

## RESEARCH ARTICLE

# A Novel Chaotic Image Encryption Algorithm Based on Coordinate Descent and SHA-256

XIYU SUN<sup>ID</sup> AND ZHONG CHEN<sup>ID</sup>College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China  
Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang 421002, China

Corresponding author: Zhong Chen (chenzhong@hynu.edu.cn)

This work was supported in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 19A066, and in part by the Science and Technology Innovation Program of Hunan Province under Grant 2016TP1020.

**ABSTRACT** In recent years, chaotic image encryption algorithms with key and plaintext association have been developed, which are essentially similar to a one-time pad at a time because each encryption requires the transmission of the key. However, some existing schemes cannot uniquely map the seed key to the initial value of the chaotic system, which leads to the reduction of the key space of the encryption system. In addition, some schemes use the same key to encrypt the same image, which does not conform to the one-time pad strategy. This paper solves these problems from two aspects. On the one hand, random pixels are inserted into a plain image and then a hash value is generated using SHA-256. Different seed keys can be obtained even if the same image is encrypted. On the other hand, the Sequential Expansion Algorithm (SEA) and Feedback Iterative Piece-Wise Linear Chaotic Mapping (FI-PWLCM) are proposed to realize the one-to-one correspondence between the seed key and the encrypted key stream. SEA can quickly generate seed key sensitive and random sequences. FI-PWLCM achieves one-to-one correspondence with the seed key through feedback iteration with more control parameters. The mapping not only has the rapidity of PWLCM, but also can produce more complex chaotic sequences. Besides, this paper proposes a Segmented Coordinate Descent (SCD) method for histogram statistical optimization of images to improve the ability of cryptosystems against statistical attacks. Experiments and security analysis show that the algorithm can resist chosen-plaintext (chosen-ciphertext) attacks, brute force attacks, statistical attacks and so on. Compared with most current algorithms, it achieves the best performance in the statistical properties of histogram and entropy.

**INDEX TERMS** Image encryption, coordinate descent, SHA-256, associate plain image, one-time pad.

## I. INTRODUCTION

In recent years, with the rapid development of the Internet and the rise of the Internet of Things, the transmission of information has become more frequent, and the issue of information security has also received more attention. One of the most effective ways to deal with the information security is to encrypt the relevant data [1], [2]. Generally speaking, images have a large amount of data and high redundancy, and their information value and accuracy are not as high as that of text (for example, an image in high-definition and not-so-high-definition expresses almost the same amount of information).

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

Therefore, ordinary image encryption generally requires less security than text encryption. The typical data encryption standard AES has high security, but this does not mean that it is universal for encryption in any situation. A good suggestion is to design a dedicated encryption algorithm according to the characteristics and application scenarios of the data. For example, it is reasonable to use lightweight encryption algorithms in restricted devices where security requirements are not too high. Likewise, encryption methods applicable to images are worth exploring.

Chaos-based image encryption algorithms are considered to be very promising due to the sensitivity of chaotic systems to initial values, ergodicity and pseudo-randomness [3]. Researchers have studied chaos-based image encryption

schemes from different perspectives [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. However, some cryptanalysts have demonstrated that there are obvious security problems with some of the proposed schemes [15], [16], [17], [18]. Most of the cracked algorithms are not resistant to plaintext-ciphertext pair attacks. Based on this, image encryption of key-associated plaintext is proposed, which is actually an image encryption scheme similar to the one-time pad strategy. The ideal one-time pad scheme is complete confidentiality [19]. In fact, the random key stream of arbitrary length is usually generated by the seed key through a pseudo-random generator, so this one-time pad-like encryption scheme cannot be completely kept secret. However, the security of this scheme is still very high, and we will call it a one-time pad scheme for the time being. According to Katz et al. [19], the security of one-time pad can be summarized into the following two principles:

- 1) Make sure that the seed key is not repeated every time it is encrypted. This is to prevent brute force attacks and plaintext-ciphertext pair attacks.
- 2) Make sure that the keystream generated by the seed key has sufficient randomness. The randomness of the key stream can ensure the randomness of the output ciphertext to prevent statistical attacks.

Some researchers use the method of correlating plaintext to simulate the one-time pad scheme [1], [9], [10], [20], [21], [22], [23], [24], [25]. They map the seed key to the initial value of the chaotic system, and then generate unpredictable pseudorandom sequences for encryption by iterating the chaotic system. However, there are some problems with this. On the one hand, the process of mapping the seed key to the initial value of the chaotic system is not one-to-one correspondence. For instance, in formula (4) in [10], a large number of equivalent keys can be generated just by exchanging the positions of the hash values, so multiple hash values are mapped to the same initial value of the chaotic system, which greatly reduces the key space. In addition, for example [1], [10], [20] encrypts the same image with the same key, which does not conform to the idea of one-time pad (This doesn't mean they are not secure, but since the key for each encryption needs to be secretly transmitted to the decrypting party, why not take a closer approach to the one-time pad?). On the other hand, although chaotic systems are pseudo-random, unpredictable. However, in finite precision devices, some literatures show that low-dimensional chaotic systems are prone to dynamic degradation [13], [26]. This makes the generated keystream predictable, which in turn leads to poor cryptographic statistical properties of the ciphertext. Some researchers solve this problem by using higher-dimensional, more complex chaotic systems [1], [10], however, this greatly increases the amount of computation and reduces the speed of encryption.

Image encryption algorithms based on meta-heuristics have been proposed to further improve the anti-statistical analysis capability of image encryption. For example, based on genetic algorithm [27], ant colony algorithm [28], simulated annealing algorithm [29], imperialist competition

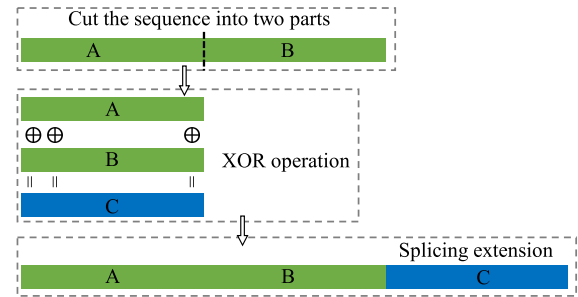


FIGURE 1. Schematic diagram of sequence extension once.

algorithm [30], dynamic harmony search algorithm [31] are used for image encryption. Compared with conventional image encryption algorithms, they improve the cryptographic image entropy, but they do not achieve a sufficiently stable and significant improvement. Kaur and Kumar [32] also pointed out that the existing meta-heuristic image encryption technology has problems such as slow calculation speed and difficult parameter adjustment.

Based on the above analysis, we propose a chaotic image encryption algorithm based on coordinate descent and SHA-256. This solution can effectively solve the above problems. First, random pixels are inserted before SHA-256 acts on the normal image to ensure that each encryption uses a different seed key. Second, we propose SEA, which is a simple and fast pseudo-random sequence generator. Third, we improve PWLCM to FI-PWLCM, which has more control parameters than PWLCM and can generate more complex chaotic sequences. Finally, we propose the SCD algorithm based on the idea of coordinate descent method to enhance the resistance of cryptosystems to statistical attacks. Both SEA and FI-PWLCM can generate random sequences that correspond one-to-one to 32-byte hashes, making the algorithm conform to the cryptographic rule of one-time pad. Compared with existing meta-heuristic image encryption algorithms, the proposed SCD algorithm can significantly optimize the histogram statistical properties of cryptographic images.

This paper will be arranged as follows. Section II describes the proposed image encryption scheme. Section III analyzes the experimental results and safety. Section IV concludes the research of this paper.

## II. THE PROPOSED IMAGE ENCRYPTION SCHEME

### A. SEQUENCE EXTENSION ALGORITHM (SEA)

The idea of the SEA is to turn a short sequence into a long sequence of a specified length. One feature of this long sequence is pseudo-randomness, which is beneficial to image encryption, and another feature is the correlation with the short sequence. After the value of the short sequence changes, the value of the long sequence will also change. The algorithm generates a long sequence that is required to associate the plain image, so the short sequence uses the 32-byte hash value generated by SHA-256. The short sequences are first correlated and then extended round by round. The extension process of each round is shown in Figure 1, and finally a long

**Algorithm 1** SEA

**Input:** The 32-byte hash value sequence  $\mathbf{k}$  generated by SHA-256, and the total number  $L$  of pixels of the plain image.

**Output:** The output sequence  $\mathbf{K}$ .

```

1:  $l \leftarrow \text{length}(\mathbf{k})$  // Get the length of  $\mathbf{k}$ .
2:  $c \leftarrow \text{mod}(k_1 \oplus k_2 + k_3 \oplus k_4 + \dots + k_{31} \oplus k_{32}, 127) + \text{mod}(k_2 \oplus k_3 + k_4 \oplus k_5 + \dots + k_{32} \oplus k_1, 128)$ ;
3:  $k_i \leftarrow \text{mod}(k_i + c, 256)$ ,  $i = 1, 2, 3, \dots, l$ ;
4:  $\mathbf{K} \leftarrow \mathbf{k}$ ;
5: while  $l < L$  do
6:    $l \leftarrow \text{floor}(l/2)$ ;  $\mathbf{A} \leftarrow \mathbf{K}(1 : l)$ ;  $\mathbf{B} \leftarrow \mathbf{K}(l + 1 : 2l)$ 
7:    $\mathbf{C}(i) \leftarrow \mathbf{A}(i) \oplus \mathbf{B}(i)$ ,  $i = 1, 2, 3, \dots, l$ 
8:    $\mathbf{K} \leftarrow [\mathbf{K}, \mathbf{C}]$ ;  $l \leftarrow \text{length}(\mathbf{K})$ ;
9: end while
10: return  $\mathbf{K} \leftarrow \mathbf{K}(1 : L)$ ;
    
```

**TABLE 1.** NIST SP 800-22 randomness test for SEA.

Test	P-value	Proportion
Frequency	0.2133	99/100
Block frequency	0.4190	98/100
Cumulative Sums	0.4634	98/100
Runs	0.7981	100/100
Longest run of ones	0.6993	100/100
Rank	0.8514	100/100
FFT	0.8165	100/100
Non-overlapping template	0.4947	99/100
Overlapping template	0.8514	98/100
Universal statistical	0.6371	99/100
Approximate entropy	0.6787	99/100
Random excursions	0.2024	59/59
Random excursions variant	0.3230	58/59
Serial	0.8276	98/100
Linear complexity	0.8677	98/100

sequence of the specified length is obtained. Algorithm 1 is its concrete realization.

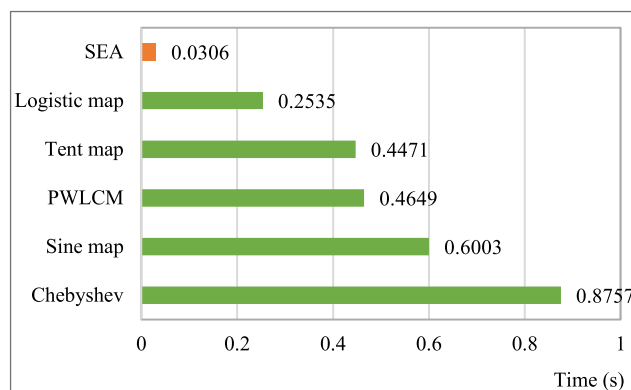
SEA is essentially a pseudo-random integer sequence generator whose purpose is to replace pixels of plain images such that the replacement data has appreciable randomness. We performed NIST randomness testing, velocity analysis and sensitivity analysis on the seed key for SEA to verify its validity.

1) NIST RANDOMNESS TEST

NIST SP 800-22 is a tool published by NIST (The National Institute of Standards and Technology) for randomness testing of random number generators and pseudo-random number generators. The tool has 15 tests and uses p-values and pass rates to assess whether the test passes. We took the significance level of 0.01 and generated 100 sets of byte streams of length  $10^6/8$  using SEA for different 32-byte seed keys  $k$  and then tested them using NIST SP 800-22. The results are shown in Table 1. It can be seen that all the P-values are greater than 0.01 and their pass rates are qualified (Note: the threshold of passable test is 96 for a total of 100 and 56 for a total of 59.). This means that the generated pseudo-random integer sequence is completely random.

2) SPEED ANALYSIS

In image encryption algorithms, chaotic systems can usually be used in pseudo-random number generators. While SEA is essentially a pseudo-random integer sequence generator, in order to verify its speed properties, we compared it with common one-dimensional chaotic systems. On one hand, we use Logistic map, Tent map, Sine map, Chebyshev, PWLCM to generate chaotic sequence  $s$  of length  $10^7$ , and then use  $\text{mod}(s \times 10^{10}, 256)$  to obtain the integer sequence. On the other hand, we generate integer sequences of length  $10^7$  using the SEA method. In MATLAB, the execution time of these two processes is calculated and the results are shown in Figure 2. Obviously, SEA has a faster speed.



**FIGURE 2.** Speed comparison of SEA with some 1D chaotic pseudo-random generators.

3) SENSITIVITY ANALYSIS TO SEED KEYS

The sensitivity of the key is a basic requirement for an encryption system. The procedure of sensitivity measurement is as follows: assume a 32-byte seed key  $\mathbf{k} = \{k_1, k_2, k_3, \dots, k_{32}\}$ , generate the original sequence  $\mathbf{K}_0$  using SEA, then change the value of  $k_i$  and obtain  $\mathbf{K}_i$  respectively. The difference in  $\mathbf{K}_i$  and  $\mathbf{K}_0$  can be measured by the NPCR (Number of Pixel Change Rate) and UACI (Unified Averaged Changed Intensity), which are calculated according to Eq. (16) and Eq. (17). The expected values of NPCR and UACI are 99.6 and 33.4, respectively. The process is repeated 10,000 times for different seed keys and the average values obtained are shown in Figure 3. It can be seen that the NPCR is about 95.2 and the UACI is about 30.7. This indicates that SEA is sensitive to seed keys.

**B. SEGMENTED COORDINATE DESCENT (SCD)**

The coordinate descent method is an optimization method that decomposes high-dimensional optimization problems into low-dimensional or even single-dimensional problems. There are many variants [33], [34], [35], [36], [37], [38]. The problem of coordinate descent can be described as

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} f(\mathbf{x}), \tag{1}$$

where  $f(\cdot)$  is differentiable,  $\mathbf{x} \in \mathbb{R}^m$ , and its lower bound is  $\mathbf{x}_{min}$ , and its upper bound is  $\mathbf{x}_{max}$ . Solve Eq. (1) by an iterative

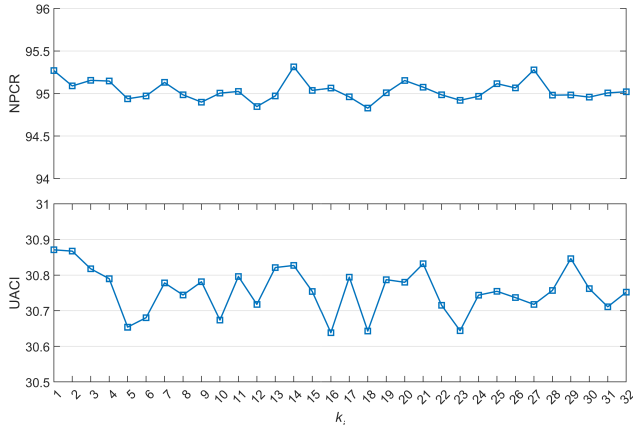


FIGURE 3. SEA sensitivity to the seed key.

method, and describe the  $t$ -th solution as:

$$\tilde{\mathbf{x}}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \dots, x_i^{(t)}, x_{i+1}^{(t)}, \dots, x_m^{(t)}]. \quad (2)$$

When  $f(\mathbf{x})$  is a convex continuously differentiable function, Eq. (1) can be transformed to solve the optimal value of each dimension, namely:

$$\tilde{x}_i^{(t)} = \arg \min_{x_i} f(x_1^{(t)}, x_2^{(t)}, \dots, x_i^{(t-1)}, \dots, x_m^{(t-1)}). \quad (3)$$

In Eqs. (1) - (3),  $m$  represents the dimensionality of the variable  $\mathbf{x}$ . The global optimum can be found by continuously and iteratively solving for the extreme values of each dimension. This conclusion has been proved in [39]. However, the above conclusion does not hold when  $f(\cdot)$  is not a continuously differentiable convex function. In fact nonconvex problems are often solved using metaheuristic search algorithms, because no algorithm can absolutely find the global optimal solution to nonconvex problems.

Inspired by coordinate descent, we propose a Segmented Coordinate Descent (SCD) search algorithm. The SCD algorithm is shown in Algorithm 2. Note that, unlike conventional metaheuristics, the proposed SCD algorithm focuses more on finding a better solution in a faster way rather than a global optimum, since finding the global optimum often requires huge computational power. A comparison with existing heuristic-based image encryption algorithms is made in Section III-F, and the results show that the proposed SCD algorithm obtains better results in less time. We will verify the properties of the SCD algorithm in 3 aspects: coordinate switching, subinterval line search and instance testing.

### 1) COORDINATE SWITCHING

We switch coordinates in a cyclic order to optimize the current solution step by step. The current solution  $\tilde{x}_i^{(t)}$  will get closer and closer to the optimal solution with the gradual optimization of each dimension. The SCD strategy is to use the current solution to partition the one-dimensional search interval into two segments and make the initial point in each segment search process closer and closer to the current solution  $\tilde{x}_i^{(t)}$ . By using the dynamic proximity coefficient  $\lambda$  to

### Algorithm 2 SCD

**Input:**  $\mathbf{x}_{min}$ ,  $\mathbf{x}_{max}$ , initial value  $\tilde{\mathbf{x}}$ , accuracy  $A_c$ , and the maximum number maxStep of iterations.

**Output:**  $\tilde{\mathbf{x}}$  and the optimized value  $\tilde{y}$ .

```

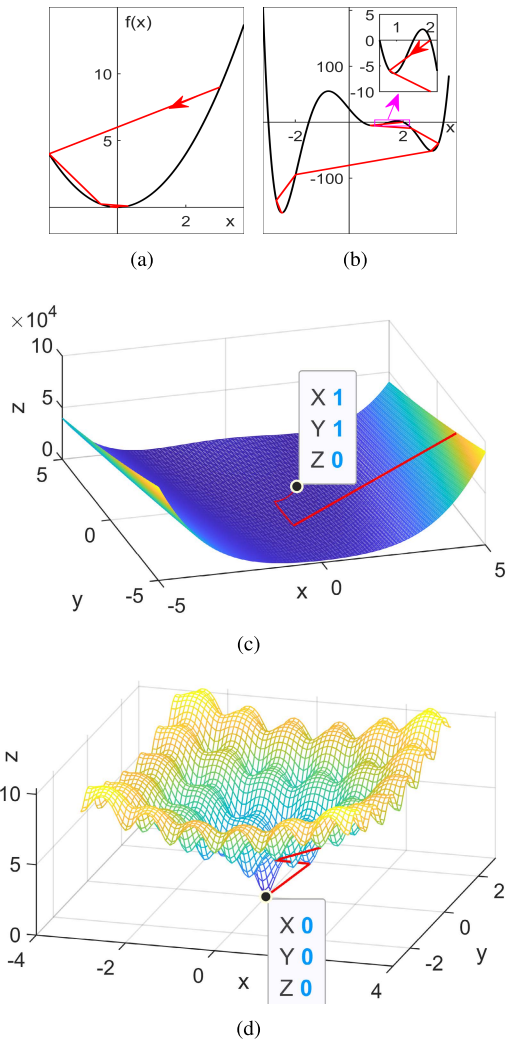
1: function  $[\tilde{\mathbf{x}}, \tilde{y}] = \text{SCD}(\mathbf{x}_{min}, \mathbf{x}_{max}, \tilde{\mathbf{x}}, A_c, \text{maxStep})$ 
2:    $m \leftarrow \text{length}(\mathbf{x}_0)$ ;  $m_1 \leftarrow \max(m, 10)$ ;
3:    $\lambda_1 \leftarrow 0.5$ ;  $\lambda_2 \leftarrow 0.9$ ;  $\lambda_3 \leftarrow 0.98$ ;
4:    $a \leftarrow \left(\frac{\lambda_3 - \lambda_2}{\lambda_3 - \lambda_1}\right)^{\frac{1}{m_1}}$ ;  $k \leftarrow \frac{\lambda_3 - \lambda_1}{a}$ ;
5:   for  $n_s \leftarrow 1$  : maxStep do
6:      $i \leftarrow \text{mod}(i - 1, m) + 1$ ;  $\lambda \leftarrow \lambda_3 - k \cdot a^{n_s}$ ;
7:      $\mathbf{x}_1 \leftarrow \mathbf{x}_{min}(i)$ ,  $\mathbf{x}_2 \leftarrow \tilde{\mathbf{x}}(i)$ 
8:      $\mathbf{x}_3 \leftarrow \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1)\lambda$ 
9:      $[\mathbf{x}_1, \sim] \leftarrow \text{subLineSea}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \tilde{\mathbf{x}}, A_c, i)$ ;
10:     $\mathbf{x}_2 \leftarrow \mathbf{x}_{max}(i)$ ;  $\mathbf{x}_3 \leftarrow \mathbf{x}_2 - (\mathbf{x}_2 - \mathbf{x}_1)\lambda$ ;
11:     $[\tilde{\mathbf{x}}, \tilde{y}] \leftarrow \text{subLineSea}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \tilde{\mathbf{x}}, A_c, i)$ ;
12:  end for
13: end function
14:
15: function  $[\mathbf{x}_1, \mathbf{y}_1] = \text{subLineSea}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \tilde{\mathbf{x}}, A_c, i)$ 
16:    $\mathbf{x}_1 \leftarrow \tilde{\mathbf{x}}$ ;  $\mathbf{x}_2 \leftarrow \tilde{\mathbf{x}}$ ;  $\mathbf{x}_3 \leftarrow \tilde{\mathbf{x}}$ ;  $\mathbf{x}_4 \leftarrow \tilde{\mathbf{x}}$ ;
17:    $\mathbf{x}_1(i) \leftarrow \mathbf{x}_1$ ;  $\mathbf{x}_2(i) \leftarrow \mathbf{x}_2$ ;  $\mathbf{x}_3(i) \leftarrow \mathbf{x}_3$ ;
18:    $\mathbf{y}_1 \leftarrow f(\mathbf{x}_1)$ ;  $\mathbf{y}_2 \leftarrow f(\mathbf{x}_2)$ ;  $\mathbf{y}_3 \leftarrow f(\mathbf{x}_3)$ ;
19:   while  $\mathbf{x}_2 - \mathbf{x}_1 > A_c$  do
20:     if  $\mathbf{y}_3 \leq \mathbf{y}_1$  and  $\mathbf{y}_3 \leq \mathbf{y}_2$  then
21:        $\mathbf{x}_4 \leftarrow \mathbf{x}_3 + 0.1A_c$ ;  $\mathbf{x}_4(i) \leftarrow \mathbf{x}_4$ ;  $\mathbf{y}_4 \leftarrow f(\mathbf{x}_4)$ ;
22:       if  $\mathbf{y}_4 > \mathbf{y}_3$  then
23:          $\mathbf{x}_2 \leftarrow \mathbf{x}_3$ ;  $\mathbf{y}_2 \leftarrow \mathbf{y}_3$ ;
24:       else
25:          $\mathbf{x}_1 \leftarrow \mathbf{x}_3$ ;  $\mathbf{y}_1 \leftarrow \mathbf{y}_3$ ;
26:       end if
27:     else
28:       if  $\mathbf{y}_1 < \mathbf{y}_2$  then
29:          $\mathbf{x}_2 \leftarrow \mathbf{x}_3$ ;  $\mathbf{y}_2 \leftarrow \mathbf{y}_3$ ;
30:       else
31:          $\mathbf{x}_1 \leftarrow \mathbf{x}_3$ ;  $\mathbf{y}_1 \leftarrow \mathbf{y}_3$ ;
32:       end if
33:     end if
34:      $\mathbf{x}_3 \leftarrow \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1)(0.2 + 0.6 \times \text{rand}())$ ;
35:      $\mathbf{x}_3(i) \leftarrow \mathbf{x}_3$ ;  $\mathbf{y}_3 \leftarrow f(\mathbf{x}_3)$ ;
36:   end while
37:   if  $\mathbf{y}_1 > \mathbf{y}_2$  then  $\mathbf{x}_1 \leftarrow \mathbf{x}_2$ ;  $\mathbf{y}_1 \leftarrow \mathbf{y}_2$ ; end
38: end function

```

indicate the closeness to the current value  $\tilde{x}_i^{(t)}$ . The closer the value is to 1, the closer the split point is to the  $\tilde{x}_i^{(t)}$ , and  $\lambda$  is defined as:

$$\begin{cases} \lambda = \lambda_3 - k \cdot a^i \\ a = \left(\frac{\lambda_3 - \lambda_2}{\lambda_3 - \lambda_1}\right)^{\frac{1}{m}} \\ k = \frac{\lambda_3 - \lambda_1}{a} \end{cases} \quad (4)$$

where  $\lambda_3$  is the upper bound of  $\lambda$ ,  $i$  represents the current search dimension,  $k$  and  $a$  are undetermined parameters,  $\lambda_1$  is

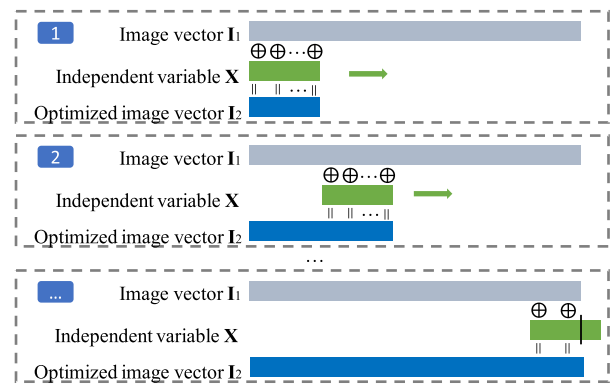


**FIGURE 4. Several examples of applying the SCD algorithm.**  
 (a)  $y = x^2, x \in [-2, 3.7]$ .  
 (b)  $y = (x + 3)(x + 1.5)(x - 0.5)(x - 1.5)(x - 2)(x - 3.5), x \in [-3.2, 3.7]$ .  
 (c)  $z = 100(y - x^2)^2 + (x - 1)^2, x \in [-5, 5], \text{ and } y \in [-5, 5]$ .  
 (d)  $z = -20e^{(-0.2\sqrt{x^2+y^2})} - e^{\frac{\cos(2\pi x)+\cos(2\pi y)}{2}} + e + 20, x \in [-3, 3], \text{ and } y \in [-3, 3]$ .

the initial value of  $\lambda$ , and  $\lambda_2$  is the value of  $\lambda$  when  $i = m$ . The intent of setting this coefficient is to reduce the amount of calculation.

2) SUBINTERVAL LINE SEARCH

In Algorithm 2 the function subLineSea(.) solves for the smaller value in the interval by continually narrowing the search interval. For a differentiable convex function, the function must be searchable to a minimum (the proof of this conclusion is obvious, since the search process ensures that the minima always lie within the search interval.). Therefore, the SCD algorithm can solve Eq. (1). subLineSea(.)’s search process is randomized, so that for non-convex functions there is a certain probability that the minimum will be searched. In addition, the design of subLineSea(.) does not use gradient information so it can be used for optimization problems with discontinuous functions.



**FIGURE 5. Scheme of changing image pixels.**

3) SEVERAL EXAMPLE TESTS

To visualize the results, as shown in Figure 4, we enumerate 2 unary functions and 2 binary functions to observe the search process of SCD. It is clear that for convex problems SCD converges to a minimum, while for nonconvex problems, SCD has a certain probability of jumping out of the local optimal solution.

C. OPTIMIZING THE STATISTICAL PROPERTIES OF CIPHER IMAGES USING SCD

The chi-square value of the cipher image is taken as the optimization target, since the chi-square value of the image is easy to calculate, and the chi-square value of the image is related to the information entropy of the image. Generally speaking, the smaller the chi-square value of the image, the more even the pixel distribution of the image, the greater the information entropy of the image. Considering that the image encryption needs to be reversible, by using the XOR operation to construct the scheme of changing the image pixel, as shown in Figure 5. The image vector  $I_1$  is XORed with a variable  $X$ , and the  $I_1$  can be converted into the optimized image vector  $I_2$  by the cyclic XOR operation. Use the SCD algorithm to find an  $X$  such that the chi-square value of  $I_2$  is as small as possible to optimize the statistical properties of the cipher image.

Let:

$$I_1 = [I_1(1), I_1(2), \dots, I_1(MN)], \tag{5}$$

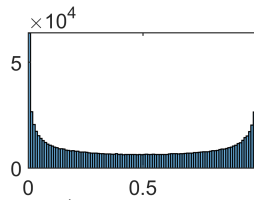
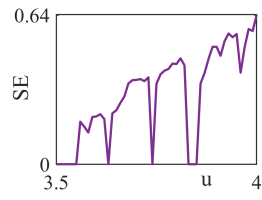
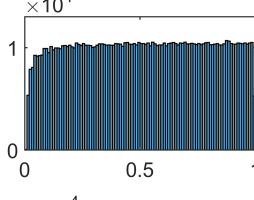
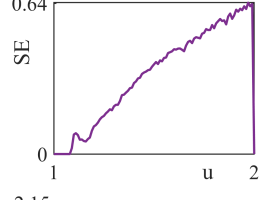
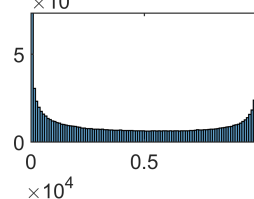
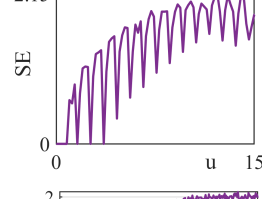
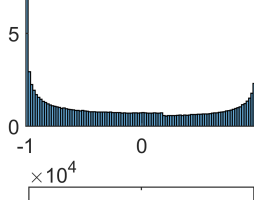
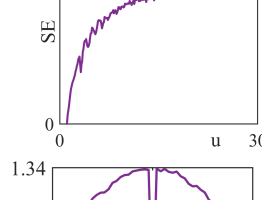
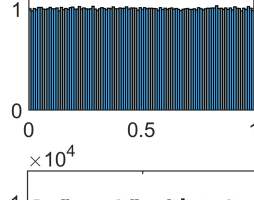
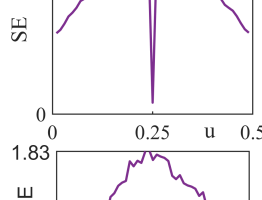
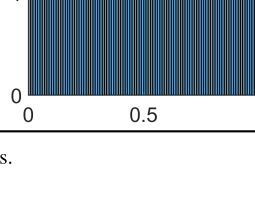

$$X = [X(1), X(2), \dots, X(m)]. \tag{6}$$

According to Figure 5,  $I_2$  can be expressed as

$$\begin{aligned} I_2 &= \text{cycXOR}(I_1, X) \\ &= [I_1(1) \oplus X(1), \dots, I_1(m) \oplus X(m), \\ &\quad I_1(m + 1) \oplus X(1), \dots, I_1(2m) \oplus X(m), \\ &\quad I_1(2m + 1) \oplus X(1), \dots, I_1(3m) \oplus X(m), \\ &\quad \dots, I_1(MN) \oplus X(\text{mod}(MN - 1, m) + 1)], \end{aligned} \tag{7}$$

where  $m$  represents the dimension of  $X$ ,  $M$  represents the height of the image, and  $N$  represents the image’s width. The objective function can be expressed as the chi-square

**TABLE 2.** Several comparisons between FI-PWLCM and typical one-dimensional chaotic systems.

Chaos map	Settings	Time (s)	Histogram	Sample entropy
Logistic map $x_{n+1} = ux_n(1 - x_n)$	$x_1 = 0.26$ $u = 4$	0.0356		
Tent map $x_{n+1} = \begin{cases} ux_n & \text{if } x_n < 0.5 \\ u(1 - x_n) & \text{otherwise} \end{cases}$	$x_1 = 0.26$ $u = 1.99$	0.0846		
Sine map $x_{n+1} = u \cdot \sin(\pi x_n)$	$x_1 = 0.26$ $u = 1$	0.2610		
Chebyshev map $x_{n+1} = \cos(u \cdot \arccos(x_n))$	$x_1 = 0.26$ $u = 3.56$	0.5276		
PWLCM Eq. (9)	$x_1 = 0.26$ $u = 0.27$	0.1152		
FI-PWLCM Eq. (10)	$\beta_1 = 0.23$ $\beta_2 = 0.56$ $\beta_3 = 0.81$ $\beta_4 = 0.64$ $\beta_5 = 0.14$ $\beta_6 = 0.46$ $u = 0.24$	0.1341		

Note: Time and histogram are the test results of iterating the chaotic map  $10^7$  times.

value of  $I_2$ , witch is

$$\chi^2 = f(\mathbf{X}) = \sum_{i=0}^{255} (f_r(\text{cycXOR}(I_1, \mathbf{X}), i) - \bar{f})^2 / \bar{f}, \quad (8)$$

where  $\bar{f} = (M \times N)/256$ ,  $\chi^2$  represents the chi-square value of  $I_2$ , and  $f_r(\text{Img}, i)$  denotes the number of pixels at pixel level  $i$  in the statistical image  $\text{Img}$ . The SCD algorithm needs to be modified to suit the case of integer computations. Specifically, use  $\text{floor}(\cdot)$  to round up lines 9, 11, 16, 18 and 45. In line 31,  $x_4 \leftarrow x_4 + 1$ . The input parameters are set as  $\mathbf{x}_{min} = [0, 0, \dots, 0]_{1 \times m}$ ,  $\mathbf{x}_{max} = [255, 255, \dots, 255]_{1 \times m}$ ,  $A_c = 1$ ,  $\text{maxStep} = 3m$ . Then the SCD algorithm is executed to obtain  $\tilde{x}$  and  $\tilde{y}$ . According to Eq. (7),  $I_2 = \text{cycXOR}(I_1, \tilde{x})$ .

#### D. FEEDBACK ITERATIVE PIECE-WISE LINEAR CHAOTIC MAP (FI-PWLCM)

Piece-Wise Linear Chaotic Map (PWLCM) is defined as

$$x_{n+1} = f_p(x_n) = \begin{cases} x_n & 0 < x_n < p \\ \frac{x_n - p}{0.5 - p} & p \leq x_n < 0.5 \\ f_p(1 - x_n) & 0.5 < x_n < 1, \end{cases} \quad (9)$$

where  $u \in (0, 0.5)$ . We conduct several analyses of common one-dimensional chaotic systems as shown in Table 2. As analyzed in Table 2, in a one-dimensional chaotic system, PWLCM has relatively fast speed and high sample entropy

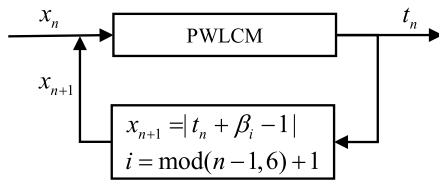


FIGURE 6. Structure of FI-PWLCM.

(The more complex the sequence, the higher the sample entropy.). However, the system is only determined by two initial values, and cannot achieve a one-to-one mapping with a 32-byte hash value within a precision of  $10^{16}$ . In addition, the system exhibits severe dynamic degradation when  $u = 0.25$ . The system cannot complete the iteration when  $x = 0.5$ .

Based on the above analysis, we improve PWLCM following the structure of Figure 6 to avoid some known weaknesses of PWLCM and ensure that the seed key  $k$  has a one-to-one correspondence with the chaotic sequences generated by the improved system. The PWLCM with this feedback iterative structure is called FI-PWLCM. Its recursive formula is:

$$t_n = \begin{cases} \frac{x_n}{u} & \text{if } x_n < u \\ \frac{x_n - u}{0.5 - u} & \text{if } x_n < 0.5 \\ \frac{1 - x_n}{1 - x_n - u} & \text{if } 1 - x \leq u \\ \frac{u}{1 - x_n - u} & \text{otherwise} \end{cases} \quad (10)$$

$$x_{n+1} = |t_n + \beta_i - 1|,$$

where  $n \geq 1$ ,  $x_1 = \beta_1$ ,  $i = \text{mod}(n-1, 6) + 1$ ,  $t_n$  is the output of the system, and  $\beta_i$  is determined by a 32-byte hash value  $k$ , which is calculated as:

$$\begin{cases} \beta_1 = \frac{\sum_{i=1}^6 k_i 2^{8(i-1)}}{2^{48}} \\ \beta_2 = \frac{\sum_{i=7}^{12} k_i 2^{8(i-7)}}{2^{48}} \\ \beta_3 = \frac{\sum_{i=13}^{18} k_i 2^{8(i-13)}}{2^{48}} \\ \beta_4 = \frac{\sum_{i=19}^{24} k_i 2^{8(i-19)}}{2^{48}} \\ \beta_5 = \frac{\sum_{i=25}^{30} k_i 2^{8(i-25)}}{2^{48}} \\ \beta_6 = \frac{\sum_{i=31}^{32} k_i 2^{8(i-31)}}{2^{48}}. \end{cases} \quad (11)$$

$u$  is the control parameter of the system,  $u \in (0, 0.5)$ , to make the generated sequence have high enough complexity, it is defined as:

$$u = 0.23 + \text{mod}(\sum_{i=1}^6 \beta_i, 1)/100. \quad (12)$$

Obviously, the hash value  $k$  and  $\beta$  constitute a one-to-one correspondence, and  $\beta_i$  will continue to act on FI-PWLCM, so that the generated sequence and the hash value are in one-to-one correspondence. Furthermore, we perform velocity,

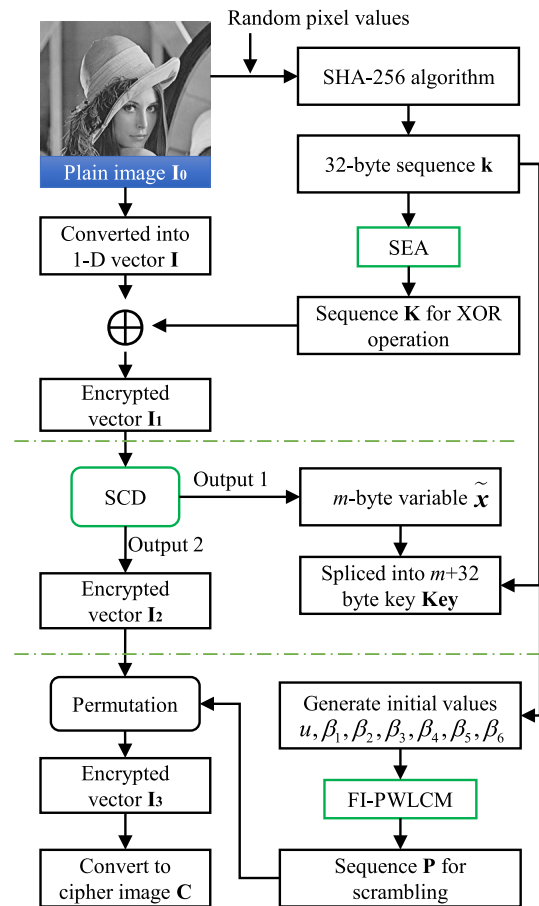


FIGURE 7. Block diagram of the proposed image encryption scheme.

sample entropy and histogram analysis on FI-PWLCM with a typical one-dimensional chaotic system. The experimental results are shown in Table 2. Although Sine map and chebyshev map have higher sample entropy, they are too slow. And FI-PWLCM not only has similar speed performance to PWLCM, but also has higher sample entropy. Therefore, it can replace PWLCM for image encryption. FI-PWLCM is iterated  $MN + 200$  times, where  $M$  and  $N$  represent the height and width of the image, respectively. The chaotic sequence is obtained by discarding the first 200 times, and then the index subscript sequence  $P$  is obtained by ascending the chaotic sequence.

**E. ENCRYPTION AND DECRYPTION ALGORITHMS**

Figure 7 is a block diagram of encryption. The key consists of two parts. One is the 32-byte hash value of the plain image processed by SHA-256. The second is the m-byte extended key of SCD optimization process, the length of which can be set by the user. The encryption process can be divided into three stages. First using SEA substitution, second using SCD optimization, and finally using FI-PWLCM permutation. The substitution process can make the plaintext information approximate to pseudo-random information, so that when some specific information (such as all 0 information) is

input in the plaintext, some obvious features will not appear after the substitution. The SCD optimization process is equivalent to extracting some common features of the information to be encrypted with the key, so that the output ciphertext contains less features about the plaintext. The FI-PWLCM permutation can make the position of the plain image different from the position of the cipher image, making it difficult for the attacker to find the corresponding relationship. The complete encryption algorithm is described in Algorithm 3. The decryption algorithm is described in Algorithm 4.

### III. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

There are some necessary clarifications that need to be done before experimental analysis. First, the development environment is a 64-bit Windows 10 system, 8.00GB RAM, Intel(R) Core (TM) i7-10510U CPU@1.8 GHz, 2304 Mhz. Moreover, use MATLAB to write programs and experiments. Second, since the key is dynamic, when the value of  $m$  is not specified, the default  $m = 80$ . Finally, all experimental data are the results obtained through multiple experiments.

#### A. ENCRYPTION AND DECRYPTION RESULT

The encryption and decryption results for different types of images are shown in Figure 8. This algorithm generates similar cipher images for different plain images. The naked eye cannot tell the difference between these cipher images and cannot detect their features related to the plain images.

---

#### Algorithm 3 Encryption Algorithm

---

**Input:** Plain image  $I_0$ , the variable part of the key has length  $m$ .

**Output:** Cipher image  $C$ , and the key Key used for decryption.

- 1: Convert  $I_0$  into one-dimensional vector  $I$ , and find its length  $l$ .
  - 2: Insert 10 random pixel values at the end of  $I_0$  to get  $\mathcal{N}_0$ .
  - 3: The 32-byte sequence  $k$  is obtained by applying the SHA-256 to  $\mathcal{N}_0$ .
  - 4: The XOR sequence  $K$  is obtained by Algorithm 1 of using  $k$  and  $l$  as input.
  - 5:  $I_1(i) \leftarrow I(i) \oplus K(i)$ , where  $i = 1, 2, 3, \dots, l$ .
  - 6: According to Section II-C,  $\tilde{x}$  and  $I_2$  are obtained.
  - 7:  $\text{Key} \leftarrow [k; \tilde{x}]$ . // Splice  $k$  with  $\tilde{x}$ .
  - 8: The sequence  $P$  is obtained by the method in Section II-D.
  - 9:  $I_3(i) \leftarrow I_2(P(i))$ , where  $i = 1, 2, 3, \dots, l$ .
  - 10:  $C$  is the final encrypted image, which is obtained by converting the vector  $I_3$  into an image matrix.
- 

#### B. THE RELATIONSHIP BETWEEN CHI-SQUARE VALUE, $m$ VALUE AND ITERATION TIMES

Figure 9 directly reflects the relationship between chi-square value,  $m$  value and iteration times of cipher image. On the one hand, the  $m$  value determines the dimension of the variables

---

#### Algorithm 4 Decryption Algorithm

---

**Input:** Cipher image matrix  $C$ , and the key Key used for decryption.

**Output:** The decrypted image  $I_0$ .

- 1: Convert  $C$  into one-dimensional vector  $I_3$ , and find its length  $l$ .
  - 2:  $k \leftarrow \text{Key}(1 : 32)$ .
  - 3: The integer sequence  $K$  is obtained by Algorithm 1 of using  $k$  and  $l$  as input.
  - 4: The sequence  $P$  is obtained by the method in Section II-D.
  - 5:  $I_2(P(i)) \leftarrow I_3(i)$ , where  $i = 1, 2, 3, \dots, l$ .
  - 6:  $X \leftarrow \text{key}(33 : \text{end})$  and  $m \leftarrow \text{length}(X)$ .
  - 7:  $I_1 \leftarrow \text{cycXOR}(I_2, X)$ . // See Eq. (7).
  - 8:  $I(i) \leftarrow I_1(i) \oplus K(i)$ , where  $i = 1, 2, 3, \dots, l$ .
  - 9:  $I_0$  is the final decrypted image, which is obtained by converting the vector  $I$  into an image matrix.
- 

in the SCD process. The larger the  $m$  value, the more variables will participate in the optimization, and the chi-square value of the cipher image will be reduced more easily. However, if  $m$  is too high, the storage space of decryption key will be increased, making key transmission difficult. On the other hand, the higher the iteration times of SCD algorithm, the lower the chi-square value. But, too high a number of iterations will waste a lot of time. Therefore, the value of  $m$  and the number of iterations can be set by the user to trade off security and efficiency. In addition, according to Figure 9, when the number of iterations is greater than  $3m$ , the chi-square value decreases slowly, so the maximum number of iterations is recommended to be  $3m$ . The curves of  $m = 80$  and  $m = 100$  are very close, so  $m = 80$  is recommended.

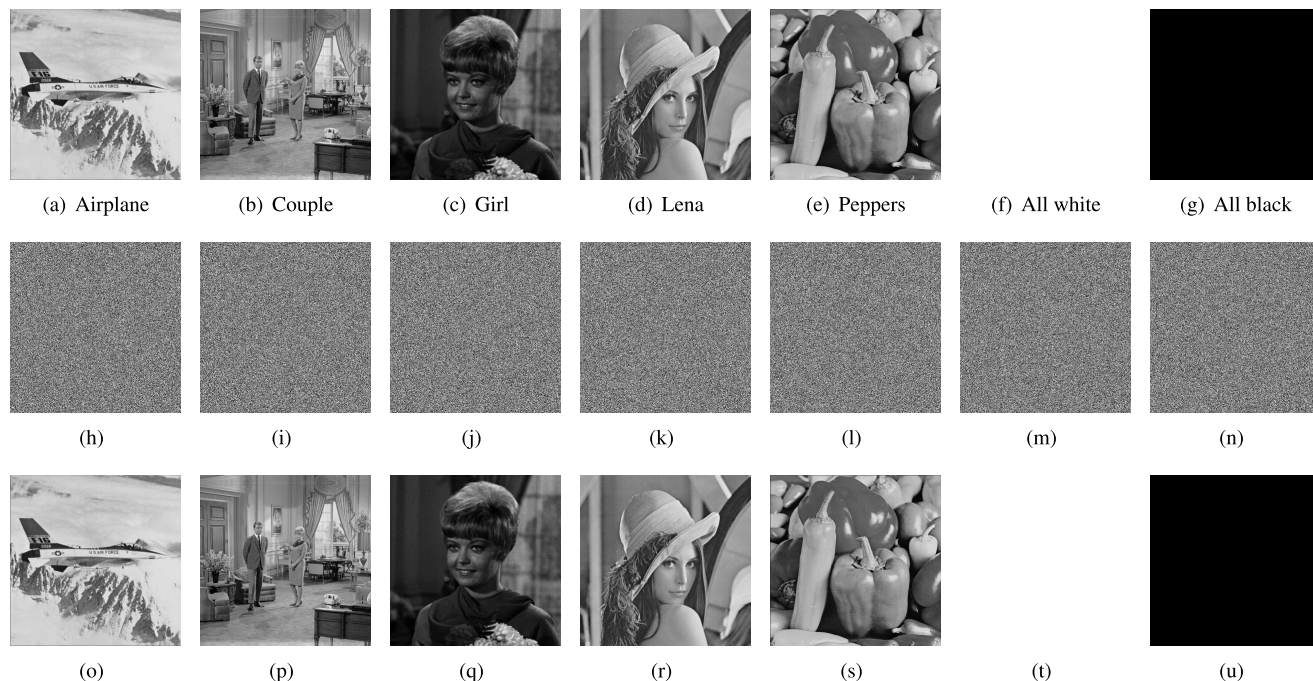
#### C. SECURITY ANALYSIS

According to Kerckhoffs' principle, an attacker knows every design detail of a cryptographic algorithm except the key. Based on this principle, attack methods can usually be divided into the following five situations:

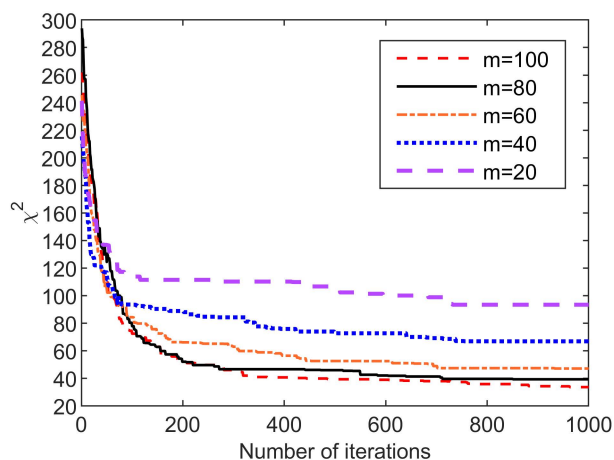
- 1) Ciphertext-only attack: The attacker knows the encryption algorithm and the ciphertext to be deciphered. This is the most difficult attack scenario.
- 2) Known-plaintext attack: The attacker knows the encryption algorithm, some plaintext ciphertext pair.
- 3) Chosen-plaintext attack: The attacker knows the encryption algorithm and can choose some special plaintext ciphertext pair.
- 4) Chosen-ciphertext attack: The attack knows the encryption algorithm and can choose some special ciphertext plaintext pair.
- 5) Selective text attack: a combination of selective plaintext attack and selective ciphertext attack.

A cryptographic algorithm can be considered highly secure if it is resistant to selective text attacks. Note that the proposed algorithm is a one-time pad encryption strategy. An attacker can easily obtain the key through a





**FIGURE 8.** Test images. The first line represents the plain images, the second line is the corresponding cipher images, and the third is the decrypted images.



**FIGURE 9.** The relationship between chi-square value,  $m$  value and iteration times.

chosen-plaintext (chosen-ciphertext) attack, but the key cannot be used to decipher other ciphertext information. Therefore, any attempt to exploit a plaintext ciphertext pair to guess the key will fail. Furthermore, differential and linear attacks would be difficult to implement because the encryption key is different each time. Brute force attack can only decipher the current cipher image, because the key obtained by it cannot be applied to the cracking of other ciphertext information. It is possible to use statistical attacks to establish the relationship between the ciphertext and the key, but this depends on the statistical properties of the cipher image representation. It is worth noting that although this algorithm uses a simple chaotic system, the combination with SCD,

SEA and SHA-256 makes the system complex enough, and SCD makes the statistical properties of the generated cryptographic images very random. Coupled with the one-time pad encryption strategy used, it is difficult for an attacker to find information related to the key from the cipher image. Therefore, this algorithm is resistant to selective text attacks. Based on the above analysis, the focus of this paper on the security analysis of this algorithm is statistical analysis.

### 1) BRUTE FORCE ATTACK ANALYSIS

A brute force attack is a method in which an attacker reaches and cracks a password by exhausting all the keys of an encryption algorithm. This attack method is a condition that an encryption system must satisfy at least. The complexity of the implementation of this method is directly related to the size of the key space, as long as the key space is large enough, this method is difficult to implement. Gonzalo et al. [40] states that the key space is at least greater than  $2^{100}$ . Due to the degradation of the chaotic system in the limited-precision computer equipment, that is, the computer has truncation error and rounding error in the numerical calculation, which will lead to the weakening of the initial value sensitivity of the chaotic system. Therefore, the initial value precision of the chaotic system should be properly selected in order to obtain an accurate key space. In Eq. (11) in Section II-D, it is determined that the initial value precision of FI-PWLCM is  $1/2^{48} > 3.5 \times 10^{-15}$ , which is within the calculation range of double precision, that is, the initial value generated by Eq. (11) is valid. The algorithm key proposed consists of two parts, the first 32 bytes are the SHA-256 sequence, and the last  $m$  bytes are the optimized parameters using SCD.

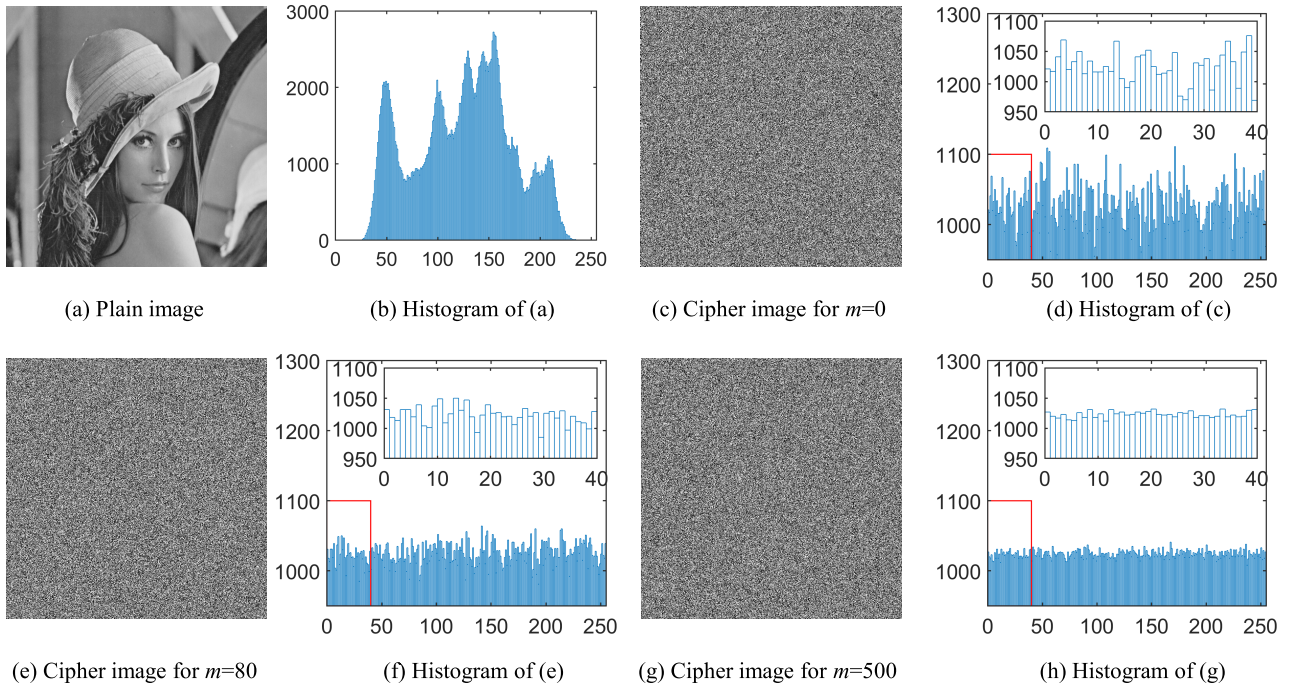


FIGURE 10. Encryption results and histograms for different  $m$  values.

Therefore, the key is  $32 + m$  bytes in total, that is, the key space is:  $2^{8 \times (32+m)} = 2^{256+8m} > 2^{100}$ . This is enough to resist brute force attacks.

## 2) STATISTICAL ATTACK ANALYSIS

Statistical attack refers to the use of certain statistical laws of plaintext and ciphertext to achieve cryptographic attacks. Images are a highly correlated and redundant medium, it has many statistical characteristics, and it is a common analysis method to use the inherent characteristics of image to perform statistical analysis on encryption system. Preishuber et al. [41] and Özkaynak [42] have proposed that even if an encryption scheme perfectly passes several statistical analysis indicators (such as NPCR, UACI, entropy, etc.), it cannot be regarded as a sufficient condition for the security of the cryptosystem. Even so, statistical analysis of a cryptographic algorithm is still necessary, because if a cryptographic algorithm cannot pass even this typical attack, then the cryptographic algorithm cannot be a good algorithm. Several typical statistical indicators are analyzed below.

### a: HISTOGRAM AND CHI-SQUARE VALUE

The histogram of an image can reflect important information in the original image. Attackers can use it to perform statistical analysis. Image encryption should change this feature. On the one hand, the distribution of pixel values is visually analyzed by plotting the histogram of the cipher image. To reduce the correlation between the histogram and the plain image, the histogram of the cipher image should be inclined towards the horizontal line [43]. The experimental results are shown in Figure 10. From the figure, it can be seen that the histogram of the cipher image obtained by the algorithm is

completely different from that of the plain image, and is very close to the horizontal line. On the other hand, by calculating the chi-square value of the histogram, which can more accurately reflect the horizontal characteristics of the histogram of the cipher image. The calculation of the chi-square value is shown in Eq. (8). Different images were experimented and compared with [33], [48], and [49]. Figure 11 is experimental result. Note: In the chi-square test, the significance level was 0.05, corresponding to a chi-square value of 293.2478. The results show that this algorithm easily passes the chi-square test. Compared to [33], [48], and [49], the chi-square value of the proposed algorithm drops by more than 70% on average. We are not surprised by this result, as it just proves the effectiveness of the SCD algorithm. SCD optimizes the replacement process so that the pixel distribution of the replacement result is more uniform, so that the histogram characteristics are significantly better than other encryption algorithms.

### b: CORRELATION BETWEEN ADJACENT PIXELS

The strong correlation of neighborhood pixels is one of the typical features of available images. Based on this, image encryption should destroy this feature [12]. The typical feature of an encrypted image is that the cipher image becomes very chaotic. Mathematically, it can be considered that the correlation between adjacent pixels of the image is highly low. Correlation is mathematically defined as:

$$\begin{cases} E(x) = \frac{\sum_{i=1}^N x_i}{N}, E(y) = \frac{\sum_{i=1}^N y_i}{N} \\ C_{xy} = \frac{\sum_{i=1}^N [x_i - E(x)][y_i - E(y)]}{\sqrt{\sum_{i=1}^N [x_i - E(x)]^2} \sqrt{\sum_{i=1}^N [y_i - E(y)]^2}} \end{cases} \quad (13)$$

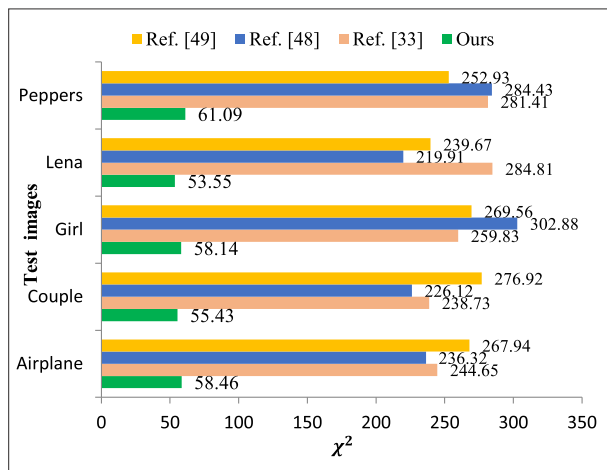


FIGURE 11. Comparison with Ref. [29], [44], and [45] in terms of chi-square values (images size: 512 × 512).

TABLE 3. Correlation coefficients of adjacent pixels for Lena (512 × 512).

Images	Horizontal	Vertical	Diagonal	Mean
Original Lena	0.971888	0.984961	0.963873	0.973574
Ref. [29]	-0.004610	0.001015	0.007802	0.004476
Ref. [44]	0.002357	<b>0.000267</b>	<b>-0.000268</b>	0.000964
Ref. [45]	0.001564	0.006659	0.000716	0.002980
Ours	<b>-0.000271</b>	-0.000965	0.000402	<b>0.000546</b>

where  $N$  represents the number of pairs of adjacent pixel points selected, and  $x$  and  $y$  represent the gray values of a pair of adjacent points. Experiments are conducted by randomly selecting 5000 pairs of pixels in the horizontal, vertical and diagonal directions. The results are shown in Figure 12 and Table 3. As can be seen from Table 3, the correlation of the proposed encryption algorithm in three directions is lower than  $10^{-3}$ . Compared with the algorithm of Hua et al. [44], the proposed algorithm is more stable and has lower average values in three directions. This result seems to have a probability factor, but it is actually an inevitable result. Assuming that an image with uniform histogram and an image with non-uniform histogram are arranged by the same shuffling method, the pixel values with high frequency of the image with non-uniform histogram are more likely to be assigned to adjacent positions, which leads to increased correlation. Therefore, the image with uniform histogram will have smaller correlation between adjacent pixels after shuffling. The proposed algorithm SCD process optimizes the histogram characteristics, and FI-PWLCM can produce highly complex chaotic sequences, which is very suitable for pixel shuffling. Therefore, the algorithm can obtain quite good correlation between adjacent pixels.

*c: INFORMATION ENTROPY*

Information entropy can reflect whether the distribution of pixels in the image is uniform [46]. To make the encrypted image more random, its information entropy should be larger. The information entropy of an image is defined as:

$$H(S) = - \sum_{i=0}^{N-1} P(s_i) \log_2 P(s_i), \quad (14)$$

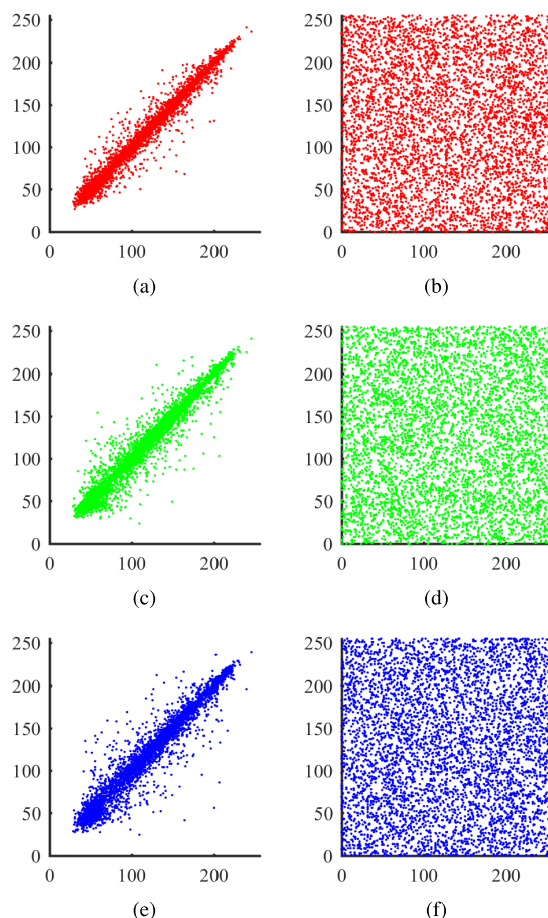
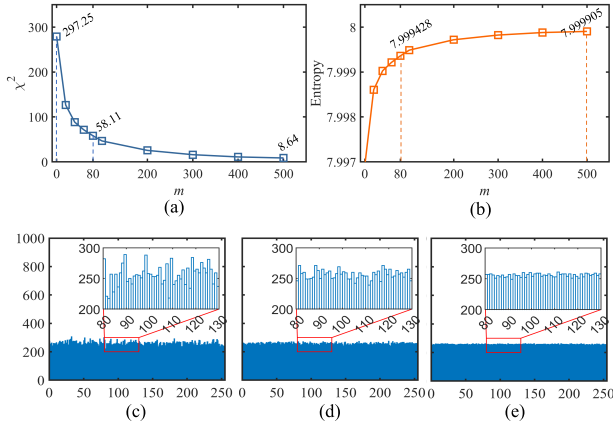


FIGURE 12. Correlation of Lena's neighboring pixels. (a), (c) and (e) are the correlations of Lena plain images in horizontal, vertical and diagonal directions, respectively. (b), (d) and (f) are the correlations of Lena cipher images in horizontal, vertical and diagonal directions, respectively.

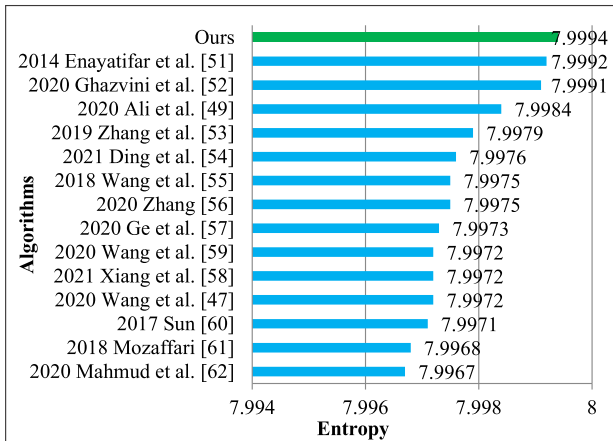
where  $S$  represents the gray value,  $N$  represents the gray level of the image, and  $P(s_i)$  represents the probability of the  $i$ -th gray level in the entire image. For an ideal gray image, the gray value of each level has the same probability, at this time,  $H(S) = 8$ . Figure 13 shows the changes in the chi-square value and information entropy of the cipher image for Lena ( $256 \times 256$ ) as the value of  $m$  changes. It can be seen from Figure 13 (a) that the chi-square value of the cipher image decreases as  $m$  increases. This is reflected in the changes in the histograms in Figure 13 (c), (d) and (e), where it can be seen that the histograms become more and more horizontal. This is also reflected in the information entropy as shown in Figure 13 (b), the entropy value gradually increases with the increase of  $m$ , and finally approaches the limit value of 8. One can see from Table 4 that proposed method get higher entropy value. Figure 14 is a comparison of the information entropy of various algorithms [20], [25], [43], [45], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56]. The experimental results show that, compared with most current image encryption algorithms, the algorithm has the best performance in terms of information entropy.

**TABLE 4. Entropy of cipher image information of different images (images size: 512 × 512).**

Test images	Ref. [29]	Ref. [44]	Ref. [45]	Ours
Airplane	7.999357	7.999348	7.999263	<b>7.999852</b>
Lena	7.999395	7.999395	7.999340	<b>7.999861</b>
Peppers	7.999299	7.999218	7.999303	<b>7.999850</b>



**FIGURE 13. The relationship between chi-square value and information entropy to m. (a) The relationship between the chi-square value and m. (b) The relationship between information entropy and m. (c) The histograms of the cipher images when m = 0. (d) The histograms of the cipher images when m = 80. (e) The histograms of the cipher images when m = 500.**



**FIGURE 14. Comparison of information entropy for Lena (256 × 256).**

*d: NIST RANDOMNESS DETECTION FOR CIPHER IMAGES*

A good enough cryptographic system should be able to encrypt any image into a messy one. This is reflected in the independence of the plain image and the cipher image in many statistical characteristics. It is very important to ensure the randomness of the ciphertext when the plaintext is unchangeable. NIST has released a set of test packages NIST SP800-22 for randomness detection of sequences. The test package includes randomness tests for 15 statistical properties, and the P-value and pass rate measure whether the sequence to be tested passes these tests. We use the proposed algorithm to encrypt a total of 200 images in the training set of BSD300 [57] and concatenate the generated cipher images into a bit stream. The bitstream is divided into 123 groups,

**TABLE 5. NIST SP 800-22 randomness test for cipher images.**

Test	P-value	Proportion
Frequency	0.0557	123/123
Block frequency	0.2278	122/123
Cumulative Sums	0.0454	123/123
Runs	0.0290	123/123
Longest run of ones	0.4607	123/123
Rank	0.5938	122/123
FFT	0.4607	121/123
Non-overlapping template	0.5131	122/123
Overlapping template	0.8270	122/123
Universal statistical	0.8690	123/123
Approximate entropy	0.0362	123/123
Random excursions	0.4986	84/85
Random excursions variant	0.5524	84/85
Serial	0.5394	122/123
Linear complexity	0.7482	123/123



**FIGURE 15. Key sensitive analysis diagram. (a) Cipher-image. (b) Use the correct key to decrypt. (c) Decrypt using a key that has been changed by one bit.**

and the length of each group is  $10^6$ . The NIST test results are shown in Table 5. The significance level of the test is 0.01, and when the number of samples is 123, the threshold for passing the test is 118. When the number of samples is 85, the threshold for passing the test is 81. The results in Table 5 show that the generated cipher images have all passed the NIST detection. Therefore, the proposed method can resist statistical attacks very well.

3) SENSITIVITY ANALYSIS

*a: KEY SENSITIVITY ANALYSIS*

The cipher image should change drastically when image encrypted with a slightly changed key. Since the key is generated after the image is encrypted, this does not need to be considered. Furthermore, decrypting with a slightly changed key, the cipher image cannot be decrypted correctly [13]. Based on this, by randomly changing a key for testing. The experimental results are shown in Figure 15. By repeating this experiment, and the experimental results obtained are similar to Figure 15, indicating that the algorithm has excellent key sensitivity.

*b: PLAINTEXT SENSITIVITY ANALYSIS*

The sensitivity of an encryption algorithm to plaintext is a necessary condition for its ability to resist chosen-plaintext attack, and this sensitivity is used in many literatures to characterize the ability to resist differential attacks. Note that this is a necessary condition but not a sufficient one, that is, a good encryption algorithm should have sensitivity to plaintext. This is usually characterized using NPCR and UACI,

which are calculated as:

$$D(i, j) = \begin{cases} 0 & \text{if } C_1(i, j) = C_2(i, j) \\ 1 & \text{if } C_1(i, j) \neq C_2(i, j), \end{cases} \quad (15)$$

$$\text{NPCR} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N D(i, j) \times 100, \quad (16)$$

$$\text{UACI} = \frac{\sum_{i=1}^M \sum_{j=1}^N |C_1(i, j) - C_2(i, j)|}{255MN} \times 100, \quad (17)$$

where  $M$  and  $N$  represent the number of rows and columns of the image respectively, and  $C_1$ ,  $C_2$  represent two cipher images, and their corresponding plain images have only a one-pixel difference. This paper randomly selects a pixel in the plain image to change it. The results are  $\text{NPCR} = 99.6095$ ,  $\text{UACI} = 33.4470$ . These results are very close to the expected value  $\text{NPCR}_E = 99.6094$  and  $\text{UACI}_E = 33.4635$  [58]. This shows that the algorithm has excellent sensitivity to plaintext.

#### D. ROBUSTNESS ANALYSIS

Noise and clipping attacks on images are a common attack way, and image encryption algorithms usually require a certain degree of resistance to them, that is, certain robustness [59]. By using 5%, 15%, and 25% noise interference or shear loss to conduct experiments, respectively, on the cipher image. On the one hand, Figure 16 and Figure 17 are used for qualitative measurement. On the other hand, the indicators of the Mean Square Error (MSE), the Peak Signal-to-Noise Ratio (PSNR), and the Structural Similarity (SSIM) are used for quantitative measurement. The calculation formulas of MSE, PSNR, and SSIM are Eqs. (18) and (19).

$$\begin{cases} \text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [C_1(i, j) - C_2(i, j)]^2 \\ \text{PSNR} = 10 \times \log_{10} \left( \frac{255 \times 255}{\text{MSE}} \right) \end{cases} \quad (18)$$

$$\begin{cases} c_1 = (K_1 L)^2, \quad c_2 = (K_2 L)^2 \\ \text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \end{cases} \quad (19)$$

In Eq. (18),  $C_1$  represents the plain image,  $C_2$  represents the decrypted image after the cipher image is attacked, and  $M$  and  $N$  respectively represent the number of rows and columns of the cipher image. In Eq. (19),  $\mu_x$ ,  $\mu_y$ , represent the mean gray value of image  $x$ ,  $y$ , respectively,  $\sigma_x$ ,  $\sigma_y$  denote the standard deviation of the gray value of  $x$ ,  $y$ , respectively, and  $\sigma_{xy}$  is the covariance of the gray value of  $x$  and  $y$ . In addition,  $K_1 = 0.01$ ,  $K_2 = 0.03$ ,  $L = 255$ , the experimental results are shown in Table 6. The results show that this algorithm has better robustness.

#### E. SPEED PERFORMANCE

The security and efficiency of encryption algorithms are often contradictory. A superior encryption algorithm must ensure the security of encryption and consider the speed of encryption. The size of the image and the key space are variable in this experiment. The test results are shown in Table 7. As the

TABLE 6. Quantitative results of noise and shear attacks.

Methods		Ref. [44]	Ref. [60]	Ref. [45]	Ours	
Noise attack	5%	PSNR	9.3024	11.7385	19.3520	<b>22.2281</b>
		SSIM	0.0177	0.3226	0.8473	<b>0.9188</b>
	15%	PSNR	9.2283	9.5763	14.8006	<b>17.4558</b>
		SSIM	0.0064	0.0549	0.6099	<b>0.7714</b>
	25%	PSNR	9.2034	9.2682	12.7946	<b>15.2383</b>
		SSIM	0.0027	0.0133	0.4361	<b>0.6422</b>
Shear attack	5%	PSNR	13.3281	11.9957	22.0118	<b>22.1679</b>
		SSIM	0.4860	0.3511	0.9150	<b>0.9178</b>
	15%	PSNR	10.2921	9.7298	17.2814	<b>17.4239</b>
		SSIM	0.1463	0.0754	0.7640	<b>0.7702</b>
	25%	PSNR	9.5485	9.3426	15.1286	<b>15.2322</b>
		SSIM	0.0500	0.0221	0.6356	<b>0.6418</b>



FIGURE 16. Image anti-noise test. (a) Decrypted image with 5% noise. (b) Decrypted image with 15%. (c) Decrypted image with 25%.

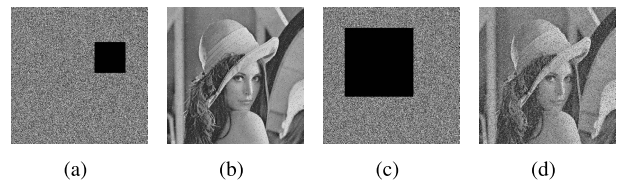


FIGURE 17. Anti-shear test during image decryption. (a) Cipher image cut 5%. (b) The decrypted result of Figure (a). (c) Cipher image cut 25%. (d) The decrypted result of Figure (c).

size of the image increases or the key space increases, the time-consuming algorithm will also increase significantly, but since the key space is variable, the encryption party can weigh security and efficiency according to its own needs. This paper proposes an encryption scheme rather than its efficient implementation. Focusing on the article of Wang et al. [25], they proposed an image encryption algorithm based on associated plain image, and this algorithm uses PWLCM, but it does not use optimization algorithm. This algorithm has a number of similarities with the proposed algorithm, in the case of both using MATLAB programming, the proposed algorithm even when  $m = 80$ , also achieved much better encryption speed than [25] 0.055MB/s, which shows that the proposed algorithm is competitive. Besides, since the key of the decryption process retains the SCD search results of the encryption process, only one cycXOR(.) operation is required to be performed during decryption, which greatly reduces the decryption time.

#### F. COMPARISON WITH META-HEURISTIC BASED ALGORITHMS

We compared the encryption time, entropy, mean correlation, NPCR and UACI with Ref. [29] (Wang et al. based on the simulated annealing algorithm.), [30] (Enayatifar et al.

**TABLE 7. Encryption speed test.**

Image size		Time (s)			
		m = 0	20	40	80
128 x 128	Encryption	0.0011	0.0221	0.0383	0.0849
	Decryption	0.0009	0.0009	0.0009	0.0009
256 x 256	Encryption	0.0036	0.0573	0.0692	0.1178
	Decryption	0.0029	0.0028	0.0029	0.0028
512 x 512	Encryption	0.0149	0.1750	0.1961	0.3042
	Decryption	0.0143	0.0141	0.0129	0.0138
1024 x 1024	Encryption	0.0650	0.8736	1.1556	1.8843
	Decryption	0.0580	0.0649	0.0585	0.0574
Average(MB/s)	Encryption	15.6820	1.10	0.85	0.51
	Decryption	18.1030	17.50	18.29	18.29

**TABLE 8. Comparison with meta-heuristic based algorithms.**

Algorithms	Time (s)	Entropy	Correlation	NPCR	UACI
Ref. [29]	4.307	7.9992	0.004476	<b>99.6110</b>	33.3730
Ref. [30]	0.945	7.9996	0.000600	99.2949	33.4290
Ref. [51]	3.284	7.9997	0.000833	99.2938	33.1803
Ours	<b>0.304</b>	<b>7.9999</b>	<b>0.000546</b>	99.6095	<b>33.4470</b>

based on weighted discrete imperialist competitive algorithm.) and [47] (Enayatifar et al. based on DNA coding and genetic algorithms.). The results are shown in Table 8. It can be seen that the proposed algorithm has the fastest speed, the highest entropy value and the lowest adjacent pixel correlation. This shows that the proposed SCD method based on coordinate descent is suitable for the optimization task of image encryption.

#### IV. CONCLUSION

In this paper, SEA and SCD methods are proposed, and PWLCM is improved to FI-PWLCM. This scheme works by randomly inserting pixel values and using SHA-256 to associate the key with the plain image. The proposed SEA and FI-PWLCM realize the one-to-one mapping between the seed key and the encryption key stream, which is very consistent with the one-time pad. The SCD method can effectively improve the histogram characteristics of the cipher image, make the distribution of pixels at all levels of the cipher image more uniform, and the information entropy is higher. This makes the scheme have better statistical characteristics of cryptography. In addition, this approach can obtain better histogram properties, entropy values and correlation in less time than existing meta-heuristic image encryption algorithms. Several experiments and security analysis show that the algorithm has a large enough key space, and can effectively resist selective text attack, brute force attack, statistical statistics, noise attack and clipping attack. This algorithm is less efficient for image encryption with large size, so in the future, we can reduce the amount of image data through compression, or speed up the encryption process through the technology of block and parallel computing.

#### REFERENCES

- [1] S. Wang, C. Wang, and C. Xu, "An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm," *Opt. Lasers Eng.*, vol. 128, May 2020, Art. no. 105995.
- [2] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Process.*, vol. 148, pp. 124–144, Jul. 2018.
- [3] D. Ravichandran, P. Praveenkumar, and J. B. B. Rayappan, "DNA chaos blend to secure medical privacy," *IEEE Trans. Nanobiosci.*, vol. 16, no. 8, pp. 850–858, Dec. 2017.
- [4] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, pp. 749–761, Jul. 2004.
- [5] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, 2006.
- [6] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, and G. Chen, "A chaos-based image encryption algorithm with variable control parameters," *Chaos, Solitons Fractals*, vol. 41, no. 4, pp. 1773–1783, 2009.
- [7] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.
- [8] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 127–133, 2017.
- [9] Q. Xu, K. Sun, C. Cao, and C. Zhu, "A fast image encryption algorithm based on compressive sensing and hyperchaotic map," *Opt. Lasers Eng.*, vol. 121, pp. 203–214, Oct. 2019.
- [10] M. Zhou and C. Wang, "A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107484.
- [11] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-ElYazeed, "A novel image encryption system merging fractional-order edge detection and generalized chaotic maps," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107280.
- [12] P. Sneha, S. Sankar, and A. S. Kumar, "A chaotic colour image encryption scheme combining Walsh–Hadamard transform and maps," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 3, pp. 1289–1308, 2020.
- [13] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Inf. Sci.*, vol. 546, pp. 1063–1083, Feb. 2021.
- [14] Y. Xian and X. Wang, "Fractal sorting matrix and its application on chaotic image encryption," *Inf. Sci.*, vol. 547, pp. 1154–1169, Feb. 2021.
- [15] 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, Oct./Dec. 2018.
- [16] S. Li and X. Zheng, "Cryptanalysis of a chaotic image encryption method," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, May 2002, pp. II–II.
- [17] E. Solak, C. Çokal, O. Yildiz, and T. Biyikoğlu, "Cryptanalysis of Fridrich's chaotic image encryption," *Int. J. Bifurcation Chaos*, vol. 20, no. 5, pp. 1405–1413, 2010.
- [18] J. Chen, L. Chen, and Y. Zhou, "Cryptanalysis of image ciphers with permutation-substitution network and chaos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2494–2508, Jun. 2021.
- [19] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: CRC Press, 2020.
- [20] X. Wang, Y. Wang, X. Zhu, and C. Luo, "A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level," *Opt. Lasers Eng.*, vol. 125, Feb. 2020, Art. no. 105851.
- [21] S. Zhu, C. Zhu, and W. Wang, "A novel image compression-encryption scheme based on chaos and compression sensing," *IEEE Access*, vol. 6, pp. 67095–67107, 2018.
- [22] Y.-Q. Zhang, J.-L. Hao, and X.-Y. Wang, "An efficient image encryption scheme based on S-boxes and fractional-order differential logistic map," *IEEE Access*, vol. 8, pp. 54175–54188, 2020.
- [23] X. Zhang, Z. Zhou, and Y. Niu, "An image encryption method based on the feistel network and dynamic DNA encoding," *IEEE Photon. J.*, vol. 10, no. 4, pp. 1–14, Aug. 2018.
- [24] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.
- [25] X. Wang, X. Zhu, X. Wu, and Y. Zhang, "Image encryption algorithm based on multiple mixed hash functions and cyclic shift," *Opt. Lasers Eng.*, vol. 107, no. 1, pp. 370–379, Aug. 2017.
- [26] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2322–2335, Jun. 2019.

- [27] M. Kaur, D. Singh, K. Sun, and U. Rawat, "Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map," *Future Gener. Comput. Syst.*, vol. 107, pp. 333–350, Jun. 2020.
- [28] N. K. Sreelaja and G. A. Vijayalakshmi Pai, "Stream cipher for binary image encryption using ant colony optimization based key generation," *Appl. Soft Comput.*, vol. 12, no. 9, pp. 2879–2895, Sep. 2012.
- [29] X. Wang, C. Liu, and D. Xu, "Image encryption scheme using chaos and simulated annealing algorithm," *Nonlinear Dyn.*, vol. 84, no. 3, pp. 1417–1429, 2016.
- [30] R. Enayatifar, A. H. Abdullah, and M. Lee, "A weighted discrete imperialist competitive algorithm (WDICA) combined with chaotic map for image encryption," *Opt. Lasers Eng.*, vol. 51, no. 9, pp. 1066–1077, Sep. 2013.
- [31] K. Mirzaei Talarposhti and M. Khaki Jamei, "A secure image encryption method based on dynamic harmony search (DHS) combined with chaotic map," *Opt. Lasers Eng.*, vol. 81, pp. 21–34, Jun. 2016.
- [32] M. Kaur and V. Kumar, "A comprehensive review on image encryption techniques," *Arch. Comput. Methods Eng.*, vol. 27, no. 1, pp. 15–43, 2020.
- [33] I. Necoara, Y. Nesterov, and F. Glineur, "Random block coordinate descent methods for linearly constrained optimization over networks," *J. Optim. Theory Appl.*, vol. 173, no. 1, pp. 227–254, Apr. 2017.
- [34] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *Ann. Appl. Statist.*, vol. 2, no. 1, pp. 224–244, Mar. 2008.
- [35] K. Fountoulakis and R. Tappenden, "A flexible coordinate descent method," *Comput. Optim. Appl.*, vol. 70, no. 2, pp. 351–394, Jun. 2018.
- [36] H.-J. Michael Shi, S. Tu, Y. Xu, and W. Yin, "A primer on coordinate descent algorithms," 2016, *arXiv:1610.00040*.
- [37] M. Gurbuzbalaban, A. Ozdaglar, P. A. Parrilo, and N. Vanli, "When cyclic coordinate descent outperforms randomized coordinate descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–9.
- [38] A. Patrascu and I. Necoara, "Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization," *J. Global Optim.*, vol. 61, no. 1, pp. 19–46, Jan. 2015.
- [39] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *J. Optim. Theory Appl.*, vol. 72, no. 1, pp. 7–35, 1992.
- [40] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.
- [41] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2137–2150, Sep. 2018.
- [42] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dyn.*, vol. 92, no. 2, pp. 305–313, Apr. 2018.
- [43] T. Wang and M.-H. Wang, "Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding," *Opt. Laser Technol.*, vol. 132, Dec. 2020, Art. no. 106355.
- [44] Z. Hua, B. Xu, F. Jin, and H. Huang, "Image encryption using Josephus problem and filtering diffusion," *IEEE Access*, vol. 7, pp. 8660–8674, 2019.
- [45] T. S. Ali and R. Ali, "A new chaos based color image encryption algorithm using permutation substitution and Boolean operation," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19853–19873, Jul. 2020.
- [46] M. Z. Talhaoui, X. Wang, and M. A. Midoun, "A new one-dimensional cosine polynomial chaotic map and its use in image encryption," *Vis. Comput.*, vol. 37, no. 3, pp. 541–551, 2021.
- [47] R. Enayatifar, A. H. Abdullah, and I. Isnin, "Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence," *Opt. Lasers Eng.*, vol. 56, pp. 83–93, May 2014.
- [48] M. Ghazvini, M. Mirzadi, and N. Parvar, "A modified method for image encryption based on chaotic map and genetic algorithm," *Multimedia Tools Appl.*, vol. 79, nos. 37–38, pp. 26927–26950, Oct. 2020.
- [49] X. Zhang, L. Wang, Z. Zhou, and Y. Niu, "A chaos-based image encryption technique utilizing Hilbert curves and H-fractals," *IEEE Access*, vol. 7, p. 74734–74746, 2019.
- [50] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K.-R. Choo, and Z. Qin, "Deep-KeyGen: A deep learning-based stream cipher generator for medical image encryption and decryption," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4915–4929, Sep. 2022.
- [51] Y. Zhang, "The fast image encryption algorithm based on lifting scheme and chaos," *Inf. Sci.*, vol. 520, pp. 177–194, May 2020.
- [52] B. Ge and H.-B. Luo, "Image encryption application of chaotic sequences incorporating quantum keys," *Int. J. Autom. Comput.*, vol. 17, no. 1, pp. 123–138, Feb. 2020.
- [53] H. Xiang and L. Liu, "A novel image encryption algorithm based on improved key selection and digital chaotic map," *Multimedia Tools Appl.*, vol. 80, no. 14, pp. 22135–22162, Jun. 2021.
- [54] S. Sun, "Chaotic image encryption scheme using two-by-two deoxyribonucleic acid complementary rules," *Opt. Eng.*, vol. 56, no. 11, 2017, Art. no. 116117.
- [55] S. Mozaffari, "Parallel image encryption with bitplane decomposition and genetic algorithm," *Multimedia Tools Appl.*, vol. 77, no. 19, pp. 25799–25819, Oct. 2018.
- [56] M. Mahmud, M. Lee, and J.-Y. Choi, "Evolutionary-based image encryption using RNA codons truth table," *Opt. Laser Technol.*, vol. 121, Jan. 2020, Art. no. 105818.
- [57] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jun. 2001, pp. 416–423.
- [58] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Select. Areas Telecomm.*, vol. 1, pp. 31–38, Apr. 2011.
- [59] Z.-J. Huang, S. Cheng, L.-H. Gong, and N.-R. Zhou, "Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform," *Opt. Lasers Eng.*, vol. 124, Jan. 2020, Art. no. 105821.
- [60] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," *Inf. Sci.*, vol. 480, no. 1, pp. 403–419, Apr. 2019.



**XIYU SUN** received the bachelor's degree from the School of Mechanical Engineering, Hunan Institute of Science and Technology, China. He is currently pursuing the master's degree in electronic information with Hengyang Normal University, Hunan, China. His research interests include chaotic encryption and image processing.



**ZHONG CHEN** received the M.S. degree in applied mathematics from the Department of Applied Mathematics, Southwest Jiaotong University, China, in 2004, and the Ph.D. degree in mechanical engineering from Hunan University, China, in 2018. He is currently an Associate Professor at the School of Computer Science and Technology, Hengyang Normal University, Hunan, China. His research interests include digital image encryption, nonlinear dynamics, and deep learning.

• • •