the key space. Since the methods of encryption and decryption are reciprocal, it is sufficient to discuss the case of encryption. For the known plaintext and ciphertext pairs, if the ciphertext is exactly the same as the known ciphertext, the randomly selected key is the true key. And the time needed to crack it by an exhaustive search of the key can be approximated by the product of the number of keys in half of the key space and the time of a single encryption, which is about \(4.4607 \times 10^{31}\) years. That is to say, the key space of Lorenz chaotic system is large enough.

6.6. Key Sensitivity Analysis. Key sensitivity analysis is to analyze the difference between two cipher images obtained by encrypting the same plain image when the key changes slightly. If there are significant differences between the two cipher images, the key sensitivity of the image encryption system is strong; if there are small differences between the two cipher images, the key sensitivity is poor. A good image encryption system should have strong key sensitivity. The test process is as follows.

Suppose the initial set of keys is recorded as keys 1, denoted as

\[
\text{keys1} = \{p, q, e, d, m_1, m_2, m_3, c_1, c_2, c_3, c_4\},
\]

(25)

where the values of \(p, q, d, m_1, m_2, m_3, c_1, c_2, c_3, c_4\) represent the private key and \(e, c_1, c_2, c_3, c_4\) are the public key. Keys 1 is used to encrypt the gray image (Lena). Figure 6(a) shows the gray Lena image, and Figure 6(b) shows the corresponding cipher image. Suppose a value in keys 1 changes slightly; then keys 2 is expressed as

\[
\text{keys2} = \{p, q, e, d, m_1 - 1, m_2, m_3, c_1, c_2, c_3, c_4\}.
\]

(26)

Keys 2 is used to encrypt the gray Lena image to obtain a new cipher image shown in Figure 6(c). Figure 6(d) is a different image of Figures 6(b) and 6(c). When the wrong decryption key, keys 2, is used to decrypt the image in Figure 6(b), the decrypted image can be obtained as shown in Figure 6(e). Figure 6(f) shows the correct decrypted image. And the NPCR and UACI values of Figures 6(b) and 6(c) were, respectively, 99.60785\% and 33.5044\%, which indicates that more than 99\% of the pixels could be changed with just a single key change. Therefore, the proposed algorithm is highly sensitive to the key.

6.7. Security Analysis. Currently, a method that involves an exhaustive search of the key is used to crack a password, which attempts all possible key combinations. However, the RSA algorithm uses exponential calculations in both the encryption and decryption processes, which has a huge computational workload. Therefore, it is impossible to decipher by using an exhaustive search. Cryptographic analysis is the only means to decipher the RSA algorithm. However, cracking RSA cryptography needs to factor two large integers, which is difficult. Moreover, Lorenz hyperchaotic system is sensitive to initial conditions, pseudorandomness, nonlinearity, and nonperiodicity. Therefore, the proposed encryption scheme can safely and effectively hide information.

6.8. Chosen-Plaintext Attacks or Chosen-Ciphertext Attacks. The capacity of resisting chosen-plaintext attack or chosen-ciphertext attacks is a significant standard to measure security of image encryption [70–72]. Plaintext sensitivity analysis refers to using the same key to encrypt two original images with little difference with the help of the image encryption system to obtain two corresponding cipher images and then compare the differences between the two cipher images. If the difference between the two cipher images is relatively large, it is said that the image encryption system has good plaintext sensitivity, which means that the cryptographic system can resist chosen-plaintext attacks. The general process of plaintext sensitivity is as follows:

(1) For a certain original image \(P_1\), with the help of a given key, use the image encryption system to encrypt \(P_1\) to obtain the corresponding cipher image, denoted as \(C_1\).

(2) Randomly select a pixel from \(P_1\), change the value of the selected pixel, the amount of change is 1, the changed image is marked as \(P_2\), and the same key is used to encrypt \(P_2\) to obtain the corresponding cipher image, denoted as \(C_2\).

(3) Compare the difference between \(C_1\) and \(C_2\), calculate the values of NPCR and UACI.

(4) Repeat steps (2) and (3) 200 times, calculate the value of a group of NPCR and UACI each time, and finally calculate the average value of 200 groups.

Table 12 shows that the NPCR and UACI of test images are approximately equal to the ideal values. In other words, the digital image encryption system proposed in this paper has good plaintext sensitivity; it can resist chosen-plaintext attacks. Chosen-ciphertext attacks are the same as chosen-plaintext attacks because the image encryption system adds permutation associated with the plaintext, which can enhance the ability of the image encryption scheme to resist chosen-plaintext attacks or chosen-ciphertext attacks.

Table 13 shows that the cryptosystem can resist chosen-ciphertext attacks.

6.9. Noise and Data Loss Attacks. It is inevitable that the digital signal may be corrupted by noise or data loss during the transmission. In this situation, we hope to acquire the content of the original images as much as possible so it is not necessary to transmit the encrypted images again. When the image information is tampered or destroyed by others after being attacked by noise (such as Gaussian noise and salt and pepper noises), the algorithm proposed in this paper can detect that the image has been tampered or destroyed by others. However, the algorithm cannot reconstruct the cipher images, because the algorithm is actually very sensitive to the ciphertext; if the ciphertext changes slightly, the
decrypted image will also change greatly. If the cipher image is artificially destroyed or tampered with during transmission, the algorithm cannot reconstruct the cipher images to obtain the content of the original images as much as possible. As for data loss attacks, the algorithm cannot also reconstruct the cipher images well. In the future, I will study the impact of noise attacks on image encryption and improve the robustness of the algorithm.

7. Conclusions

This study proposes a new image encryption scheme based on Lorenz hyperchaotic system and Rivest–Shamir–Adleman (RSA) algorithm. Firstly, the initial values of the Lorenz hyperchaotic system are generated by RSA algorithm, and the key stream is produced iteratively. Then, the image data are hidden by performing additive mode diffusion. Secondly, the diffusion image matrix is confused without repetition to hide the image data again. Then, the finite field diffusion algorithm is executed. In order to diffuse the pixel information into the entire ciphertext image, the additive mode diffusion algorithm needs to be looped twice. Finally, the ciphertext image can be obtained. The experimental results prove that the image encryption scheme proposed in this research is effective and has strong antiattack and key sensitivity. In the future, we will study how to improve the efficiency of the scheme for color images.

Data Availability

This paper uses some different images from the USC-SIPI database through http://sipi.usc.edu/database/.

Conflicts of Interest

The authors declare no conflicts of interest.
Authors' Contributions

Riguang Lin and Sheng Li contributed to methodology, validation, data curation, and review and editing; Riguang Lin contributed to software, formal analysis, resources, and original draft preparation; Sheng Li contributed to investigation, supervision, and funding acquisition. All authors have read and agreed to the published version of the paper.

Acknowledgments

This work was supported by the Natural Science Foundation of Guangdong Province (2018A030307062) and the Fund of Southern Marine Science and Engineering Guangdong Laboratory (Zhanjiang) (ZJW-2019-04).

References


