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

Research Article

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

Wei Feng, Jing Zhang, and Zhentao Qin

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

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

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

Academic Editor: Ahmed A. Abd El-Latif

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

1. Introduction

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

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

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

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

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

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

(2) Dynamic perturbation parameters (DPPs) derived from system times and last encryption times are designed. So, ITS-CDD not only guarantees the diversity of equivalent key streams, but also does not rely on external algorithms.

(3) The secret key design of SIT-SS is more practical and requires fewer hardware resources.

(4) 2D-CS based on lightweight chaotic map can reduce resource overhead.

(5) Discrete logarithms under finite multiplicative group Z_{257}^{*} are introduced to ensure higher security.

The remainder of this paper is organized as follows. 2D-CS, discrete logarithms, and two chaotic systems are introduced in Section 2. ITS-CDD is described in Section 3. Simulation tests and theoretical analyses are carried out in Section 4. Finally, conclusions are drawn in Section 5.

2. Fundamental Knowledge

In SIT-SS, 2D-CS is introduced to realize image data compression and encryption. Discrete logarithms are used to enhance the security of SIT-SR. Two chaotic systems called Tent-Logistic Map (TLM) [38] and 2D Logistic-Sine-Coupling Map (2D-LSCM) [13] are adopted to generate the chaotic sequences.

2.1. 2D-CS. In terms of computational complexity and storage space, 2D-CS has obvious advantages over traditional CS [39, 40]. Assuming that $\mathbf{A}$ and $\mathbf{B}$ are random matrices, they both have the size of $M \times N$ ($M \ll N$). Then, one can obtain the 2D measurements $\mathbf{Y} \in R^{M \times M}$ of an image $\mathbf{X} \in R^{N \times N}$. Specifically,

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{B}^{T}, \tag{1}$$

where $\mathbf{A}$ and $\mathbf{B}$ operate on the rows and columns of $\mathbf{X}$, respectively.

When decoding, one can regularize the image signal recovery by using signal prior information in the form of penalty:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} f(\mathbf{X}) = \frac{1}{2}\left\|\mathbf{Y} - \mathbf{A}\mathbf{X}\mathbf{B}^T\right\|_F^2 + \lambda J(\mathbf{X}), \quad (2)$$

where λ is the regularization parameter, $J(\mathbf{X})$ is a cost function which is used to handle the ill-posed problem, and $(1/2)\|\mathbf{Y} - \mathbf{A}\mathbf{X}\mathbf{B}^T\|_F^2$ is the l_2 data-fidelity term. Moreover, researchers have proposed many 2D-CS reconstruction algorithms to solve the optimization problem mentioned above. In this paper, 2D projected gradient with embedding decryption (2DPG-ED) [12] algorithm is adopted.

2.2. Discrete Logarithms. Discrete logarithm calculation is a complex nonlinear calculation. In the encryption process, the use of discrete logarithms can improve its nonlinearity [14]. For the prime 257 and its corresponding finite multiplicative group Z_{257}^*, one can define the discrete logarithms as follows: if $d \in Z_{257}^*$ satisfies $n \equiv g^d \bmod p$, then d is said to be the discrete logarithm of $n \in Z_{257}^*$. Since Z_{257}^* has 128 generators, we can use them to enhance the diversity of equivalent key streams. To avoid complex discrete logarithm calculation, we calculate the discrete logarithm values under different generators in advance and save them to the 2D matrix **DV M** with the size of 128×256. Consequently, in ITS-CDD, discrete logarithm values can be obtained by directly accessing **DV M**. If one wants to calculate the discrete logarithm value of 107 under the generator 3, namely, calculating $(\log_3 107) \bmod 257$, one can access $\mathbf{DVM}_{1,107}$ to obtain the discrete logarithm value 31. Table 1 shows the discrete logarithm values of 101 to 107 under the first eight generators.

2.3. TLM and 2D-LSCM. To save hardware resources, TLM is adopted in SIT-SS, which is easy to implement and has good chaotic performance. TLM can be defined as

$$x_i = \begin{cases} r_1 r_2 x_{i-1}(1 - r_2 x_{i-1}), & x_{i-1} < 0.5, \\ r_1 r_2 (1 - x_{i-1})(1 - r_2(1 - x_{i-1})), & x_{i-1} \geq 0.5, \end{cases} \quad (3)$$

where x_i is generated by the i-th iteration, x_{i-1} is the input of the i-th iteration, $x_0 \in (0, 1)$ is the initial state, and $r_1 \in [3.57, 4], r_2 \in (1, 2]$ are the control parameters. Figure 1 shows the 2D bifurcation diagram and Lyapunov Exponents (LE) diagrams of TLM.

Compared with TLM, 2D-LSCM has better chaotic performance, but its structure is more complex, so it is more suitable for environments with more hardware resources. 2D-LSCM can be defined as

$$\begin{cases} x_i = \sin\left(\pi\left(4\gamma x_{i-1}(1 - x_{i-1}) + (1 - \gamma)\sin\left(\pi y_{i-1}\right)\right)\right), \\ y_i = \sin\left(\pi\left(4\gamma y_{i-1}(1 - y_{i-1}) + (1 - \gamma)\sin\left(\pi x_{i-1}\right)\right)\right), \end{cases} \quad (4)$$

where (x_i, y_i) is the system state generated by the i-th iteration, (x_{i-1}, y_{i-1}) is the input of the i-th iteration, (x_0, y_0) is the initial state, and γ is the control parameter. The value ranges of all these parameters are $[0, 1]$. Figure 2 shows the 2D bifurcation diagram and LE diagram of 2D-LSCM.

3. Proposed Image Transmission Scheme

Different from some existing schemes, ITS-CDD consists of two parts, secure image transmission between sensor nodes and sink nodes (SIT-SS) and secure image transmission between sensor nodes and receivers (SIT-SR). Figure 3 shows the secure image transmission between sensor nodes and sink nodes.

Compared with the existing schemes, SIT-SS has two main innovations. One is introducing TLM to save the hardware resources of sensors, and the other is introducing DPPs to enhance the ability to resist CPAs. Figure 4 shows the secure image transmission between sink nodes and receivers.

Considering that sink nodes have more resources, there are huge security threats in the process of transmitting images to receivers through the media cloud. We have adopted some measures to improve the security of image transmission, such as the adoption of 2D-LSCM with better chaotic performance and the introduction of discrete logarithms.

3.1. Transmission between Sensor Nodes and Sink Nodes. To save space, in this subsection, we mainly introduce the improvements to 2DCS-ETC [12].

3.1.1. DPP Generation. According to previous cryptanalysis works, the main reason why some schemes cannot resist CPAs is that equivalent key streams only depend on the secret key [41–47]. Therefore, some researchers use the hash value of the plain image to ensure the diversity of equivalent key streams. However, the implementation of hash algorithm is not suitable for sensor nodes with limited resources. Considering that system times and last encryption times are constantly changing and would be affected by many factors, they are used to generate DPPs. The specific generation process of DPPs is as follows:

(i) Step 1: obtain the system time T_s in milliseconds.

(ii) Step 2: get the time T_e spent in the last encryption process in milliseconds. If it is the first time to encrypt, set T_e to an initial value T_i.

(iii) Step 3: one DPP is obtained by $\beta = (T_s + T_e) \bmod 256$.

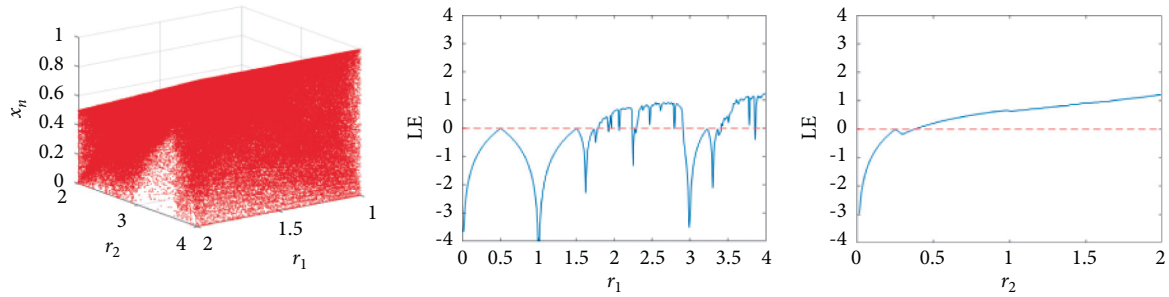
(iv) Step 4: repeat Step 1 through Step 3 until 32 DPPs are obtained, namely, $\beta_1, \beta_2, \ldots, \beta_{32}$.

In this way, we can obtain a set of DPPs. Like the hash value, DPPs can ensure that the equivalent key streams used when encrypting different images are different, thereby effectively resisting CPAs. More importantly, no complicated calculations are required to obtain DPPs, and even if the same plain image is encrypted, different equivalent key streams would be generated.

3.1.2. TLM-Driven Global Permutation. Obviously, confusion is the requirement that must be considered when designing modern cryptosystems. Confusion means that each

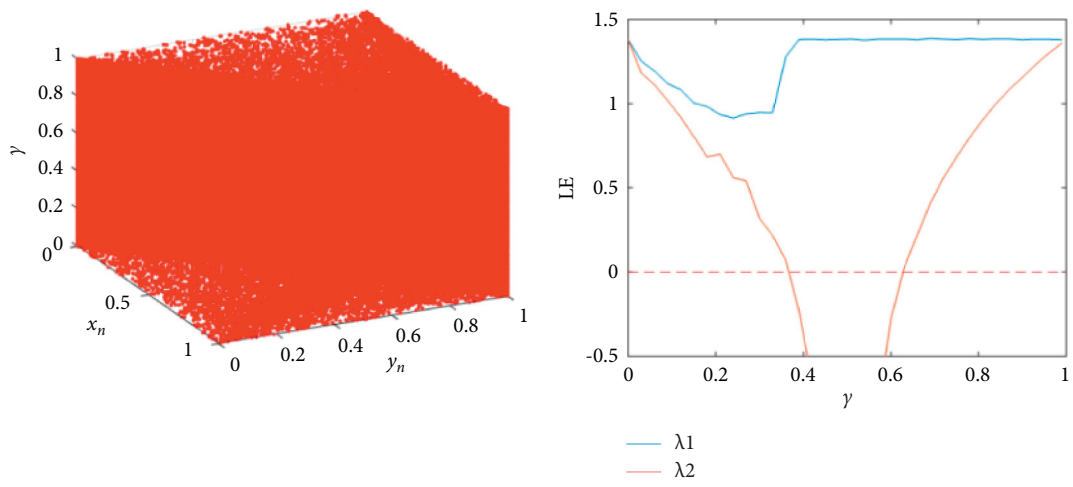
Table 1: Discrete logarithm values of 101 to 107 under the first eight generators.

Row index of **DV M**	Corresponding generator g	n (column index of **DV M**)						
		101 (101)	102 (102)	103 (103)	104 (104)	105 (105)	106 (106)	107 (107)
1	3	75	169	201	250	141	137	31
2	5	141	31	255	214	91	63	89
3	6	59	249	25	26	29	217	79
4	7	31	5	165	18	89	101	163
5	10	125	111	79	246	235	143	137
6	12	43	73	105	58	173	41	127
7	14	143	213	117	50	105	53	83
8	19	103	157	61	2	81	253	203



(a) (b) (c)

Figure 1: 2D bifurcation diagram and LE diagrams of TLM: (a) 2D bifurcation diagram; (b) LE diagram versus parameter r_1; (c) LE diagram versus parameter r_2.



(a) (b)

Figure 2: 2D bifurcation diagram and LE diagram of 2D-LSCM: (a) 2D bifurcation diagram; (b) LE diagram versus parameter γ.

bit of the secret key should affect as many cipher image bits as possible [48]. Permutation operations are commonly used to achieve confusion, but permutation-only image encryption schemes have been proven to be insecure [49]. Therefore, SIT-SS introduces DPPs in the permutation process. This makes the permutation process not only dependent on the secret key, but also dependent on the DPPs that will inevitably change every time the plain image is encrypted. Compared with 2DCS-ETC using the random permutation matrix to complete the permutation and treat it as secret key, we use TLM and DPPs to complete the permutation. This can not only reduce the resource overhead of sensor nodes, but also

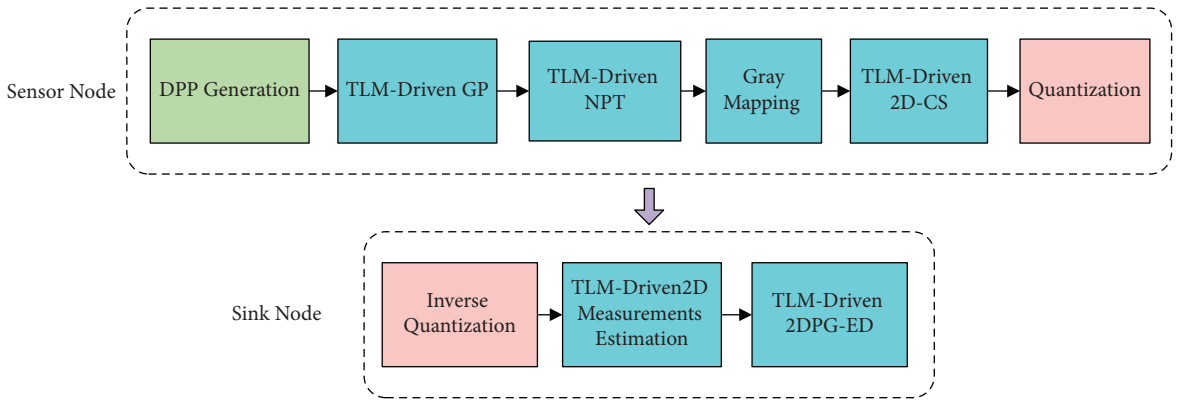


Figure 3: Secure image transmission between sensor nodes and sink nodes.

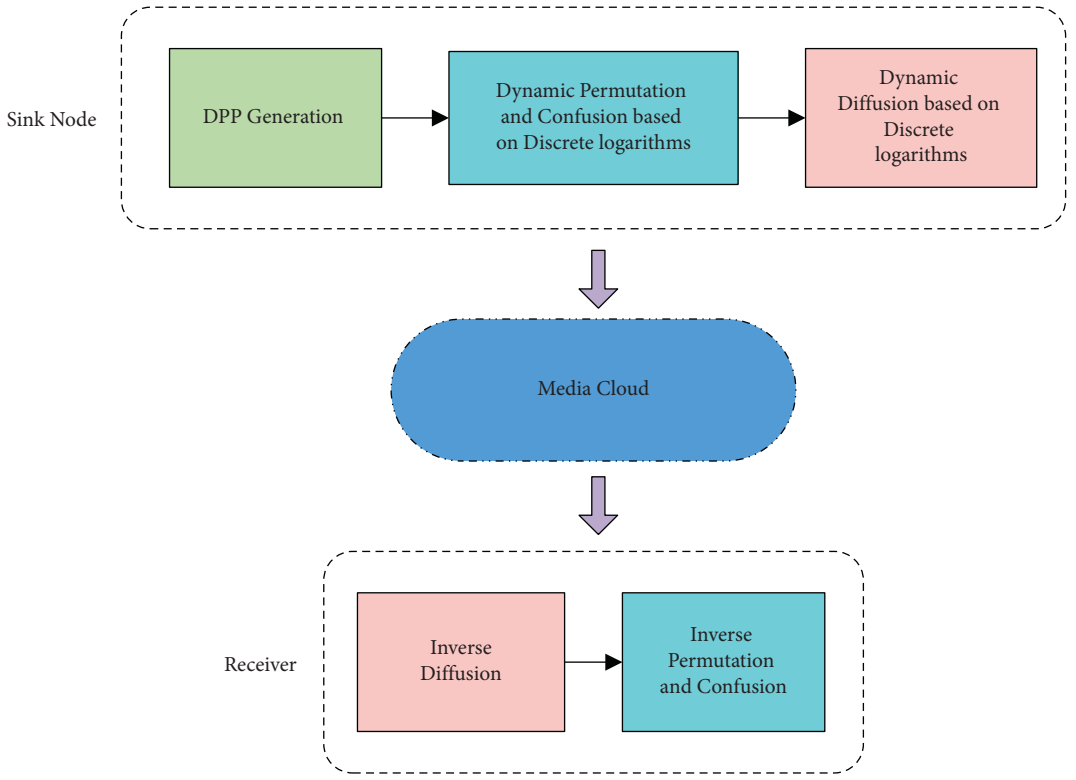


Figure 4: Secure image transmission between sink nodes and receivers.

improve the ability to resist CPAs. The specific process of TLM-Driven Global Permutation (GP) is as follows:

(i) Step 1: use the parameters $(r_1^{(1)}, r_2^{(1)}, x^{(1)})$ to iterate TLM $N \times N + r_3^{(1)}$ times. In order to avoid negative effects, discard the first $r_3^{(1)}$ chaotic state values.

(ii) Step 2: convert the obtained chaotic sequence $\mathbf{S}$ of length $N \times N$ into the integer sequence

$$\mathbf{S}_i = \left(\text{floor}\left(\mathbf{S}_i \times 10^{15}\right) \text{mod}\,(N \times N)\right) + 1, \quad (5)$$

where $i = 1, 2, \ldots, N \times N$, $\text{floor}(\cdot)$ returns the integer part of an operand.

(iii) Step 3: stretch the plain image $\mathbf{P}$ of size $N \times N$ into the 1D sequence $\tilde{\mathbf{P}}$.

(iv) Step 4: calculate the index

$$I = (\mathbf{S}_i \text{mod}\, 32) + 1, \quad (6)$$

of 32 DPPs and the permutation position

$$\alpha = ((\mathbf{S}_i + \beta_I) \text{mod}\,(N \times N)) + 1, \quad (7)$$

where $i = 1, 2, \ldots, N \times N$. Swap two pixels of $\tilde{\mathbf{P}}$ according to α.

3.1.3. TLM-Driven Negative-Positive Transformation. A nonlinear operation called Negative-Positive Transformation (NPT) is introduced by 2DCS-ETC to improve security. Similarly, we use TLM and DPPs to complete NPT instead of using a random matrix in the form of secret key. This can further reduce the resource overhead of sensor nodes and improve the ability to resist CPAs.

(i) Step 1: use parameter $(r_1^{(2)}, r_2^{(2)}, x^{(2)})$ to iterate TLM $N \times N + r_3^{(2)}$ times. In order to avoid negative effects, discard the first $r_3^{(2)}$ chaotic state values.

(ii) Step 2: convert the obtained chaotic sequence $\mathbf{S}$ of length $N \times N$ into the bit sequence

$$\mathbf{S}_i = \text{floor}\left(\mathbf{S}_i \times 10^{15}\right) \bmod 2, \tag{8}$$

where $i = 1, 2, \ldots, N \times N$.

(iii) Step 3: according to $\mathbf{S}$, perform the following NPT operation on $\widetilde{\mathbf{P}}$.

$$\mathbf{C}_i = \begin{cases} \widetilde{\mathbf{P}}_i, & \mathbf{S}_i = 1, \\ 255 - \widetilde{\mathbf{P}}_i, & \mathbf{S}_i = 0, \end{cases} \tag{9}$$

where $i = 1, 2, \ldots, N \times N$.

(iv) Step 4: reshape $\mathbf{C}$ into the 2D cipher image.

3.1.4. TLM-Driven 2D-CS. If the chaotic sequence generated by the chaotic system is assembled into a complete measurement matrix, its performance is usually almost the same as other commonly used random matrices [11]. Moreover, compared with directly using a random matrix and treating it as secret key, the chaotic measurement matrix can significantly save the resource overhead of sensor nodes. In SIT-SS, TLM is used to generate the measurement matrices required for 2D-CS. Suppose the size of the measurement matrices $\mathbf{A}$ and $\mathbf{B}$ to be created is $M \times N$ ($M \ll N$); the specific process of TLM-Driven 2D-CS is as follows:

(i) Step 1: use the parameters $(r_1^{(3)}, r_2^{(3)}, x^{(3)})$ to iterate TLM $N \times N + r_3^{(3)}$ times. In order to avoid negative effects, discard the first $r_3^{(3)}$ chaotic state values.

(ii) Step 2: arrange the obtained chaotic sequence into the square matrix $\mathbf{S}$ of size $N \times N$.

(iii) Step 3: take M rows from the orthogonal basis of $\hat{\mathbf{S}}$ as the measurement matrix $\mathbf{A}$.

(iv) Step 4: repeat Step 1 through Step 3; create the measurement matrix $\mathbf{B}$ in a similar manner.

(v) Step 5: use $\mathbf{A}$ and $\mathbf{B}$ to obtain the 2D measurements of the cipher image $\mathbf{C}$.

In addition to the improvements made above, the other steps of SIT-SS are basically the same as those of 2DCS-ETC, which are not repeated here. Since we have introduced TLM and DPPs in SIT-SS, the security of image transmission between sensor nodes and sink nodes has become higher, and the resource requirements for sensors are also lower. Significantly, SIT-SS still maintains the advantages of 2DCS-ETC, which is demonstrated and discussed in Section 4.1. To save hardware resources, we directly use $r_1^{(1)}, r_2^{(1)}, x^{(1)}, r_1^{(2)}, r_2^{(2)}, x^{(2)}, r_1^{(3)}, r_2^{(3)}, x^{(3)}$ as the secret key of SIT-SS.

3.2. Transmission between Sink Nodes and Receivers. In SIT-SR, we use 2D-LSCM [13] which has better chaotic performance to generate chaotic sequences. Moreover, discrete logarithms and DPPs are introduced to achieve secure image transmission between sink nodes and receivers. It should be noted that through the use of discrete logarithms and our targeted design, DPPs can be directly sent out in plaintext form by sink nodes. When decrypting, receivers can directly use DPPs that arrived in plaintext form. In other words, DPPs are not one-time pad secret keys, nor are they secret parameters. Next, we introduce the specific process of SIT-SR, as shown in Figure 5.

3.2.1. Secret Key and Chaotic System Parameters. In order to avoid the secret key issues pointed out in some cryptanalysis works and simplify the generation process of chaotic system parameters [14, 41, 42], we set the secret key K in this stage as a binary sequence with the length of 270 bits. Namely, $K = a_1 a_2 \cdots a_{270}$. In specific implementation, we directly use nine 32-bit unsigned integers $b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9$ to generate three sets of parameters $(\hat{x}_0^{(1)}, \hat{y}_0^{(1)}, \hat{r}_0^{(1)})$, $(\hat{x}_0^{(2)}, \hat{y}_0^{(2)}, \hat{r}_0^{(2)})$, $(\hat{x}_0^{(3)}, \hat{y}_0^{(3)}, \hat{r}_0^{(3)})$ for 2D-LSCM. As shown in equation (11), this means that the 30×9 bits of K correspond to the 30 bits of each unsigned integer, respectively.

$$\begin{cases} \hat{x}_0^{(1)} = (b_1 \times 2 + 1) \times 2^{-32}, \\ \hat{y}_0^{(1)} = (b_2 \times 2 + 1) \times 2^{-32}, \\ \hat{r}_0^{(1)} = (b_3 \times 2 + 1) \times 2^{-32}, \\ \hat{x}_0^{(2)} = (b_4 \times 2 + 1) \times 2^{-32}, \\ \hat{y}_0^{(2)} = (b_5 \times 2 + 1) \times 2^{-32}, \\ \hat{r}_0^{(2)} = (b_6 \times 2 + 1) \times 2^{-32}, \\ \hat{x}_0^{(3)} = (b_7 \times 2 + 1) \times 2^{-32}, \\ \hat{y}_0^{(3)} = (b_8 \times 2 + 1) \times 2^{-32}, \\ \hat{r}_0^{(3)} = (b_9 \times 2 + 1) \times 2^{-32}, \end{cases} \tag{10}$$

where

$$\begin{cases} b_1 = a_1 a_2 \cdots a_{30}, \\ b_2 = a_{31} a_{32} \cdots a_{60}, \\ b_3 = a_{61} a_{62} \cdots a_{90}, \\ b_4 = a_{91} a_{92} \cdots a_{120}, \\ b_5 = a_{121} a_{122} \cdots a_{150}, \\ b_6 = a_{151} a_{152} \cdots a_{180}, \\ b_7 = a_{181} a_{182} \cdots a_{210}, \\ b_8 = a_{211} a_{212} \cdots a_{240}, \\ b_9 = a_{241} a_{242} \cdots a_{270}. \end{cases} \tag{11}$$

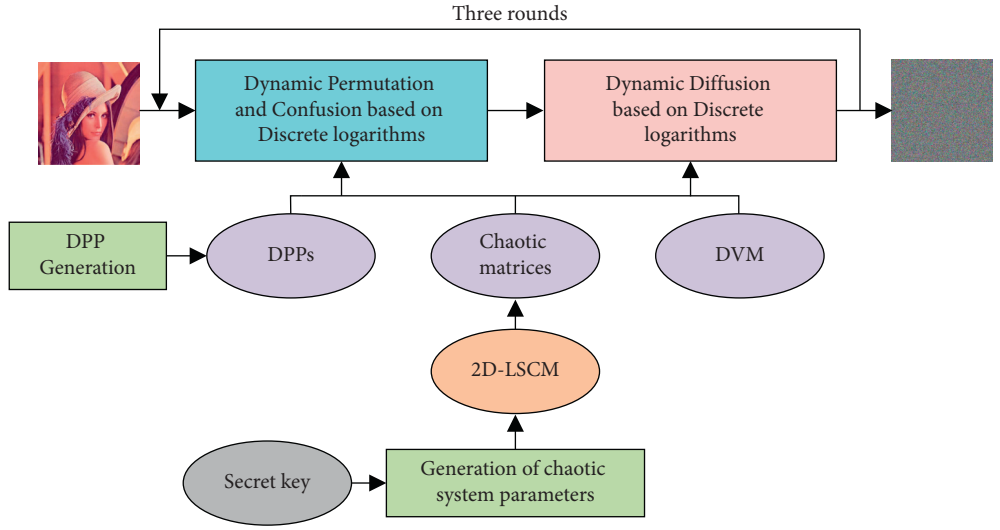


Figure 5: Flowchart of SIT-SR.

Besides, these three sets of chaotic system parameters $(\hat{x}_0^{(1)}, \hat{y}_0^{(1)}, \hat{r}^{(1)})$, $(\hat{x}_0^{(2)}, \hat{y}_0^{(2)}, \hat{r}^{(2)})$, $(\hat{x}_0^{(3)}, \hat{y}_0^{(3)}, \hat{r}^{(3)})$ are used to generate chaotic matrices for the encryption process.

3.2.2. DPP Generation. The generation process of DPPs in SIT-SR is exactly the same as SIT-SS. And we mark 32 DPPs used in SIT-SR as **H**.

3.2.3. Dynamic Permutation and Confusion Based on Discrete Logarithms. As mentioned above, discrete logarithms and DPPs are introduced in dynamic permutation and confusion based on discrete logarithms (DPC-D), so as to enhance the security of image transmission. Specifically, compared with some existing permutation operations, DPC-D has the following advantages:

(1) Use **H** to further perturb the permutation results and adopt different perturbation strategies for the row index and column index. Therefore, the permutation results depend not only on the secret key, but also on **H**.

(2) Based on discrete logarithms, **H** and the sorting results of the chaotic matrix **S** are used to nonlinearly transform the pixel value of each plain image pixel, thereby further improving the security of image transmission.

In order to better describe the specific steps of DPC-D, an algorithm is provided in Algorithm 1.

3.3. Dynamic Diffusion Based on Discrete Logarithms. To further improve security, dynamic diffusion based on discrete logarithms (DD-D) also adopts discrete logarithms and **H**. Specifically, compared with some existing diffusion operations, DD-D has the following advantages:

(1) Considering that multipixel diffusion is of little significance, single-pixel diffusion is adopted, thereby reducing the amount of computation

(2) The nonlinearity of the diffusion process is improved by introducing discrete logarithms; thus the security of image transmission is further improved

In order to better describe the specific steps of DD-D, an algorithm is provided in Algorithm 2.

Since a symmetric encryption structure is adopted in SIT-SR, the decryption process is actually constituted by the corresponding inverse operations of the encryption operations. With the received DPPs and the agreed secret key K, receivers can decrypt the plain image from the cipher image. To save space, these inverse operations are not repeated here.

4. Simulation Tests and Analyses

In this section, extensive simulation tests are performed to demonstrate the superiority of ITS-CDD. ITS-CDD is an image transmission scheme composed of two parts, and the resource conditions and design goals of each part are different. Therefore, SIT-SS is compared with 2DCS-ETC for resource-constrained environments, whereas SIT-SR is compared with more versatile schemes for general application environments. Without loss of generality, randomly generated secret keys are used to complete the tests. Table 2 lists the hardware and software configurations used in the tests.

4.1. Simulation Tests for SIT-SS. Since reducing the resource consumption of sensors and improving the security of image transmission is our goal in designing SIT-SS, the analysis and verification of SIT-SS are mainly focused on these two aspects. The test images used are eight images used in [12].

4.1.1. Encryption and Decryption. Four plain images Lena, Boats, House, and Parrots are shown in Figure 6. Their

Require: the plain image **P** with the size of $N \times N$, the chaotic matrix **S** with the size of $N \times N$, the dynamic perturbation parameters **H** with the size of 1×32 and the discrete logarithm value matrix **DV M** with the size of 128×256.

(1) Set $\mathbf{T} \in N^{N \times N}$;
(2) Set the sum hs of the dynamic perturbation parameters to 0;
(3) Set the row index value g used to represent the adopted generator to 0;
(4) Set the index value idx used to represent the adopted dynamic perturbation parameters to 0;
(5) Sort each column of **S** in ascending order, thus get the column index matrix **O** and sorted result **B**;
(6) **for** $i = 1$ to 32 **do**
(7) $\quad hs = hs + \mathbf{H}_i$;
(8) **end for**
(9) Calculate the row index value of the generator to be used, namely let $g = (hs \bmod 128) + 1$;
(10) **for** $i = 1$ to N **do**
(11) $\quad$ Sort $\mathbf{B}_i$ in ascending order and obtain the row index vector **v**;
(12) $\quad$ **for** $j = 1$ to N **do**
(13) $\quad\quad$ idx $= ((i-1) \times N + j \bmod 32) + 1$;
(14) $\quad\quad \mathbf{T}_{((\mathbf{O}_{i,j}+\mathbf{H}_{\text{idx}}) \bmod N)+1, ((j+hs) \bmod N)+1} = \mathbf{DVM}_{g, ((\mathbf{P}_{\mathbf{O}_{i,\mathbf{v}_j}, \mathbf{v}_j} + \mathbf{H}_{\text{idx}} + \mathbf{v}_j) \bmod 256)+1} - 1$;
(15) $\quad$ **end for**
(16) **end for**

Ensure: the permuted and transformed image **T**.

Algorithm 1: DPC-D algorithm.

Require: the permuted and transformed image **T** with the size of $N \times N$, the chaotic matrix **R** with the size of $N \times N$, the dynamic perturbation parameters **H** with the size of 1×32 and the discrete logarithm value matrix **DV M** with the size of 128×256.

(1) Set $\mathbf{C} \in N^{N \times N}$;
(2) Convert **R** into the integer matrix **IR** with the same format as the pixels of **T**, namely $\mathbf{IR} = (\lfloor \mathbf{R} \times 2^{32} \rfloor) \bmod 256$;
(3) Set the bitwise XOR result hx of the dynamic perturbation parameters to 0;
(4) Set the row index values g_1, g_2 used to represent the adopted generators to 0 s;
(5) **for** $i = 1$ to 32 **do**
(6) $\quad hx = \text{bitxor}(hx, \mathbf{H}_i)$;
(7) **end for**
(8) Calculate the row index values of the generators to be used, let $g_1 = (\text{bitxor}(hx, \mathbf{IR}_{1,1}) \bmod 128) + 1$, $g_2 = (\text{bitxor}(g_1, \mathbf{IR}_{N,N}) \bmod 128) + 1$;
(9) $\mathbf{tmp}_{:,1} = (\mathbf{T}_{:,1} + \mathbf{DVM}_{g_1, (\mathbf{T}_{:,N}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{:,1}+1)}) \bmod 256$;
(10) **for** $i = 2$ to N **do**
(11) $\quad \mathbf{tmp}_{:,i} = (\mathbf{T}_{:,i} + \mathbf{DVM}_{g_1, (\mathbf{tmp}_{:,i-1}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{:,i}+1)}) \bmod 256$;
(12) **end for**
(13) $\mathbf{C}_{1,:} = (\mathbf{tmp}_{1,:} + \mathbf{DVM}_{g_1, (\mathbf{tmp}_{N,:}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{1,:}+1)}) \bmod 256$;
(14) **for** $i = 2$ to N **do**
(15) $\quad \mathbf{C}_{i,:} = (\mathbf{tmp}_{i,:} + \mathbf{DVM}_{g_1, (\mathbf{C}_{i-1,:}+1)} + \mathbf{DVM}_{g_2, (\mathbf{IR}_{i,:}+1)}) \bmod 256$;
(16) **end for**

Ensure: the diffused image **C**.

Algorithm 2: DD-D algorithm.

corresponding cipher images and decrypted images generated in SIT-SS are also provided. As can be seen from these images, the cipher images are similar to noise, attackers cannot obtain useful information from them, and there are no significant visual differences between the decrypted images and corresponding plain images.

4.1.2. Reconstruction Quality. Researchers often use Peak Signal-to-Noise Ratio (PSNR) to evaluate image reconstruction quality. Generally, a higher PSNR value indicates a better reconstruction quality. The definition of PSNR is as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{(1/(M \times N)) \sum_{i=1}^{M} \sum_{i=1}^{N} [R(i,j) - P(i,j)]^2}, \tag{12}$$

where $M \times N$ is the size of the reconstructed image R and original image P. PSNR versus sampling ratio for 2DCS-ETC and SIT-SS is listed in Table 3. As can be seen from Table 3, SIT-SS can achieve the same or slightly different PSNR

Table 2: Software and hardware configurations used in simulation tests.

Configuration item	Description
CPU	Intel Xeon CPU E3-1231 v3 3.40 GHz
Memory capacity	8 GB
Operating system	Windows 7 Ultimate (64 bit)
Test platform	MATLAB R2017a (9.2.0538062)

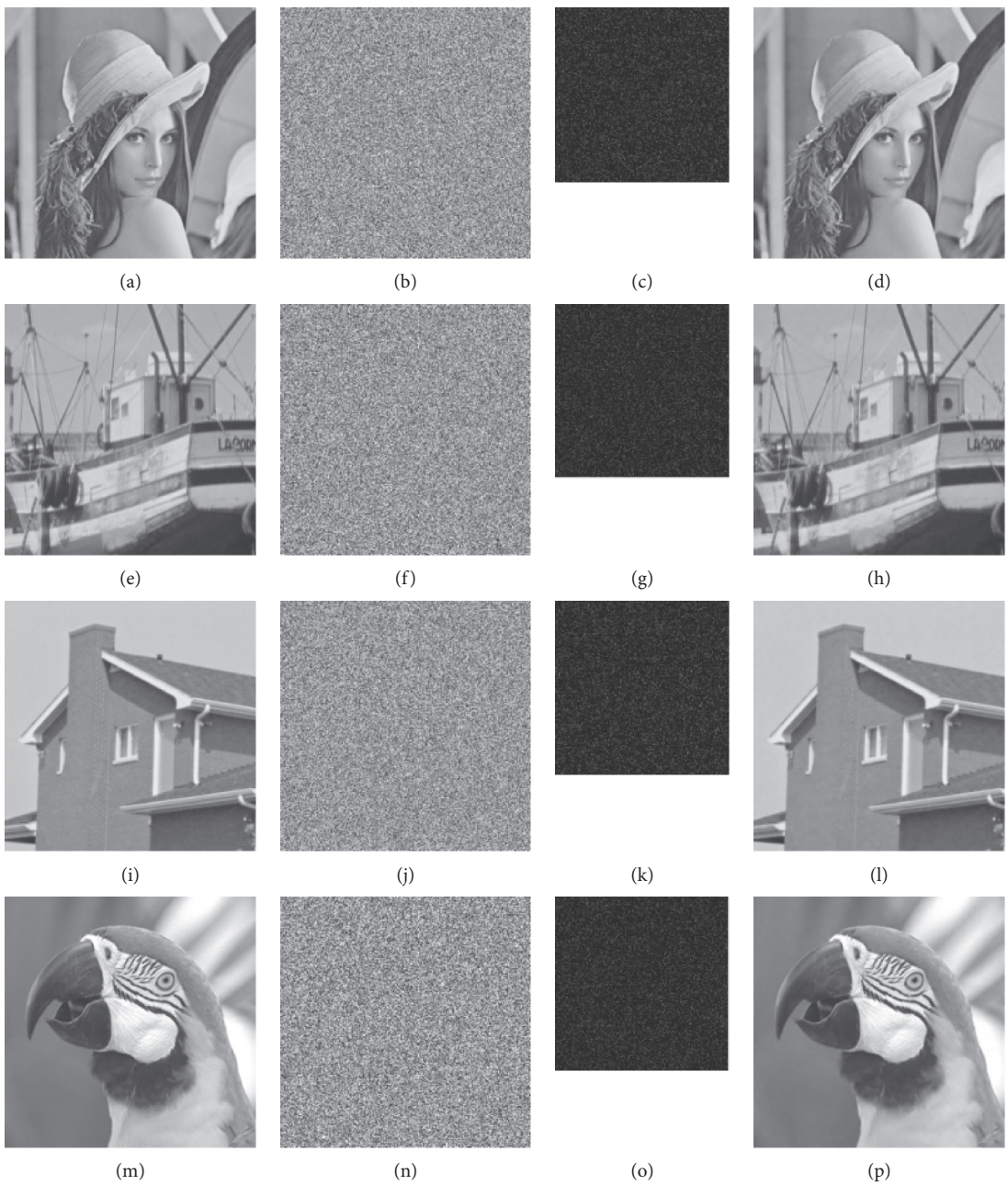


Figure 6: Encryption and reconstruction of four plain images: (a) plain image Lena; (b) cipher image of (a); (c) compressed cipher image of (a); (d) reconstructed image of (c); (e) plain image of Boats; (f) cipher image of (e); (g) compressed cipher image of (e); (h) reconstructed image of (g); (i) plain image of House; (j) cipher image of (i); (k) compressed cipher image of (i); (l) reconstructed image of (k); (m) plain image of Parrots; (n) cipher image of (m); (o) compressed cipher image of (m); (p) reconstructed image of (o).

values as 2DCS-ETC. And when sampling ratio is greater than 0.6, its image reconstruction quality has obvious advantages.

4.1.3. Secret Key. In 2DCS-ETC, the random permutation matrix and random binary integer matrix are used as the secret key, thereby obtaining a huge key space. However, in

Table 3: PSNR (dB) of 2DCS-ETC and SIT-SS under different sampling ratios.

Image	Scheme	Sampling ratio								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Lena	2DCS-ETC	25.1608	**30.7238**	**33.7894**	36.3407	38.5169	39.0429	40.3088	41.1759	41.7017
	SIT-SS	**25.5232**	30.6675	33.7680	**36.3926**	**38.6439**	**40.4935**	**42.6165**	**44.9562**	**47.9543**
Barbara	2DCS-ETC	22.0135	**25.8589**	**28.8967**	**31.7166**	**34.2147**	35.6315	37.3399	39.0916	40.2334
	SIT-SS	**22.1990**	25.8260	28.8175	31.6593	34.1841	**36.4853**	**38.8797**	**41.5256**	**45.0071**
Boats	2DCS-ETC	**24.6290**	**29.7913**	32.8112	35.3011	37.4047	38.0778	39.1809	40.3438	41.5691
	SIT-SS	24.1095	29.6903	**32.8657**	**35.3405**	**37.4213**	**39.3371**	**41.3561**	**43.5876**	**46.6325**
Cameraman	2DCS-ETC	**22.3125**	**28.7665**	**31.8389**	**34.2630**	**36.4953**	36.9751	38.2231	39.0237	39.7931
	SIT-SS	21.8324	28.7046	31.8201	34.2575	36.4449	**38.6534**	**40.9500**	**43.3287**	**46.3889**
Foreman	2DCS-ETC	29.5930	**35.8537**	**38.1558**	38.5413	39.4111	39.8570	40.2641	40.3616	40.6835
	SIT-SS	**29.7572**	35.7866	38.1025	**39.8524**	**41.4978**	**42.9477**	**44.5395**	**46.3175**	**48.4910**
House	2DCS-ETC	**28.9899**	**34.1209**	36.0911	**37.5883**	38.9428	39.1408	40.0489	41.1202	42.1401
	SIT-SS	28.3764	34.1188	**36.1386**	37.5654	**38.9717**	**40.3546**	**42.0492**	**44.1757**	**47.0669**
Monarch	2DCS-ETC	**22.3624**	**29.0054**	32.5729	**35.6175**	**37.9890**	38.1359	39.3164	40.0515	40.9473
	SIT-SS	22.0071	28.8123	**32.6323**	35.3864	37.8005	**39.9611**	**42.1377**	**44.5657**	**47.4934**
Parrots	2DCS-ETC	**25.9860**	**33.0968**	**35.7941**	**37.8344**	38.0873	39.1139	39.7870	40.3589	40.6380
	SIT-SS	25.7881	32.9618	35.6443	37.7954	**39.5943**	**41.2956**	**43.0655**	**45.0295**	**47.6101**

Each bold value means that one of the two compared schemes has a higher PSNR value than the other.

resource-constrained environments, it is not suitable to use such secret key that requires a large amount of storage space. For example, if the size of the plain image is 1024×1024, the secret key used would be at least 2 228 224 bytes in length. In addition, in the encryption and decryption process, the generation and storage of two measurement matrices also bring significant resource requirements. However, in SIT-SS, we only need to store six floating-point numbers used as the secret key. Meanwhile, SIT-SS also enjoys a large enough key space, which is about 10^{133}. Apparently, such a large key space is sufficient to resist brute force attacks.

4.1.4. Chosen-Plaintext Attack. As we know, CPAs are the reasons why some encryption schemes are cracked. It is generally believed that a secure encryption scheme should be able to resist CPAs. Derived from system times and last encryption times, DPPs always change dynamically and will be affected by many factors, thereby ensuring the diversity and unpredictability of equivalent key streams. Without relying on external algorithms such as hash algorithms, the diversity of equivalent key streams brings excellent resistance to CPAs.

4.2. Simulation Tests for SIT-SR. The simulation tests presented in this subsection are to demonstrate the superiority of SIT-SR in terms of security and encryption efficiency. The test images used for SIT-SR are from The USC-SIPI Image Database (https://sipi.usc.edu/database/).

4.2.1. Encryption and Decryption of Different Types of Images. For different types of images, we uniformly treat them as 8-bit grayscale images in ITS-CDD. Specifically, for images with a pixel depth of less than 8 bits, we directly process them as 8-bit grayscale images, whereas for images with pixel depth greater than 8 bits, we encrypt them in groups of 8 bits. For example, if we need to encrypt an image with a pixel depth of 16 bits, we can encrypt the lower 8 bits and higher 8 bits of each plain image pixel separately. Figure 7 shows the encryption and decryption effects of SIT-SR for different types of images. One can see that SIT-SR has excellent encryption effects for different types of images. The generated cipher images are very similar to noise images, and attackers cannot directly obtain any valuable information from these cipher images.

4.2.2. Key Space and Key Sensitivity. Since the key space would affect the ability to resist brute force attacks, a secure encryption scheme should have a sufficiently large key space [50]. We carefully design the secret key, which not only solves the issues about equivalent secret keys, but also expands the key space to 2^{270}. Therefore, SIT-SR is excellent in terms of the ability to resist brute force attacks.

It is well known that a secure encryption scheme should be able to achieve superior confusion effect [50, 51]. That is, one smallest change in the secret key should make the cipher image change significantly. To evaluate the performance of SIT-SR in this regard, we randomly generated the secret key

$$K_R = 03D7\ D6F3E884\ 829564ED\ BA77D868\ 3B811AE4\ FC8655CC\ 7EE7BC30\ 5537BFEA\ 2EE2238E. \quad (13)$$

Using K_R, we encrypted elaine.512.tiff to obtain the corresponding cipher image $\check{C}_R$. Next, we changed one bit of K_R to get two secret keys $K_R^{(1)}$, $K_R^{(2)}$ with single smallest changes. These two changed secret keys also were used to encrypt elaine.512.tiff, thus obtaining the corresponding cipher images $\check{C}_R^{(1)}$, $\check{C}_R^{(2)}$. Finally, the difference images between $\check{C}_R^{(1)}$, $\check{C}_R^{(2)}$, and $\check{C}_R$ were calculated. As can be seen from Figure 8, the difference images between $\check{C}_R^{(1)}$, $\check{C}_R^{(2)}$, and $\check{C}_R$ look similar to an ordinary cipher image. This means that the key sensitivity of SIT-SR in the encryption process is extraordinary.

Similarly, in order to verify the key sensitivity of SIT-SR in the decryption process, $K_R^{(1)}$ and $K_R^{(2)}$ were adopted to

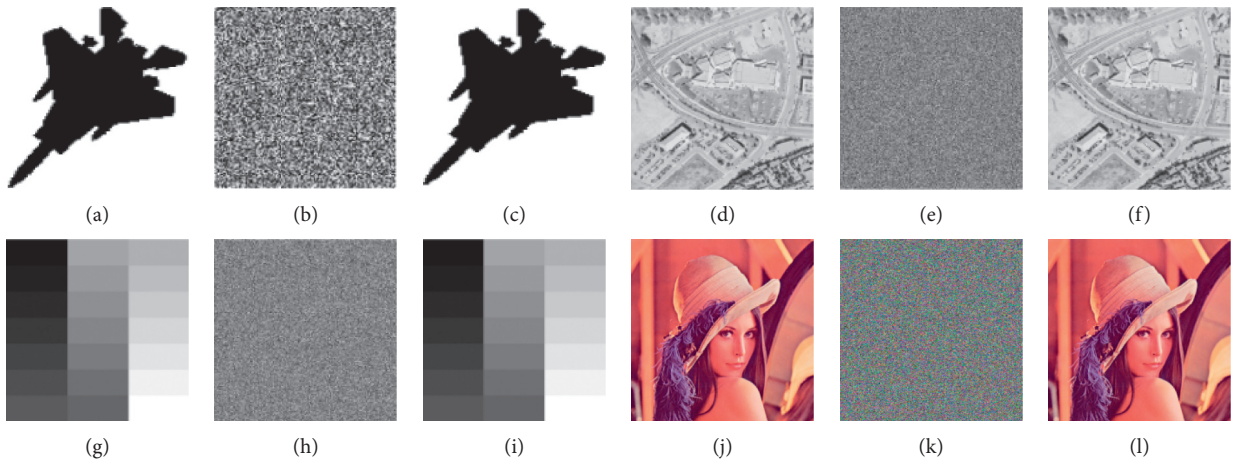


Figure 7: Encryption and decryption effects for different types of images: (a) f15.png; (b) cipher image of (a); (c) decrypted image of (b); (d) 5.2.09.tiff; (e) cipher image of (d); (f) decrypted image of (e); (g) gray21.512.tiff; (h) cipher image of (g); (i) decrypted image of (h); (j) 4.2.04.tiff; (k) cipher image of (j); (l) decrypted image of (k); (m) f15.png; (n) cipher image of (m); (o) decrypted image of (n); (p) f15.png; (q) cipher image of (p); (r) decrypted image of (q); (s) f15.png; (t) cipher image of (s); (u) decrypted image of (t); (v) f15.png; (w) cipher image of (v); (x) decrypted image of (w).

decrypt $\check{C}_R$. The test results are shown in Figure 9. Once again, judging from the difference image between the wrongly decrypted images, the key sensitivity of SIT-SR in the decryption process is excellent.

For measuring the degree of changes between images, NPCR (Number of Pixel Change Ratio) and UACI (Unified Average Change in Intensity) are commonly used indicators [13]. Among them,

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i,j} \mathbf{D}_{i,j} \times 100\%, \tag{14}$$

refers to the ratio of the pixels that change, whereas

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i,j} \frac{\left|\mathbf{C}_{i,j}^{(1)} - \mathbf{C}_{i,j}^{(2)}\right|}{255} \times 100\%, \tag{15}$$

refers to the average intensity of the pixel value changes. In equations (14) and (15), $M \times N$ is the size of the images, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, and $\mathbf{D}$ is the difference matrix between the image $\mathbf{C}^{(1)}$ before the change and the image $\mathbf{C}^{(2)}$ after the change. If $\mathbf{C}_{i,j}^{(1)} \neq C_{i,j}^{(2)}$, then $\mathbf{D}_{i,j} = 1$. Otherwise, $\mathbf{D}_{i,j} = 0$. In order to quantitatively analyze the key sensitivity of SIT-SR, we calculated the NPCR and UACI values between the cipher images before and after the secret key changes. As one can see from Table 4, all the test results are very close to the ideal values, which means that SIT-SR does have extremely high key sensitivity.

4.2.3. Pixel Value Distribution. Unlike plain images, cipher images must have extremely uniform pixel value distributions; otherwise attackers will have the opportunity to conduct statistics based attacks [50, 51]. In order to visually demonstrate the pixel value distribution characteristics of the plain images and the cipher images generated by SIT-SR, the pixel value distribution histograms of these images are plotted. As can be seen from Figure 10, the pixel distributions of the plain images are relatively concentrated, whereas the pixels of the cipher images are extremely uniformly distributed throughout the entire value range.

In addition, the histogram variance analysis and chi-square test analysis are also performed on the cipher images to qualitatively analyze the ability of SIT-SR to resist statistical attacks. In general, if the histogram variance of a cipher image is smaller, then the uniformity of the cipher image is higher. The histogram variance of an 8-bit grayscale image can be defined as follows:

$$V(\mathbf{H}) = \frac{1}{256^2} \sum_{i=1}^{256} \sum_{j=1}^{256} \frac{1}{2}\left(p_i - p_j\right)^2, \tag{16}$$

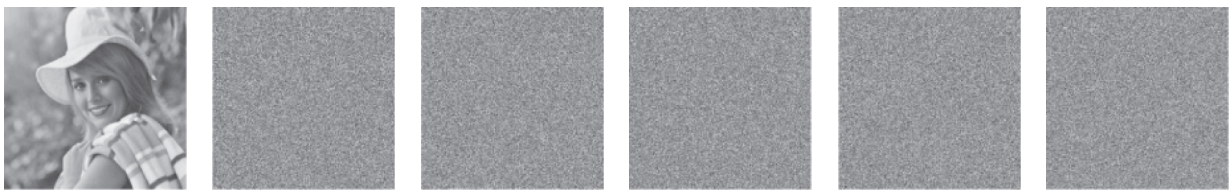
where $\mathbf{H} = \{p_1, p_2, \ldots, p_{256}\}$; p_i and p_j are the numbers of the pixels whose grayscale values are equal to $i - 1$ and $j - 1$. Table 5 lists the histogram variance test results of some images. These images include one random image, the 8-bit grayscale image Lena, and its cipher images generated by different schemes.

From Table 5, one can see that the histogram variance of the plain image is very large, which means that its pixel value distribution is extremely uneven, whereas among the cipher images generated by the four image encryption schemes, the cipher image of SIT-SR has the smallest histogram variance, which indicates that this cipher image has the most uniform pixel value distribution and is closest to the random image.

Another way to quantitatively analyze the uniformity of a cipher image is the chi-square test. The chi-square value of a cipher image can be calculated as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{\left(s_i - H \times W \times p\right)^2}{H \times W \times p}, \tag{17}$$

where $H \times W$ is the height and width of the cipher image, s_i is the number of pixels whose pixel value is $i - 1$, n is the



(a) (b) (c) (d) (e) (f)

Figure 8: Key sensitivity test for encryption process: (a) elaine.512.tiff; (b) $\check{C}_R$ obtained by K_R; (c) $\check{C}_R^{(1)}$ obtained by $K_R^{(1)}$; (d) $\check{C}_R^{(2)}$ obtained by $K_R^{(2)}$; (e) difference image between $\check{C}_R$ and $\check{C}_R^{(1)}$; (f) difference image between $\check{C}_R$ and $\check{C}_R^{(2)}$.

(a) (b) (c) (d) (e)

Figure 9: Key sensitivity test for decryption process: (a) $\check{C}_R$ obtained by K_R; (b) decrypted image $\check{C}_R$ obtained by K_R; (c) decrypted image $\check{C}_R^{(1)}$ obtained by $K_R^{(1)}$; (d) decrypted image $\check{C}_R^{(2)}$ obtained by $K_R^{(2)}$; (e) difference image between $\check{C}_R^{(1)}$ and $\check{C}_R^{(2)}$.

Table 4: UPCR and UACI values between cipher images when secret key changes.

Change	NPCR (%)	UACI (%)
Lowest bit of b_1 is inverted	99.6121	33.4601
Lowest bit of b_2 is inverted	99.6028	33.4625
Lowest bit of b_3 is inverted	99.6097	33.4837
Lowest bit of b_4 is inverted	99.6013	33.4633
Lowest bit of b_5 is inverted	99.6009	33.4764
Lowest bit of b_6 is inverted	99.6174	33.4512
Lowest bit of b_7 is inverted	99.6075	33.4810
Lowest bit of b_8 is inverted	99.6122	33.4706
Lowest bit of b_9 is inverted	99.6130	33.4562
Random image	**99.6094**	**33.4635**

The bold values are the ideal values.

number of all possible pixel values (for 8-bit grayscale images, $n = 256$), and $p = 1/n$. Next, the critical value $\chi^2_{0.05}(255)$ of the chi-square test at the significant level $\alpha = 0.05$ can be determined, which is 293.2478. If a cipher image has a chi-square value less than 293.2478, then this image can be considered to have passed the chi-square test; that is, its pixel value distribution is statistically uniform. Consequently, the smaller the chi-square value of a cipher image is, the better its uniformity is. As can be seen from Table 6, for some commonly used test images, the cipher images generated by SIT-SR have all passed the chi-square test. And in most cases, SIT-SR performs better than another scheme.

4.2.4. Plain Image Sensitivity. When the plain image changes, the corresponding change degree of the cipher image is plain image sensitivity. Generally speaking, to effectively resist differential attacks, an image encryption scheme must have excellent plain image sensitivity. To intuitively show the plain image sensitivity of SIT-SR, we first encrypted some commonly used test images. Next, after changing one pixel bit for each plain image, the plain images with the smallest changes were encrypted. At last, we calculated the difference images between the cipher images obtained before and after the smallest changes. The relevant test results are shown in Figure 11. As one can see from Figure 11, each plain image has undergone only one smallest change, but almost all cipher pixels have changed. In addition to that, these significant differences are independent of where the plain images change and are very close to random images.

UPCR and UACI are also utilized to qualitatively analyze the plain image sensitivity of SIT-SR. The UPCR and UACI values between the cipher images obtained before and after the smallest changes of 15 common test plain images are shown in Table 7. Judging from the test results, SIT-SR has excellent plain image sensitivity. The test results of SIT-SR are closer to the ideal values 99.6094% and 33.4635% and perform better in terms of stability.

4.2.5. Information Entropy. Information entropy is an indicator that can be used to measure the randomness or disorder of an information source. If the information

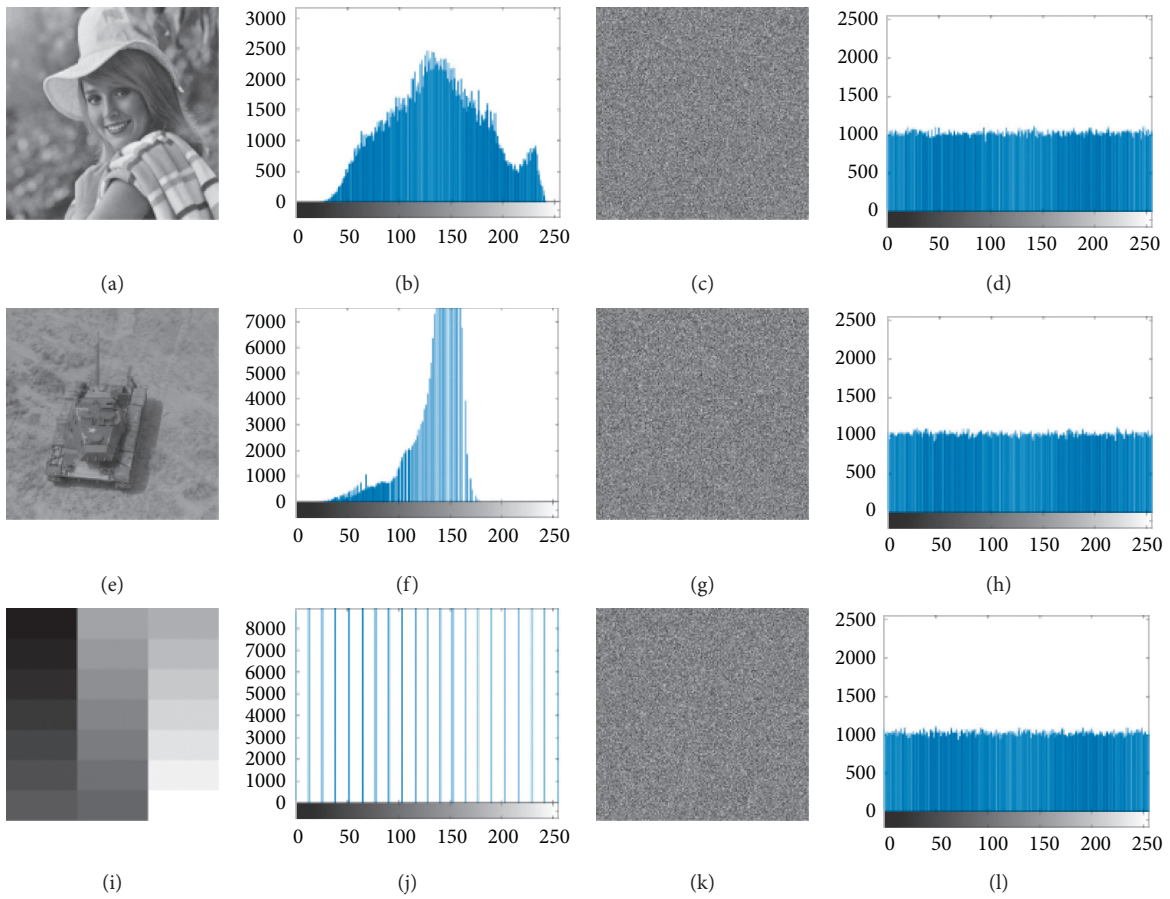


Figure 10: Pixel value distribution histograms of elaine.512.tiff, 7.1.03.tiff, gray21.512.tiff, and their corresponding cipher images: (a) elaine.512.tiff; (b) histogram of (a); (c) cipher image of (a); (d) histogram of (c); (e) 7.1.03.tiff; (f) histogram of (e); (g) cipher image of (e); (h) histogram of (g); (i) gray21.512.tiff; (j) histogram of (i); (k) cipher image of (i); (l) histogram of (k).

Table 5: Histogram variance test results of some images.

Algorithm	Image	Variance
	Lena256.bmp	33860.0546
[13]	Cipher image	266.7578
[16]	Cipher image	260.7188
[17]	Cipher image	276.3906
SIT-SR	Cipher image	**257.1094**
	Random image	253.8946

The bold value means that SIT-SR has the best test result among four compared schemes.

entropy of the information source is higher, it can be considered that the information source has higher randomness or disorder [18–20]. When it comes to an 8-bit grayscale image, the information entropy of the grayscale image can be calculated as follows:

$$H(\mathbf{S}) = -\sum_{i=1}^{n} \left(p(s_i) \times \log_2\ p(s_i)\right), \tag{18}$$

where n is the total number of symbols s_i; $p(s_i)$ is the occurrence probability of symbol s_i. For the 8-bit grayscale cipher images, the ideal value of the information entropy is 8 [18–20]. From Table 8, one can see that the information entropy of each cipher image generated by SIT-SR is very close to the ideal value 8. As shown in Table 9, compared with several image encryption schemes, the information entropy of the Lena cipher image generated by SIT-SR is closest to the ideal value 8. Therefore, SIT-SR performs best in terms of the information entropy.

In order to measure the randomness of cipher images more comprehensively, a measure named Local Shannon Entropy (LSE) is proposed [52]. This measure is increasingly adopted to verify the randomness of cipher images [13]. Mathematically, LSE can be defined as follows:

$$L_{q,s}(r) = \sum_{i=1}^{q} \frac{H(r_i)}{q}, \tag{19}$$

Table 6: Chi-square values of different cipher images.

Filename	Size	Chi-square value	
		[13]	SIT-SR
Lena256.bmp	256 × 256	255.8555	253.3035
5.1.10.tiff	256 × 256	261.3125	254.1953
5.1.12.tiff	256 × 256	256.2578	244.5328
5.1.13.tiff	256 × 256	274.8750	245.3797
5.2.08.tiff	512 × 512	252.7471	247.5434
5.2.09.tiff	512 × 512	274.3906	257.3434
5.3.01.tiff	1024 × 1024	236.3027	229.8125
7.1.02.tiff	512 × 512	252.9141	226.2197
7.1.03.tiff	512 × 512	248.8984	257.2324
7.1.04.tiff	512 × 512	281.2773	258.4043
7.1.05.tiff	512 × 512	275.1055	263.8584
boat.512.tiff	512 × 512	230.2256	232.7012
elaine.512.tiff	512 × 512	266.6377	230.0078
gray21.512.tiff	512 × 512	244.8789	245.3027
ruler.512.tiff	512 × 512	290.8057	223.2813
testpat.1k.tiff	1024 × 1024	258.6455	239.7627

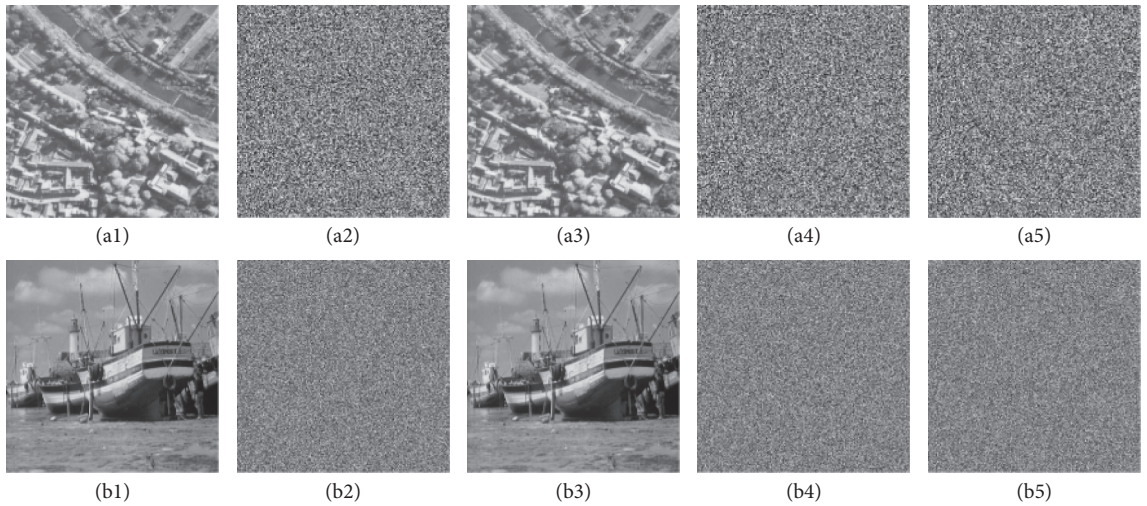


Figure 11: Plain image sensitivity test results for SIT-SR: (a1) 5.1.10.tiff; (a2) cipher image of (a1); (a3) the least significant bit of the pixel at (1,1) in (a1) is reversed; (a4) cipher image of (a3); (a5) difference image between (a2) and (a4); (b1) boat.512.tiff; (b2) cipher image of (b1); (b3) the least significant bit of the pixel at (256,256) in (b1) is reversed; (b4) cipher image of (b3); (b5) difference image between (b2) and (b4).

where $r_1, r_2, \ldots, r_q$ are q randomly selected nonoverlapping image blocks, s is the number of pixels in each block, and $H(r_i)$ is the information entropy of r_i. According to the test method suggested in [52], we carried out the LSE test on the cipher images generated by SIT-SR, and the relevant test results are shown in Table 10. Compared with two recent image encryption schemes, SIT-SR has the best performance in terms of standard deviation and pass rate.

4.2.6. Pixel Correlation. The extremely high correlation between adjacent pixels is one of the salient features of plain images and also one of the important reasons why traditional encryption schemes are not suitable for image encryption [50]. Therefore, a secure image encryption scheme should eliminate the correlation between adjacent pixels as much as possible. CC (correlation coefficient) is an effective indicator to measure the correlation between adjacent pixels, and its definition is as follows:

$$\mathrm{CC}_{ab} = \frac{E((a - E(a)) \times (b - E(b)))}{\sqrt{D(a) \times D(b)}}, \tag{20}$$

where a and b are the grayscale values of two adjacent pixels; $E(a)$ and $D(a)$ are the expectation and variance of the grayscale value a. In order to verify the performance of SIT-SR in terms of the pixel correlation, for the horizontal, vertical, and diagonal directions, we have randomly selected 20,000 pairs of adjacent pixels from each plain image and its corresponding cipher image to calculate the CCs. The relevant test results are shown in Table 11.

From Table 11, one can see that there are very high correlations between adjacent pixels of the plain images; that is, the absolute values of CCs are extremely high, whereas in

Table 7: NPCR and UACI test results on plain image sensitivity.

Filename	[13]		SIT-SR	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
5.1.10.tiff	99.6014	33.4774	99.6162	33.4645
5.1.12.tiff	99.6222	33.4668	99.6093	33.4642
5.1.13.tiff	99.6091	33.4782	99.6119	33.4597
5.2.08.tiff	99.6138	33.4596	99.6080	33.4589
5.2.09.tiff	99.6072	33.4496	99.6117	33.4521
5.3.01.tiff	99.6103	33.4551	99.6107	33.4595
7.1.02.tiff	99.6088	33.4749	99.6140	33.4635
7.1.03.tiff	99.6026	33.4930	99.6081	33.4836
7.1.04.tiff	99.6117	33.4699	99.6025	33.4645
7.1.05.tiff	99.6101	33.4766	99.6083	33.4643
boat.512.tiff	99.6045	33.4618	99.6098	33.4825
elaine.512.tiff	99.6139	33.4918	99.6098	33.4521
gray21.512.tiff	99.6069	33.4660	99.6122	33.4713
ruler.512.tiff	99.6113	33.4394	99.6115	33.4453
testpat.1k.tiff	99.6054	33.4571	99.6091	33.4624
Average	99.6080	33.4695	**99.6113**	**33.4598**
Std. Dev.	0.0042	0.0161	**0.0028**	**0.0045**

The bold values here emphasize that SIT-SR has better performance than the other scheme.

Table 8: Information entropy test results of plain images and cipher images.

Filename	Plain image	Cipher image
5.2.08.tiff	7.2010	7.9993
5.2.09.tiff	6.9940	7.9994
5.3.01.tiff	7.5237	7.9998
7.1.02.tiff	4.0045	7.9993
7.1.03.tiff	5.4957	7.9994
7.1.04.tiff	6.1074	7.9993
7.1.05.tiff	6.5632	7.9994
boat.512.tiff	7.1914	7.9994
elaine.512.tiff	7.5060	7.9994
gray21.512.tiff	4.3923	7.9993
ruler.512.tiff	0.5000	7.9994
testpat.1k.tiff	4.4077	7.9998

Table 9: Information entropy test results of Lena cipher images.

Scheme	[13]	[18]	[19]	[20]	SIT-SR
Information entropy	7.9992	7.9979	7.9909	7.9991	**7.9994**

The bold value here emphasizes that SIT-SR has better performance than other schemes.

the cipher images generated by SIT-SR, there is almost no correlation between adjacent pixels; that is, the absolute values of CCs are extremely low (< 0.006).

In addition, in order to more intuitively show the correlation changes between adjacent pixels caused by the encryption of SIT-SR, the correlation distribution charts of the plain image elaine.512.tiff and its corresponding cipher image are drawn. As can be seen from Figure 12, after the encryption processing of SIT-SR, there is almost no correlation between adjacent pixels in each direction.

4.2.7. Chosen-Plaintext Attack. In fact, almost all simulation tests related to security analysis can only ensure the security of image encryption schemes under ciphertext-only attacks (COAs) [51, 53]. This is exactly why some image encryption schemes have been broken. Among the four types of attacks, which are COAs, Known-Plaintext Attacks (KPAs), CPAs, and Chosen-Ciphertext Attacks (CCAs), CCAs are the most threatening ones, but the attack conditions required by them are practically meaningless [3, 50]. If attackers can choose cipher images arbitrarily, then they do not need to crack at all, because for any cipher image, they can directly recover its plain image. Therefore, it is generally believed that CPAs are the most threatening ones among common practical attacks. Actually, in the cryptanalysis works about image encryption, the vast majority of them adopt CPAs [3, 51]. Next, from the perspective of attackers, the ability of SIT-SR to resist CPAs is analyzed.

Table 10: LSE test results of different schemes.

Filename	[13]	[15]	SIT-SR
5.2.08	7.9023	7.9024	7.9022
5.2.09	7.9020	7.9021	7.9023
7.1.02	7.9020	7.9015	7.9021
7.1.03	7.9026	7.9019	7.9024
7.1.04	7.9019	7.9021	7.9023
boat.512	7.9018	7.9022	7.9024
gray21.512	7.9026	7.9026	7.9025
ruler.512	7.9041	7.9028	7.9026
Std.Dev.	0.0007	0.0004	**0.0002**
Pass/All	6/8	7/8	**8/8**

The bold values indicate that compared with the other two schemes, SIT-SR has the best performance in terms of standard deviation and pass rate.

Apparently, attackers will encounter several problems when they try to break SIT-SR with CPAs. Firstly, we assume that they could obtain the equivalent key streams of the encryption process from chosen plain images and corresponding cipher images. However, because SIT-SR introduce DPPs in serval encryption steps, the equivalent key streams they obtained cannot be used to recover other ordinary plain images, which are encrypted under different DPPs. Secondly, SIT-SR also performs nonlinear

Table 11: Correlation test results for adjacent pixels of plain images and their cipher images.

Filename	Horizontal		Vertical		Diagonal	
	Plain image	Cipher image	Plain image	Cipher image	Plain image	Cipher image
5.2.08.tiff	0.8906	0.0008	0.9322	−0.0038	0.8452	0.0028
5.2.09.tiff	0.8591	−0.0004	0.9000	−0.0018	0.8007	0.0020
5.3.01.tiff	0.9817	0.0015	0.9776	0.0044	0.8981	−0.0033
7.1.02.tiff	0.9446	−0.0024	0.9431	−0.0016	0.9003	0.0001
7.1.03.tiff	0.9317	−0.0021	0.9436	0.0002	0.9059	−0.0017
7.1.04.tiff	0.9672	0.0038	0.9771	0.0028	0.9552	−0.0059
7.1.05.tiff	0.9108	0.0038	0.9425	0.0058	0.8919	0.0024
boat.512.tiff	0.9711	0.0030	0.9394	0.0048	0.9245	0.0002
elaine.512.tiff	0.9720	−0.0011	0.9761	0.0010	0.9696	0.0016
gray21.512.tiff	0.9998	−0.0023	0.9968	−0.0036	0.9966	0.0031
ruler.512.tiff	0.4702	−0.0026	0.4524	0.0004	−0.0312	−0.0023
testpat.1k.tiff	0.7992	0.0035	0.7501	0.0051	0.6997	0.0005

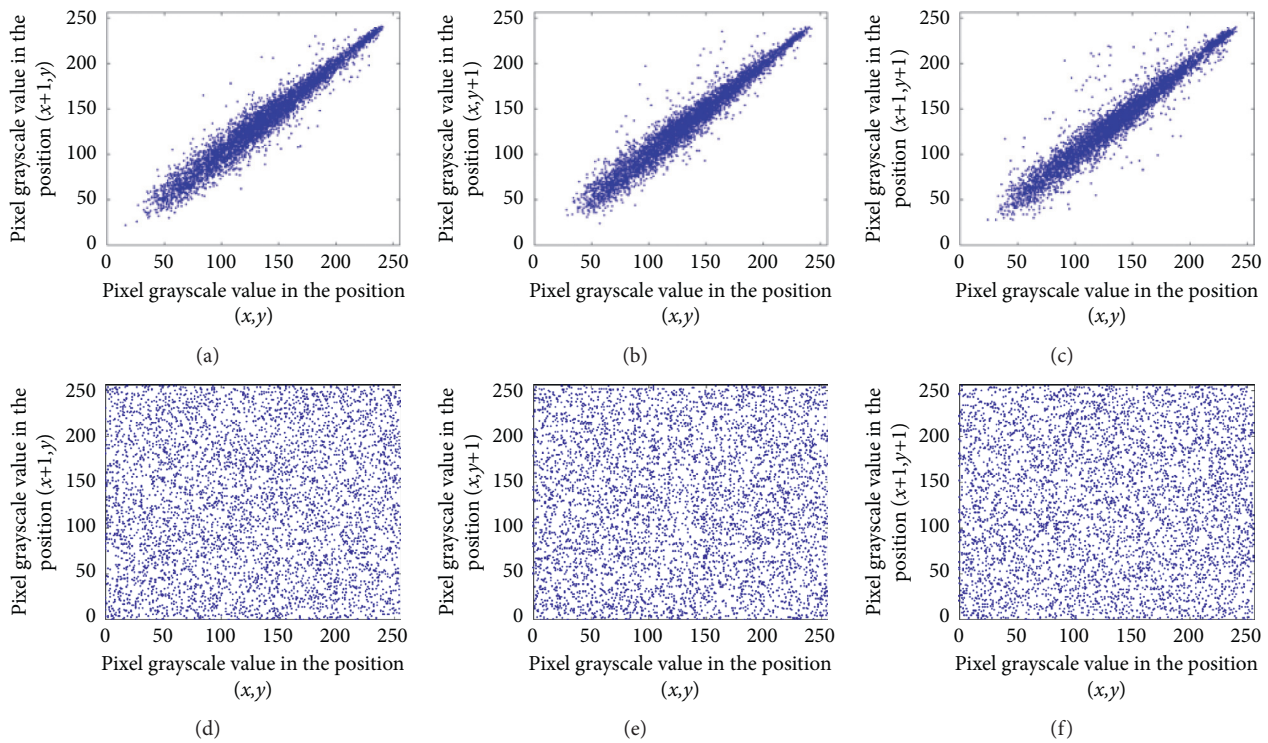


Figure 12: Pixel correlation distribution charts of elaine.512.tiff and its cipher image: (a, d) distribution charts in horizontal direction; (b, e) distribution charts in vertical direction; (c, f) distribution charts in diagonal direction.

transformations on the plain image pixels during the permutation process, so the common attack method that ignores the permutation process by the chosen plain images composed of single-value pixels cannot work. Thirdly, SIT-SR adopts a nonlinear diffusion structure; that is, it adopts the discrete logarithms based on two different generators, which makes the encryption process cannot be simplified by chosen plain images. To sum up, SIT-SR can effectively resist CPAs.

4.2.8. Encryption Efficiency. Improving encryption efficiency is one of the most important motivations to design new image encryption schemes. SIT-SR introduce DPPs and discrete logarithms, but in fact, discrete logarithms can be calculated in advance, and the calculation process of DPP is very simple, so the impact on encryption efficiency is very small. In addition, SIT-SR uses single-pixel diffusion and only performs three iterations in the encryption process. These also help to reduce the total number of primitive operations that need to be executed. Table 12 shows the average time required by SIT-SR and some other recent image encryption schemes to encrypt the 8-bit grayscale image Lena (256×256). As can be seen from Table 12, although the time complexity of each image encryption scheme is $O(M \times N)$, the scheme proposed in [13] requires the least number of primitive operations, so it has the

Table 12: Comparison of test results obtained by different encryption schemes to encrypt Lena image.

Scheme	SIT-SR	[13]	[18]	[19]	[21]	[22]
Time (s)	**0.089**	0.072	0.417	0.683	0.275	0.264
Throughput (Mbps)	**5.6180**	6.9444	1.1990	0.7321	1.8182	1.8939

The bold values here emphasize that SIT-SR has better performance than other schemes.

highest encryption efficiency, whereas for SIT-SR, it adds a certain number of primitive operations to ensure the security, but it still maintains the significant advantage of high encryption efficiency. That is, in terms of encryption efficiency, SIT-SR is significantly better than the remaining four image encryption schemes.

5. Conclusions

In order to improve the efficiency and security of image transmission, an image transmission scheme based on two chaotic maps is proposed in this paper. The proposed scheme divides the image transmission from sensor nodes to receivers into two stages and carries out a targeted design, which can better adapt to heterogeneous application environments. For image transmission between sensor nodes and sink nodes, the proposed scheme reduces the requirements for hardware resources and improves the image reconstruction quality by introducing a lightweight chaotic map. Besides, the design of dynamic perturbation improves the security of image transmission at this stage, whereas for image transmission between sink nodes and receivers, the proposed scheme improves the security and efficiency of image transmission by introducing another chaotic map with better chaotic performance and discrete logarithms. In order to verify and demonstrate the excellent performance of the proposed scheme, extensive simulation tests and theoretical analyses are carried out. These tests and analyses show that, compared with some recent schemes, the proposed scheme has higher feasibility, security, and practicability. In the future, we will extend the proposed scheme to video transmission.

Data Availability

The figure data and table data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Wei Feng conceptualized the study, was responsible for methodology and software, performed formal analysis and investigation, prepared the original draft, and was involved in funding acquisition; Jing Zhang was involved in conceptualization, funding acquisition, and supervision, validated the data, and reviewed and edited the manuscript; Zhentao Qin was responsible for methodology and software, validated the data, and reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was supported by the Science and Technology Development Center Project of Chinese Ministry of Education (no. 2018A0105), the Natural Science Key Project of Education Bureau of Sichuan Province (no. 18ZA0288), the Guiding Science and Technology Plan Project of Panzhihua City (nos. 2019ZD-G-18 and 2020ZD-S-40), and the Doctoral Research Startup Foundation of Panzhihua University (no. 2020DOC019).

References

[1] Y. Zhang, Y. Xiang, L. Y. Zhang, Y. Rong, and S. Guo, "Secure wireless communications based on compressive sensing: a survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1093–1111, 2019.

[2] Y. Zhang, Q. He, Y. Xiang et al., "Low-cost and confidentiality-preserving data acquisition for internet of multimedia things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3442–3451, 2018.

[3] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: a year in review," *Journal of Information Security and Applications*, vol. 48, Article ID 102361, 2019.

[4] Z. Niu, M. Zheng, Y. Zhang, and T. Wang, "A new asymmetrical encryption algorithm based on semitensor compressed sensing in WBANs," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 734–750, 2020.

[5] L. Wang, L. Li, J. Li, J. Li, B. B. Gupta, and X. Liu, "Compressive sensing of medical images with confidentially homomorphic aggregations," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1402–1409, 2019.

[6] L. Li, G. Wen, Z. Wang, and Y. Yang, "Efficient and secure image communication system based on compressed sensing for iot monitoring applications," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 82–95, 2020.

[7] L. Li, L. Liu, H. Peng, Y. Yang, and S. Cheng, "Flexible and secure data transmission system based on semitensor compressive sensing in wireless body area networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3212–3227, 2019.

[8] R. Zhao, Y. Zhang, X. Xiao, X. Ye, and R. Lan, "Tpe2: three-pixel exact thumbnail-preserving image encryption," *Signal Processing*, vol. 183, Article ID 108019, 2021.

[9] Y. Zhang, R. Zhao, X. Xiao, R. Lan, Z. Liu, and X. Zhang, "HF-TPE: high-fidelity thumbnail-preserving encryption," *IEEE Transactions on Circuits and Systems for Video Technology*, p. 1, 2021.

[10] A. S. Unde and P. P. Deepthi, "Design and analysis of compressive sensing-based lightweight encryption scheme for multimedia iot," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 167–171, 2020.

[11] Y. Zhang, Q. He, G. Chen, X. Zhang, and Y. Xiang, "A low-overhead, confidentiality-assured, and authenticated data

acquisition framework for iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7566–7578, 2020.

[12] B. Zhang, D. Xiao, and Y. Xiang, "Robust coding of encrypted images via 2D compressed sensing," *IEEE Transactions on Multimedia*, vol. 23, no. 99, pp. 2656–2671, 2021.

[13] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.

[14] W. Feng, Y. He, H. Li, and C. Li, "A plain-image-related chaotic image encryption algorithm based on DNA sequence operation and discrete logarithm," *IEEE Access*, vol. 7, pp. 181589–181609, 2019.

[15] H. Li, T. Li, W. Feng et al., "A novel image encryption scheme based on non-adjacent parallelable permutation and dynamic dna-level two-way diffusion," *Journal of Information Security and Applications*, vol. 61, Article ID 102844, 2021.

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

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

[18] Q. Yin and C. Wang, "A new chaotic image encryption scheme using breadth-first search and dynamic diffusion," *International Journal of Bifurcation and Chaos*, vol. 28, no. 4, Article ID 1850047, 2018.

[19] X. Wu, D. Wang, J. Kurths, and H. Kan, "A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system," *Information Sciences*, vol. 349-350, pp. 349-350, 2016.

[20] R. Zahmoul, R. Ejbali, and M. Zaied, "Image encryption based on new beta chaotic maps," *Optics and Lasers in Engineering*, vol. 96, pp. 39–49, 2017.

[21] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "An image encryption algorithm based on compound homogeneous hyper-chaotic system," *Nonlinear Dynamics*, vol. 89, no. 1, pp. 61–79, 2017.

[22] A.-V. Diaconu, "Circular inter-intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355-356, pp. 314–327, 2016.

[23] S. S. Moafimadani, Y. Chen, and C. Tang, "A new algorithm for medical color images encryption using chaotic systems," *Entropy*, vol. 21, no. 6, Article ID 577, 2019.

[24] P. Ramasamy, V. Ranganathan, S. Kadry, R. Damaševičius, and T. Blažauskas, "An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic-tent map," *Entropy*, vol. 21, no. 7, Article ID 656, 2019.

[25] S. Zhu, G. Wang, and C. Zhu, "A secure and fast image encryption scheme based on double chaotic s-boxes," *Entropy*, vol. 21, no. 8, Article ID 790, 2019.

[26] S. Zhou, P. He, and N. Kasabov, "A dynamic dna color image encryption method based on sha-512," *Entropy*, vol. 22, no. 10, Article ID 1091, 2020.

[27] L. Liu, D. Jiang, X. Wang, X. Rong, and R. Zhang, "2D logistic-adjusted-Chebyshev map for visual color image encryption," *Journal of Information Security and Applications*, vol. 60, Article ID 102854, 2021.

[28] D. Jiang, L. Liu, L. Zhu, X. Wang, X. Rong, and H. Chai, "Adaptive embedding: a novel meaningful image encryption scheme based on parallel compressive sensing and slant transform," *Signal Processing*, vol. 188, Article ID 108220, 2021.

[29] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[30] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[31] Y. Zhang, P. Wang, H. Huang, Y. Zhu, D. Xiao, and Y. Xiang, "Privacy-assured FOGCS: chaotic compressive sensing for secure industrial big image data processing in fog computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3401–3411, 2021.

[32] Y. Zhang, P. Wang, L. Fang, X. He, H. Han, and B. Chen, "Secure transmission of compressed sampling data using edge clouds," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6641–6651, 2020.

[33] L. Y. Zhang, K.-W. Wong, Y. Zhang, and J. Zhou, "Bi-level protected compressive sampling," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1720–1732, 2016.

[34] J. Wang, L. Y. Zhang, J. Chen, G. Hua, Y. Zhang, and Y. Xiang, "Compressed sensing based selective encryption with data hiding capability," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6560–6571, 2019.

[35] H. Peng, Y. Mi, L. Li, H. E. Stanley, and Y. Yang, "P-tensor product in compressed sensing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3492–3511, 2019.

[36] H. Gan, S. Xiao, T. Zhang, and F. Liu, "Bipolar measurement matrix using chaotic sequence," *Communications in Nonlinear Science and Numerical Simulation*, vol. 72, pp. 139–151, 2019.

[37] L. Li, Y. Fang, L. Liu, H. Peng, J. Kurths, and Y. Yang, "Overview of compressed sensing: sensing model, reconstruction algorithm, and its applications," *Applied Sciences*, vol. 10, no. 17, Article ID 5909, 2020.

[38] Y. Yicong Zhou, Z. Zhongyun Hua, C. Chi-Man Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 2001–2012, 2015.

[39] A. Eftekhari, M. Babaie-Zadeh, and H. Abrishami Moghaddam, "Two-dimensional random projection," *Signal Processing*, vol. 91, no. 7, pp. 1589–1603, 2011.

[40] G. Chen, D. Li, and J. Zhang, "Iterative gradient projection algorithm for two-dimensional compressive sensing sparse image reconstruction," *Signal Processing*, vol. 104, pp. 15–26, 2014.

[41] C. Li, D. Lin, B. Feng, J. Lü, and F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834–75842, 2018.

[42] W. Feng, Y. He, H. Li, and C. Li, "Cryptanalysis and improvement of the image encryption scheme based on 2D logistic-adjusted-sine map," *IEEE Access*, vol. 7, pp. 12584–12597, 2019.

[43] W. Feng and Y.-G. He, "Cryptanalysis and improvement of the hyper-chaotic image encryption scheme based on DNA encoding and scrambling," *IEEE Photonics Journal*, vol. 10, no. 6, pp. 1–15, Article ID 7909215, 2018.

[44] L. Y. Zhang, Y. Liu, F. Pareschi et al., "On the security of a class of diffusion mechanisms for image encryption," *IEEE transactions on cybernetics*, vol. 48, no. 4, pp. 1163–1175, 2018.

[45] M. Li, D. Lu, W. Wen, H. Ren, and Y. Zhang, "Cryptanalyzing a color image encryption scheme based on hybrid hyperchaotic system and cellular automata," *IEEE Access*, vol. 6, pp. 47102–47111, 2018.

[46] C. Li, D. Lin, and J. Lü, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE MultiMedia*, vol. 24, no. 3, pp. 64–71, 2017.

[47] C. Li, D. Lin, J. Lü, and F. Hao, "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE MultiMedia*, vol. 25, no. 4, pp. 46–56, 2018.

[48] M. Li, H. Fan, Y. Xiang, Y. Li, and Y. Zhang, "Cryptanalysis and improvement of a chaotic image encryption by first-order time-delay system," *IEEE MultiMedia*, vol. 25, no. 3, pp. 92–101, 2018.

[49] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 235–246, 2016.

[50] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

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

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

[53] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018.