Multimedia Tools and Applications (2020) 79:2719–2743
https://doi.org/10.1007/s11042-019-08295-5

Plaintext related image hybrid encryption scheme using algebraic interpolation and generalized chaotic map

Xikun Liang[1,2] • Xiao Tan[1,2] • Limin Tao[1,2]

Received: 30 January 2019 / Revised: 6 August 2019 / Accepted: 30 September 2019 /
Published online: 1 December 2019

Abstract

In this paper, a plaintext related image hybrid encryption scheme is proposed based on Lagrange interpolation, generalized Henon map and nonlinear operations of matrices. The proposed scheme consists of three parts. In the first part, a generalized chaotic map is constructed on the basis of Henon map. Using the novel map, a chaotic sequence is built. And then, both the chaotic sequence and the plaintext pixels are used to implement the first nonlinear operation for generating the first cipher matrix associated with the plaintext. By performing an exclusive XOR operation between the original pixels matrix and the first cipher matrix, the diffusion encryption is carried out. In the second part, Lagrange interpolation is used to create the second cipher matrix related to the diffused image; the second nonlinear transformation is developed between the diffused image and the second cipher matrix; and sequence rearrangement is adopted to scramble the diffused image. In the third part, the third nonlinear transformation of matrices based on point operation and rounding operation is implemented on the scrambled image to complete the image encryption. Accordingly, the decryption process is executed by the inverse operations in the opposite order. The proposed algorithm has some distinctive features: a variety of nonlinear tools such as nonlinear polynomial interpolation, nonlinear chaotic map, and nonlinear operations were involved in the scheme. The cryptosystem is designed with the plaintext to enhance the algorithm security. Due to the combination of multiple nonlinear methods and random factors, the scheme is one time pad, which can withstand multiple types of attacks. The algorithm has a clear structure and a simple calculation, so it is easy to program. In addition, encryption simulation and performance analysis are carried out. The feasibility and effectiveness of the algorithm are verified by the simulated results. The security of the algorithm are proved by the objective indicators such as the running time, key space, statistical properties, key sensitivity, and differential analysis, etc.

Keywords Non-linear methods · Lagrange interpolation · Chaotic map · Plaintext related · Image encryption

✉ Xikun Liang
schenken@163.com

Extended author information available on the last page of the article

1 Introduction

With the rapid development of network communication and multimedia technology, information exchange and sharing are more and more frequent. In the meantime, the situation of information security becomes increasingly severe. As the main information carrier, images account for more than 70% of the total amount of daily information. Under a variety of security threats, image encryption has become an important issue. In recent years, image encryption methods [11, 23, 29] mainly represented by chaotic encryption algorithms have attracted extensive attentions, and many important researching results have been achieved.

Adopting sequence diffusion transform, pixels 8-bit decomposition and chaotic scrambling, Zhu, Zhang, Wong et al. [33] constructed the bit-plane scrambling encryption scheme (2011). Applying DNA coding and Chen hyper-chaotic system, Wei, Guo, Zhang et al. [25] discussed DNA image encryption method (2012). Using the chaotic encryption system of three-dimensional Arnold map, Kanso and Ghebleh [13] performed a hybrid algorithm for gray images direct encryption (2012). Combing public key cryptosystem with mixed reality technology, Amalarethinam and Geetha [3] provided a new image encryption scheme (2015). Based on chaotic map, Liu, Sun and Zhu [17] put forward a fast image encryption algorithm (2016). On the basis of block scrambling and dynamic index based diffusion, Xu, Guo, Li et al. [26] proposed a novel chaotic image encryption algorithm (2017). Utilizing SPIHT coding and Chirikov standard map, Fu, Chen, Zou et al. [9] studied a selective compression encryption of images (2017). By means of DNA sequence operations, Chai, Chen and Broyde [4] suggested a novel chaos-based image encryption algorithm (2017).

Since 2012, the chaotic digital image encryption related to plaintext has received special attentions. Because part of the plaintext data is added to the cipher generating pattern, this method enhanced the security of the cryptogram, and thus, highlighted the characteristics of one time pad for encryption. Zhang [27, 28] is one of the early scholars to study the chaotic image encryption associated with the plaintext. He proposed a chaotic cipher system using plaintext data, and suggested a perfect image encryption mode of diffusion-scrambling-diffusion. These works laid a solid foundation for further research. After this, other researchers have developed a series of related studies. Using reverse 2-dimensinal chaotic map and dependant diffusion, Zhang, Wong, Yu, et al. [30] constructed a new image encryption scheme (2013). Based on hash function, Norouzi, Seyedzadeh, Mirzakuchaki et al. [19] designed an image encryption scheme with only two-round diffusion process (2014). On the basis of eight dimensional chaotic cat map, Ganesan and Murali [10] researched image encryption (2014). Adopting temp-value feedback technique, Zhang, Hu, Liu et al. [31] proposed a chaotic image encryption algorithm (2014). Utilizing chaotic and lookup table, Cheng, Yang, Wei, et al. [6] provided a fast image encryption method (2015). Applying chaos and Langton's Ant cellular automation, Wang and Xu [24] put forward a novel image encryption scheme (2015). For medical applications, Kanso and Ghebleh [14] formed an efficient and robust image encryption algorithm (2015). By the aid of chaos maps with Markov properties, Liu, Li, Zhang et al. [16] studied a novel image encryption algorithm (2015).

In general, a secure image encryption system should have the following characteristics [21, 29]: First, the key space is large enough, and the key length is not less than 128bit. Second, the system is sensitive to the key, plaintext, and ciphertext. Third, the ciphertext has statistical properties similar to noise. Fourth, the cryptosystem can effectively resist various types of attacks. Fifth, the encryption and decryption algorithms are efficient and effective. To meet these requirements, some problems should be taken seriously. Firstly, only the encryption scheme related to the keys cannot effectively defend the selected plaintext attack and the known plaintext attack. The research on the algorithm both related to the keys and to the plaintext should be highly valued. Secondly, the traditional chaotic transformation is limited in variety and easy to crack. It is necessary to discuss the new chaotic map to encrypt image. Thirdly, the research of the combination of multiple mathematical methods for the generation of cryptogram should be accelerated.

It is worth pointing out that, as early as 1979, Shamir [22] put forward a key sharing method based on Lagrange interpolation, which opened the applied research of interpolation polynomial in cryptography. In 2012, Akif [2] comprehensively applied Lagrange polynomial, quadtree technology, and RC4 standard to image encryption and achieved good security results.

Using Lagrange interpolation [18], generalized Henon map, matrix nonlinear operations, and the random functions in Matlab [15], we construct a novel image encryption scheme related to plaintext. From the perspective of mathematics, our work focuses on expanding the application of algebraic interpolation, irreversible chaotic map and non-linear transformation in image processing. In view of practical technology, this paper provides a candidate of image encryption method.

The paper consists of the following main parts: The first part introduces the concepts of algebraic interpolation, Lagrange polynomial, and the generating of interpolation cryptographic matrix. The second part constructs a generalized chaotic map based on Henon map [23], and gives the method of the chaotic cipher sequence. The third part discusses three types of matrix nonlinear transformations. The fourth part describes the algorithm in details. The fifth part carries out the encryption simulation. The sixth part gives the performance and security evaluation of the proposed algorithm, and compares the effects of the algorithm with those of other algorithms. The seventh part summarizes the full paper.

2 Algebraic interpolation, Lagrange polynomial and interpolation cipher

2.1 Algebraic interpolation and Lagrange polynomial

Definition 1 Suppose that the continuous function $y = f(x)$ is defined on the interval $[a, b]$. At the $n + 1$ different points $x_i \in [a, b]$, $i = 0, 1, \cdots, n$, $f(x)$ satisfies $f(x_i) = y_i$. If there is a $n\text{-}th$ degree polynomial $P_n(x)$ which satisfies $P_n(x_i) = y_i$, then $P_n(x)$ is called the $n\text{-}th$ degree algebraic interpolation of $f(x)$. The function $f(x)$ is known as the interpolated function. The points x_i are the interpolating nodes. And, $[a, b]$ is the interpolation interval [18].

The basic theory of numerical analysis has proved that the solution of algebraic interpolation is existing and unique. Lagrange polynomial [18] is one of the most common solutions, which is denoted as follows:

$$P_n(x) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_n l_n(x) = \sum_{k=0}^{n} y_k l_k(x), \tag{1}$$

where, $l_i(x)$ are the $n - th$ bases of Lagrange polynomial, which are defined as follows:

$$l_i(x) = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)} = \prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{(x - x_i)}{(x_j - x_i)}, i = 0, 1, \cdots, n. \tag{2}$$

As we known, the bases $l_i(x)$ satisfy the following Kronecker equation:

$$l_i(x_k) = \begin{cases} 1k = i \\ 0k \neq i \end{cases}. \tag{3}$$

In practice, either the quadratic or the cubic Lagrange polynomials are the most widely used. They are respectively expressed as follows:

$$P_2(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x), \tag{4}$$

$$P_3(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) + y_3 l_3(x). \tag{5}$$

2.2 Lagrange interpolation cipher sequence (matrix)

Suppose that the grayscales matrix is $G_{hk} = (g_{ij})_{hk}$, $i = 1, 2, \cdots h, j = 1, 2, \cdots k$, h, $k \in N$; It is easy to convert this matrix into a row vector $L_{1 \times (hk)}$. After using the internal function 'rand' or 'randn' [15] in Matlab to generate the $n + 1$ random integers x_j, $j = 0, 1, \cdots n$ on the interval [1, $h \times k$], we normalize x_j to double precision floating decimals on the interval [0, 1] by the equation $\widetilde{x}_j = \left(x_j - 1\right)/(h \cdot k - 1)$, and record the corresponding pixels by y_j. And then, by means of the interpolating notes $\left(\widetilde{x}_j, y_j\right)$ and Eq. (4), we construct Lagrange polynomial $N_n(x)$, where, $x \in [a, b]$, $a = min\left(\widetilde{x}_j\right)$, and $b = max\left(\widetilde{x}_j\right)$. Furthermore, we uniformly take the samples x_i, $i = 1, 2, \cdots hk$ from the interval $[a, b]$, compute the corresponding $N_n(x_i)$, and rearrange $N_n(x_i)$ into the matrix K_{hk}. Thus, we achieve the interpolating cipher matrix K_{hk}.

In encryption practice, the parameter n, which is the degree of the Lagrange polynomial, can be regarded as an optional key.

3 Henon map, generalized Henon map and chaotic cipher

3.1 Henon map

Henon map [23] is one of the commonly used chaotic systems for image encryption. It is defined as follows:

$$\begin{cases} x(k+1) = 1 - a \cdot x^2(k) + y(k) \\ y(k+1) = b \cdot x(k) \end{cases},$$

where, the parameters a, b are free variables. When $0.54 < a < 2$, $0 < \mid b \mid < 1$, the system is in a hyper chaotic state with two positive Lyapunov exponents. So it is suitable for a cryptogram generating system.

3.2 GHM

Based on Henon map, we construct a novel map named as GHM (**Generalized Henon Map**), which is expressed as follows:

$$\begin{cases} x(k+1) = 1{-}a{\cdot}x^2(k) + b{\cdot}x(k){\cdot}y(k) + c{\cdot}y(k) \\ y(k+1) = d{\cdot}x(k) \end{cases}, \tag{6}$$

where, the parameters a, b, c, d are system variables. It is not hard to proved that this map has one positive Lyapunov exponent when $0.54 < a < 2, 0 < \mid b \mid < 1, 0 < \mid c \mid < 1, 0 < \mid d \mid < 1$, so it can be used to generate cipher codes.

Considering that the subsequent contents do not involve the inverse of GHM, we do not discuss the inverse of GHM here.

4 Matrices nonlinear transformation based on dot power operation

Suppose that the original image is $P = (p_{ij})_{h \times k}$, where, $p_{ij} \in [0, 1]$ are pixels grayscales. The dot power of a matrix [15] is the power operation objected to the elements of the matrix, it is defined as follows:

$$P.^m = \left(\left(p_{ij}\right)^m\right)_{h \times k}, \tag{7}$$

where, .^ is the operator of dot power, and $m \in R$.

Let A_1, A_2, A_3 and A_4 be the nonzero matrices with the size of $h \times k$, the first type nonlinear transformation (FTNT) of the matrix P is expressed as follows:

$$P_1 = f_1(P) = a_1(b_1{\cdot}P + c_1{\cdot}A_1).^{d_1} + e_1{\cdot}A_2, \tag{8}$$

where, the parameters a_1, b_1, c_1, d_1, $e_1 \in R^+$ are optional.

Similarly, the second type nonlinear transformation (STNT) and the third type nonlinear transformation (TTNT) are respectively defined as follows:

$$P_2 = f_2(P) = a_2(b_2{\cdot}P + c_2{\cdot}A_3).^{d_2}, \tag{9}$$

$$P_3 = f_3(P) = a_3{\cdot}P.^{b_3} + c_3{\cdot}A_4, \tag{10}$$

where, the variables a_2, b_2, c_2, d_2, a_3, b_3, $c_3 \in R^+$ are candidate.

Perform the inverse operations of Eqs. (8), (9) and (10) respectively, we can obtain:

$$P = f^{-1}{}_1(P) = \Big(((P_1{-}e_1{\cdot}A_2)/a_1).^{(1/d_1)}{-}c_1{\cdot}A_1\Big)/b_1, \tag{11}$$

$$P = f_2{}^{-1}(P) = \Big((P_2/a_2).^{(1/d_2)}{-}c_2{\cdot}A_3\Big)/b_2, \tag{12}$$

$$P = f^{-1}{}_3(P) = ((P_3{-}c_3{\cdot}A_4)/a_3).^{(1/b_3)}. \tag{13}$$

In addition, for the known nonzero matrix M_{hk}, the F transformation of M_{hk} is denoted as follows:

$$F(M) = M_{hk}{-}floor(M_{hk}), \tag{14}$$

where, the symbol *floor* is the rounding function of Matlab. If we consider $C_d = floor(M_{hk})$ as a constant matrix (called the third cipher matrix), then we can easily derive the inverse of F transformation as follows:

$$M = F(M) + C_d. \tag{15}$$

5 Plaintext related image hybrid encryption algorithm

5.1 Pixels diffusion using GHM

Using GHM in Eq. (6) to generate the chaotic sequence $C_{1 \times (hk)}$, we rearrange it into the matrix C_1 with the size of hk. Performing FTNT in Eq. (8) among the known matrices P,C_1 and the random matrix R_1, we obtain the following result:

$$P_1 = a_1 (b_1 \cdot P + c_1 \cdot C_1).^{d_1} + e_1 \cdot R_1. \tag{16}$$

And then, we rearrange P_1 into a row vector to form the first cipher sequence $\widetilde{C}$, convert the plaintext P into a row vector $\widetilde{P}$, and successively implement the forward and backward diffusion algorithms between the vector $\widetilde{P}$ and $\widetilde{C}$. The diffusion calculations are accomplished by the following XOR operation [29]:

$$Ct_i = Ct_{i-1} \oplus C_i \oplus P_i, i = 1, 2, \cdots hk, \tag{17}$$

$$Ct_i = Ct_{i+1} \oplus C_i \oplus P_i, hk, hk-1, \cdots 1. \tag{18}$$

Where, the matrix Ct is referred to as the diffused image. The initial value of Ct_0 in Eq. (17) and that of Ct_{hk} in Eq. (18) come from the keys. The symbol $\oplus$ denotes the operator of XOR.

In this stage, the parameters a, b, c, d, a_1, b_1, c_1, d_1, and e_1 are the alternative keys.

5.2 Pixels scrambling based on Lagrange interpolating cipher

For the diffused result Ct, we apply Lagrange interpolation in Eq. (4) to deduce the second cipher matrix K_{hk}. After carrying out STNT in Eq. (9) between Ct and K_{hk}, we acquire the matrix as follows:

$$Ct = a_2 (b_2 \cdot Ct + c_2 \cdot K_{hk}).^{d_2}. \tag{19}$$

Furthermore, we sort the row vector $\overline{Ct}$ derived from $\widetilde{Ct}$ by the inner function randperm in Matlab, and record the number of the elements in $\overline{Ct}$ before and after sorting as I_{old} and I_{new}, respectively. For the row vector $\hat{Ct}$ converted from Ct, we firstly sort it by I_{new}, and then, rearrange the sorted result into the matrix Ct with the size of hk. Thus, we obtain the scrambled image Ct.

In this stage, both the variable n and the parameters a_2, b_2, c_2, d_2 are optional keys.

5.3 The encryption based on TTNT

Performing TTNT in Eq. (10) between the scrambled image Ct and the random matrix R_2, we obtain the following result:

$$Ct_f = a_3 \cdot Ct.^{b_3} + c_3 \cdot R_2. \tag{20}$$

For the matrix Ct_f, we implement the transformation in Eq. (14) to achieve:

$$Ct_F = Ct_f\text{–}floor\left(Ct_f\right). \tag{21}$$

At this point, the encryption process is completed. The matrix Ct_F is the final encrypted image.

In this stage, the parameters a_3, b_3 and c_3 serve as keys. The third cipher matrix C_d is interpreted as a procedural cryptogram.

5.4 Decryption algorithm

The image decryption can be fulfilled in opposite order by performing the inverse transformations and operations. By means of Eqs. (15) and (13), we successively transform the encrypted image Ct_F into the matrices Ct_{F1} and Ct_{F2}. Using Ct_{F2} to form a row vector, we rearrange the vector according to I_{old}, and then change the result into the matrix Ct_{F3} with the size of hk. Thus, the scrambling decrypted image is gotten by the matrix Ct_{F3}. Furthermore, we convert Ct_{F3} into the row vector CT, and carry out the inverse operation of forward and backward diffusing algorithms by following two equations:

$$P_i = CT_{i\text{–}1} \oplus CT_i \oplus C_i, i = 1, 2, \cdots hk, \tag{22}$$

$$P_i = CT_{i+1} \oplus CT_i \oplus C_i, i = hk, hk\text{–}1, \cdots 1. \tag{23}$$

And then, we rearrange the result into the matrix P_0 with the size of hk. The matrix P_0 is the expected diffusion decryption image, and it is also the final decrypted image.

It should be specially noted that, in the process of encryption and decryption, there is a specific matching between the various operations and the data types. If necessary, the conversion between the double precision data and 8-bit integer data should be executed in time to satisfy the requirements of the calculations.

5.5 Algorithm description

5.5.1 The encryption algorithm

The encryption algorithm includes the following steps:

Step 1 **Image preprocessing**

$\bigstar_1$ Convert the original image into the pixels gray matrix P_{hk}.

Step 2 **Diffusion encryption**

$\bigstar_2$ Set the parameters a, b, c, and d respectively, generate the chaotic sequence $C_{1\times(hk)}$ using Eq. (6), and rearrange $C_{1\times(hk)}$ to the matrix C_1 with the size of hk.
$\bigstar_3$ Apply the function rand of Matlab to form the matrix R_1 with the same size of P_{hk}.
$\bigstar_4$ Assign the parameters a_1, b_1, c_1, d_1, and e_1 respectively. Perform FTNT among the matrices $P_{hk,}$ C_1 and R_1 to obtain the matrix P_1.

$\bigstar_5$ Rearrange P_1 into the first cipher sequence $\widetilde{C}$, and covert P_{hk} into a row vector P. For the vectors P and $\widetilde{C}$, successively implement the operations in Eqs. (17) and (18) to infer the vector Ct, and then change Ct into the matrix Ct_{hk}.

In this step, a, b, c, d and a_1, b_1, c_1, d_1, e_1 are candidate keys and the matrix Ct_{hk} is the diffused image.

Step 3 **Scrambling encryption**

$\bigstar_6$ Form the second cipher matrix $K_{hk}$by Eq. (4), and carry out SINT in Eq. (19) between Ct_{hk} and K_{hk} to deduce the matrix $\widetilde{Ct}$.

$\bigstar_7$ Convert $\widetilde{Ct}$ into the row vector $\overline{Ct}$, sort $\overline{Ct}$ using the function randperm of Matlab, and record the elements number of $\overline{Ct}$ before and after sorting as I_{old}andI_{new}, respectively.

$\bigstar_8$ Change Ct_{hk} into the row vector $\hat{C}t$, rearrange $\hat{C}t$ according to I_{new}, and transform the result to the matrix Ct_{hk}.

In this step, the candidate keys include a_2, b_2, c_2, d_2, n. The matrix Ct_{hk} is the scrambled image.

Step 4 **Encryption using nonlinear transformation**

$\bigstar_9$ Use the function rand to generate the random matrix R_2, and perform TTNT in Eq. (20) between Ct_{hk} and $R_2$to derive the matrix Ct_f.

$\bigstar_{10}$ Utilize the third cipher matrix C_d to implement the transformation in Eq. (21) to receive the matrix Ct_F. The matrix Ct_F is the final encrypted image.

In this step, the parameters a_3, b_3, c_3 are optional keys. The matrix Ct_F is the final encrypted image.

5.5.2 The decryption algorithm

Based on the encrypted image Ct_F, the decryption process can be fulfilled as follows:

Step 1 **Decryption of nonlinear transformation**

$\blacklozenge_1$ Convert the matrix Ct_F into the matrix Ct_{F1} by performing the transformation in Eq. (15), and implement the transformation in Eq. (13) to achieve the matrix Ct_{F2}.

Step 2 **Scrambling decryption**

$\blacklozenge_2$ Transform Ct_{F2} into a row vector, rearrange the vector according to I_{old}, and then change the rearranged result into the matrix Ct_{F3} with the size of hk.

Step 3 **Diffusion decryption**

$\blacklozenge_3$ Rearrange Ct_{F3} to the row vector CT, successively implement the transformations in Eqs. (22) and (23), and then rearrange the operating result into the matrix P_0 with the size of hk.

Thus, we obtain the final decrypted image presented by the matrix P_0.

5.6 The diagram of the proposed algorithm

The process of encryption and decryption are now demonstrated in Figs. 1 and 2, respectively.

6 Encryption simulation

The original images are downloaded from the website of BSD-S500 [8], and all of the experiments are conducted under the MATLAB 2016b integrated environment. The software is run on the platform with Intel Core (TM) i7 CPU (2.4 GHz), 8.0-GB RAM, and 64-bit operating system Win7.1.

The sizes of the three experimental images Tiger, Pilots, Pyramid are 342 × 512, 512 × 342, and 684 × 1024 pixels, respectively. The encryption and decryption algorithms are run in the key space of $\Omega_1 \times \Omega_2 \times \Omega_3 = (a, b, c, d, a_1, b_1, c_1, d_1, e_1) \times (n, a_2, b_2, c_2, d_2) \times (a_3, b_3, c_3)$.

Figures 3, 4 and 5, respectively exhibit the encrypted and decrypted results of the three sample images, where, $a = 1.2$, $b = 0.5$, $c = 0.8$, $d = 0.314$, $n = 3$, $a_1 = 20$, $b_1 = 3$, $c_1 = 2.5$, $d_1 = 0.5$, $e_1 = 4$, $a_2 = 2$, $b_2 = 3$, $c_2 = 2$, $d_2 = 4$, $a_3 = 30$, $b_3 = 2$, $c_3 = 3$. In these figures, (a) denotes the original image; (b) shows the encrypted image of (a); and (d) presents the recovered image of (b).

From these figures, we observe that the cipher images are messy and rambling, so the attackers could hardly obtain any useful information from the ciphertexts.

In order to evaluate the encryption and decryption effect of the algorithm objectively, we calculate the absolute values of PSNR (Peak-Signal to Noise Ratio) and SSIM (Structural Similarity) between the encrypted images and the original ones, as well as those between the decrypted images and the original ones. The data are shown in Table 1. Where, P_{01}, S_{01} present the PSNR, the SSIM between the encrypted image and the original one, respectively; P_{02}, S_{02} indicate the PSNR, the SSIM between the decrypted image and the original one, respectively.

The data shown in Table 1 prove that the algorithm can achieve good encryption and decryption effects. In addition, after performing the decryption algorithm on the decrypted images, we obtain the same results as the original images. Hence, we confirm that the proposed algorithm is a lossless encryption method.

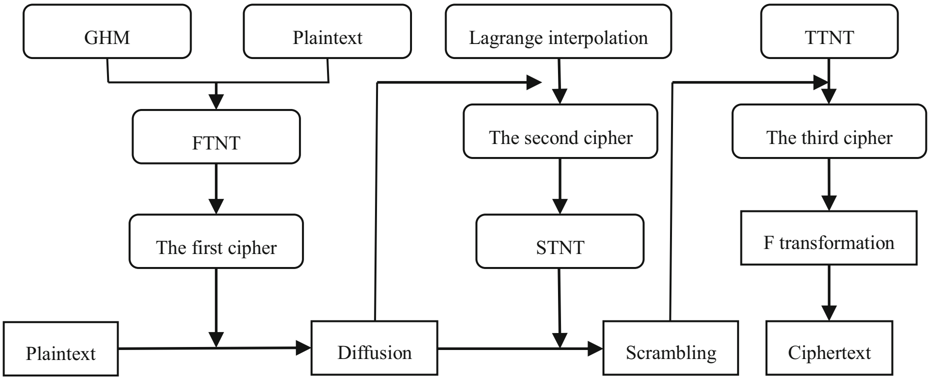


Fig. 1 The encryption diagram

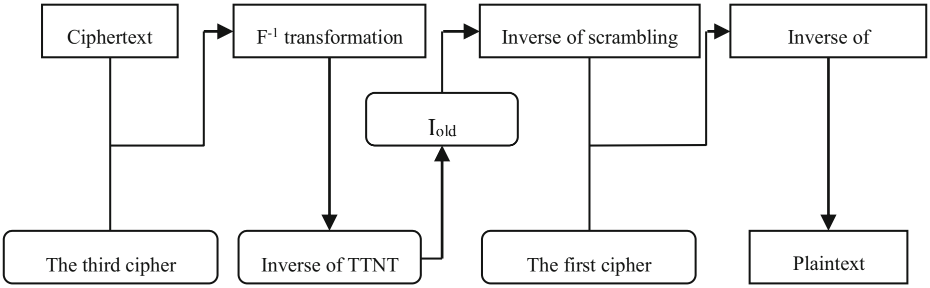


Fig. 2 The decryption diagram

7 Encryption performance analysis and security

In this section, we discuss several performance indicators of the proposed algorithm such as the time consuming and space occupying of the algorithm, the size of the key space, the statistical properties of the images, key sensitivity, plaintext sensitivity, etc. These indexes objectively reflect the efficiency and security of our scheme.

It is necessary to explain that, since the numerical calculations and random factors are involved in the algorithm, there may be differences in analysis data obtained from different experimental rounds, but these differences in quantity have nothing to do with the final conclusions.

7.1 Time and space consumption of the algorithm

For the five images selected from large scale datasets [1, 7, 8] with the sizes of 32 × 32, 256 × 256, 342 × 512, 512 × 512 and 1024 × 684 pixels, respectively, we record the average running time of the algorithm after 500 times tests. The results are shown in Table 2. In addition, the total amount of encoding, the size of the files and the storage spaces are also listed in this table. The data indicates that the algorithm takes up limited system resources and has high operating efficiency. Note that, since the running time of the algorithm is related to the computer system

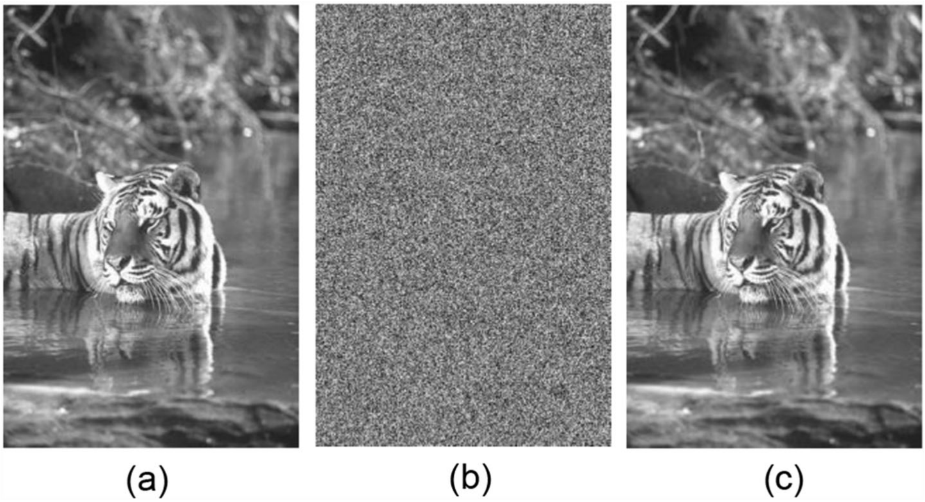


Fig. 3 The encrypted and decrypted result of the image Tiger

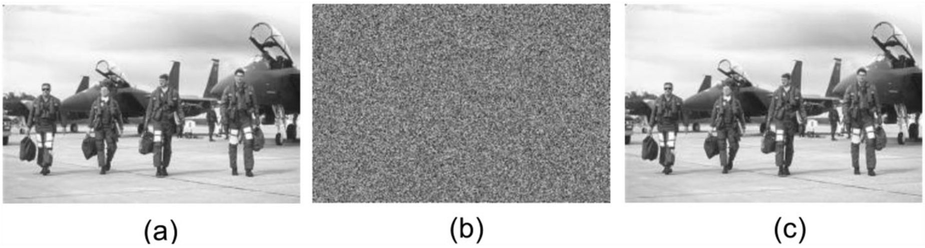


Fig. 4 The encrypted and decrypted result of the image Pilots

configuration and the coding volume and the storage space are subject to programming languages, the data in Table 2 is not absolute and just for reference.

According to the data in the table, we fit out the encryption and decryption time curves using quadratic polynomial. The results are demonstrated in Fig. 6. In the integrated environment of Matlab 2016b, the memory occupied by running the encryption and decryption files of this size can be neglected, while the memory occupied by the platform is approximated to 560 MB. Thus, the system resources used by the encryption scheme are limited. A conventionally configured personal computer can meet this demand.

7.2 Resistance to the brute force attack

7.2.1 Keys space

The set of all possible values of the keys is called the key space. It is one of the important indexes to measure whether an image encryption algorithm can resist brute force attack or not. For 8 bit integer images, the applied cryptography has affirmed that the secure key space of an encryption algorithm should be not less than 128 bit. Otherwise the encryption scheme will be broken by exhaustive search to get the secret keys in a limited amount time.

As mentioned above, the key space of the proposed algorithm is $\Omega_1 \times \Omega_2 \times \Omega_3$. If all of the parameters are taken as double precise decimals, then they should be set to be $O\,(10^{-14})$. So the key space of the algorithm is not less than $log_2(10^{238}) \approx 790$bit. In the encryption practice, the conservative range of the parameters is $[10^{-4}, 10^4]$, so the key space is not less than $log_2(10^{136}) \approx 451$bit. Such a key space is large enough to ensure the algorithm to resist the brute force search for secret keys. In addition, the key space of the proposed algorithm is obviously larger than those of the algorithms in [5] (about 144 bit) and [20]. This indicates that the proposed algorithm is more secure than the later two with respect to resistance to brute force attack.

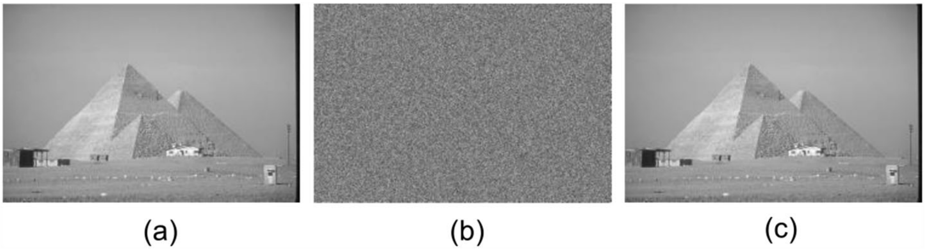


Fig. 5 The encrypted and decrypted result of the image Pyramid

Table 1 The PSNRs and SSIMs of different images

Images	Tiger	Pilots	Pyramid
Image sizes	342 × 512	512 × 342	684 × 1024
P_{01}	4.7985	4.8218	4.8122
S_{01}	3.6274×10^{-8}	3.5264×10^{-8}	9.7235×10^{-9}
P_{02}	107.2011	107.4921	107.0527
S_{02}	0.9996	0.9994	0.9991

7.2.2 Keys sensitivity

Keys sensitivity is another important index to measure the ability of an algorithm to resist brute force attacks. In general, using two sets of keys with small difference to encrypt the same plain image, if we obtain two cipher images which have great difference, we consider that the encryption algorithm is sensitive to the keys. Obviously, the more sensitive the algorithm is to the keys, the higher the security of the algorithm.

As mentioned above, the keys of the proposed algorithm include: a, b, c, d, a_1, b_1, c_1, d_1, e_1, n, a_2, b_2, c_2, d_2, a_3, b_3, c_3. Without loss of generality, we chose the parameters a, b_1, c_2, and c_3 for key sensitivity testing. For each parameter, we set the variation as 10^{-10}, and let

$$\begin{aligned}
K_0 &= (a,b,c,d,a_1,b_1,c_1,d_1,e_1,n,a_2,b_2,c_2,d_2,a_3,b_3,c_3);\\
K_1 &= \left(a+10^{-10},b,c,d,a_1,b_1,c_1,d_1,e_1,n,a_2,b_2,c_2,d_2,a_3,b_3,c_3\right);\\
K_2 &= \left(a,b,c,d,a_1,b_1+10^{-10},c_1,d_1,e_1,n,a_2,b_2,c_2,d_2,a_3,b_3,c_3\right);\\
K_3 &= \left(a,b,c,d,a_1,b_1,c_1,d_1,e_1,n,a_2,b_2,c_2+10^{-10},d_2,a_3,b_3,c_3\right);\\
K_4 &= \left(a,b,c,d,a_1,b_1,c_1,d_1,e_1,n,a_2,b_2,c_2,d_2,a_3,b_3,c_3+10^{-10}\right).
\end{aligned}$$

Applying the parameter values used in Section 6, we respectively encrypt the image Tiger using the keys K_0, K_1, K_2, K_3, and K_4. The ciphertexts are shown in Figs. 1c, and 7a, b, c, and d, respectively. And also, the difference figures between Figs. 7a, b, c, d and 1c are exhibited in Figs. 6e, f, g, and h, respectively. Furthermore, we calculate some objective comparison indexes between the later four ciphertexts and the first ciphertext. These indexes include the correlation coefficients, SSIM, NPCR (Number of Pixels Change Rate), UACI (Unified Average Changing Intensity) and BACI (Block Average Changing Intensity), which are fully listed in Table 3.

As can be seen in Fig. 7, while keeping other parameters constant, a small change in one key resulted in a great difference in ciphertexts. This shows that the proposed encryption scheme is quite sensitive to the key. Meanwhile, Table 3 reveals that the correlations and SSIM are almost zero, and the other indexes approach to the ideal values. Thus, we conclude that the algorithm has strong ability to resist exhaustive attack.

Table 2 Time and space consumption of the proposed algorithm

Images	1	2	3	4	5
Sizes (pixels)	32 × 32	256 × 256	342 × 512	512 × 512	684 × 1024
Average encryption time (s)	0.7816	1.9054	2.5096	3.7109	5.9196
Average decryption time (s)	0.7533	1.8453	2.4477	3.1101	5.7581
Total Code Volume (Matlab lines)	188				
Total Code Volume (Number of Characters)	3380	3542	3701	3916	4168
File sizes (kb)	5.09	5.25	5.41	5.62	5.83
Storage space (kb)	8.0				

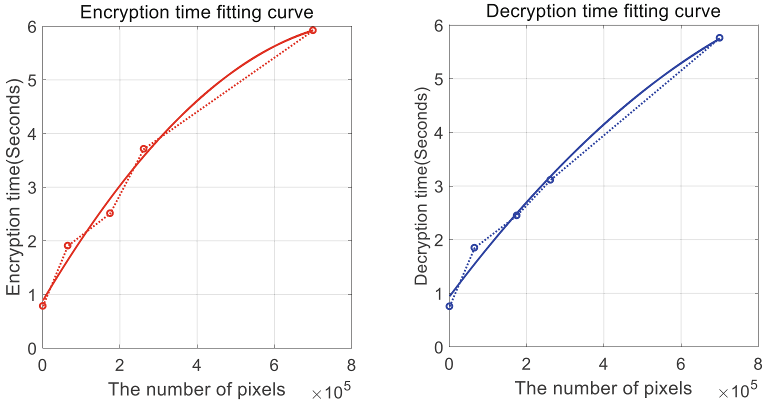


Fig. 6 The fitting curves of encryption and decryption time

7.3 Resistance to the statistic attack

7.3.1 Gray histogram and gray surface

Generally, the gray distribution of the ciphertext obtained by a robust encryption algorithm should be uniform, which is usually reflected in the ciphertext gray histogram. Meanwhile, another intuitive tool, the gray surface, can be either used to reveal this phenomenon. The gray surface is a spatial graphics, in which the pixels coordinates constitute a plane grid, and every pixel grayscale denotes the altitude of each vertex for the grid. Corresponding to the uniformity of the gray histogram, the gray surface should put up the shape of equal height almost everywhere.

Figure 8 shows the gray histograms of the plaintext and ciphertext for the three experimental images. Figure 9 is the corresponding gray surfaces.

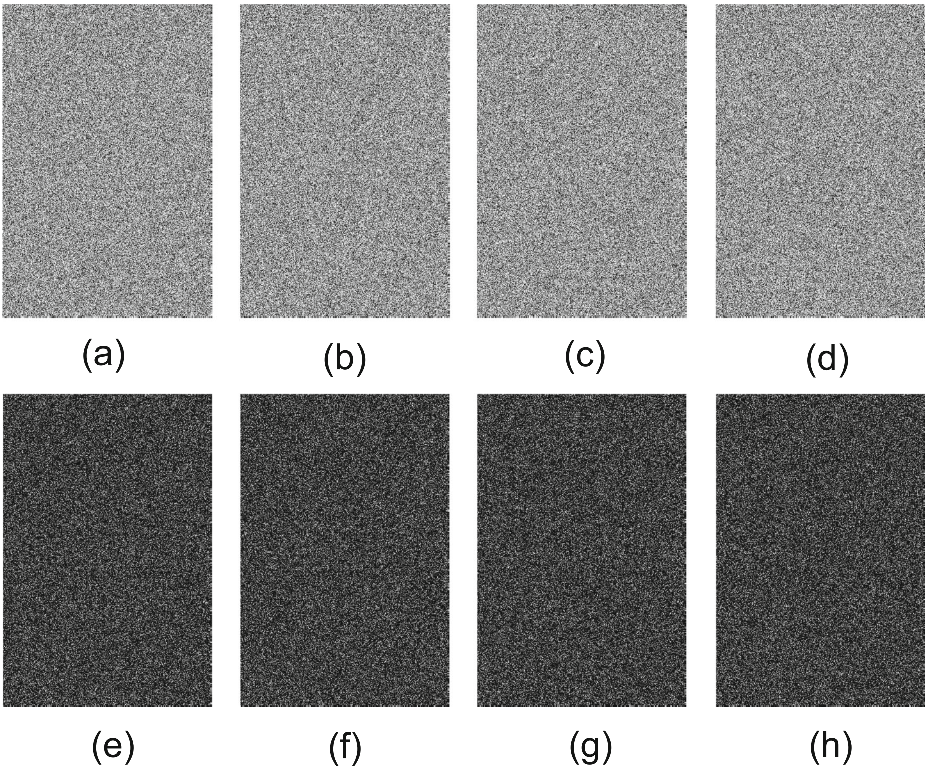
Figure 8 reveals that the gray histograms of the plaintexts have a large fluctuation, whereas those of the ciphertexts are flat and uniform. This means the statistical attacks are impossible. Meanwhile, the same conclusion can be drawn from the gray surfaces in Fig. 9.

7.3.2 Pixels correlation

In general, the pixels of a plain image with clear meanings have a certain correlation, but an effective and safe encryption algorithm can successfully eliminate this correlation. That is, the pixels correlation of the corresponding cipher image is uncertain.

Table 3 Correlation, SSIM, NPCR, UACI and BACI of the ciphertexts before and after the change of the keys

Indexes	K_1-K_0	K_2-K_0	K_3-K_0	K_4-K_0	Ideal values
correlation	−0.0010	7.2569×10^{-4}	0.0037	0.0015	0.0000
SSIM	0.0050	0.0065	0.0092	0.0065	0.0000
NPCR	99.6242%	99.6088%	99.6145%	99.6065%	99.6094%
UACI	33.3545%	33.3176%	33.2678%	33.3122%	33.4636%
BACI	26.6567%	26.6976%	26.6576%	26.6886%	26.77124182%



Taking the image Tiger as an example, we now discuss the pixels correlation of the plaintext and ciphertext in the horizontal, vertical, and diagonal directions, respectively. For the plaintext, we randomly select 3000 pixels in the three directions, respectively calculate the absolute values of their correlation coefficients, and list them in Table 4. Then we draw their distribution diagrams as shown in Fig. 10a, b, and c, respectively. As a contrast, the corresponding data of the ciphertext are also given in Table 4, and the diagrams are shown in Figs. 10d, e, and f, respectively.

From Fig. 10 we find that the plaintext pixels exhibit approximately linear distribution in all of the three directions, whereas the distributions of the ciphertext pixels are not clear. This indicates the encryption algorithm effectively destroyed the pixels correlation of the plaintext. Thus, the statistical attacks are invalid. This conclusion is also supported by the data in Table 4.

7.3.3 Information entropy

Information entropy (IE) is the measure of the disorder state for an information system. It reflects the randomness and unpredictability of an information sources. For an 8-bit integer image, the ideal value of IE is 8. In fact, the closer the IE of the cipher image is to the ideal value, the stronger the robustness of the algorithm.

The information entropies of the plaintexts and ciphertexts for the three experimental images are now shown in Table 5. The data show that the information entropies of the

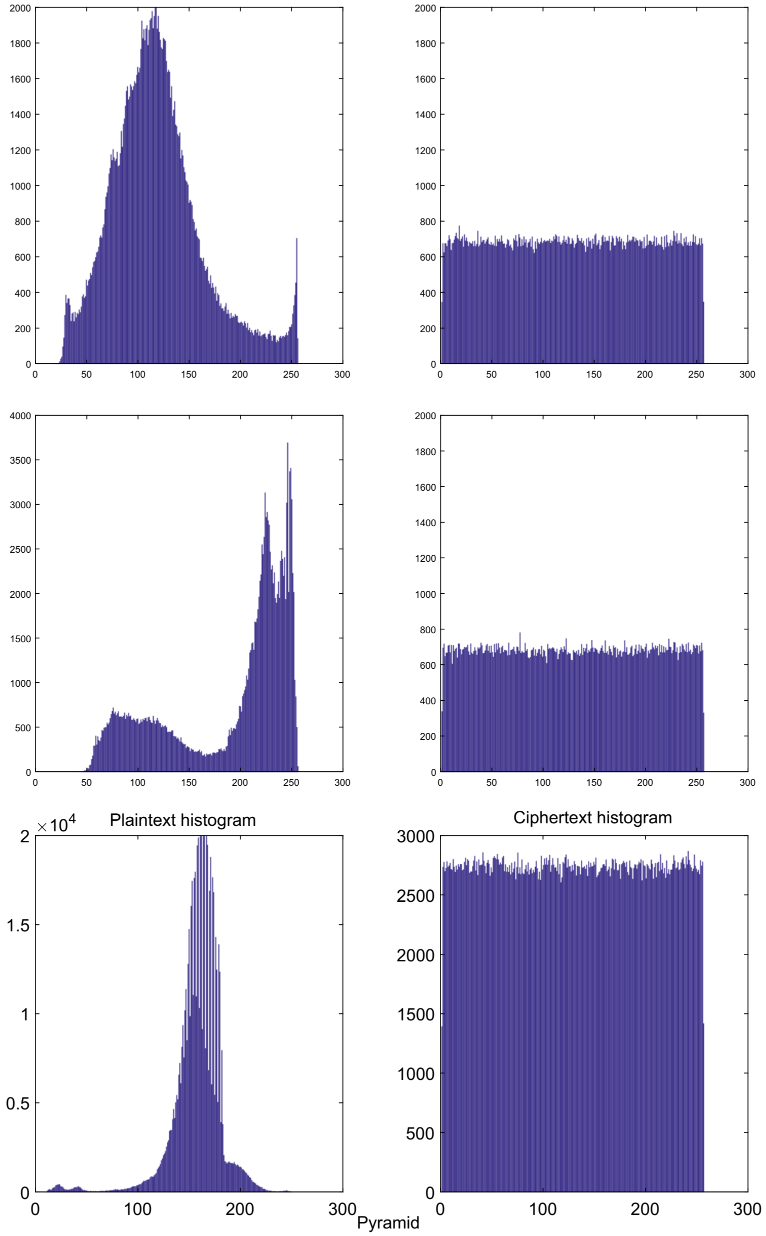
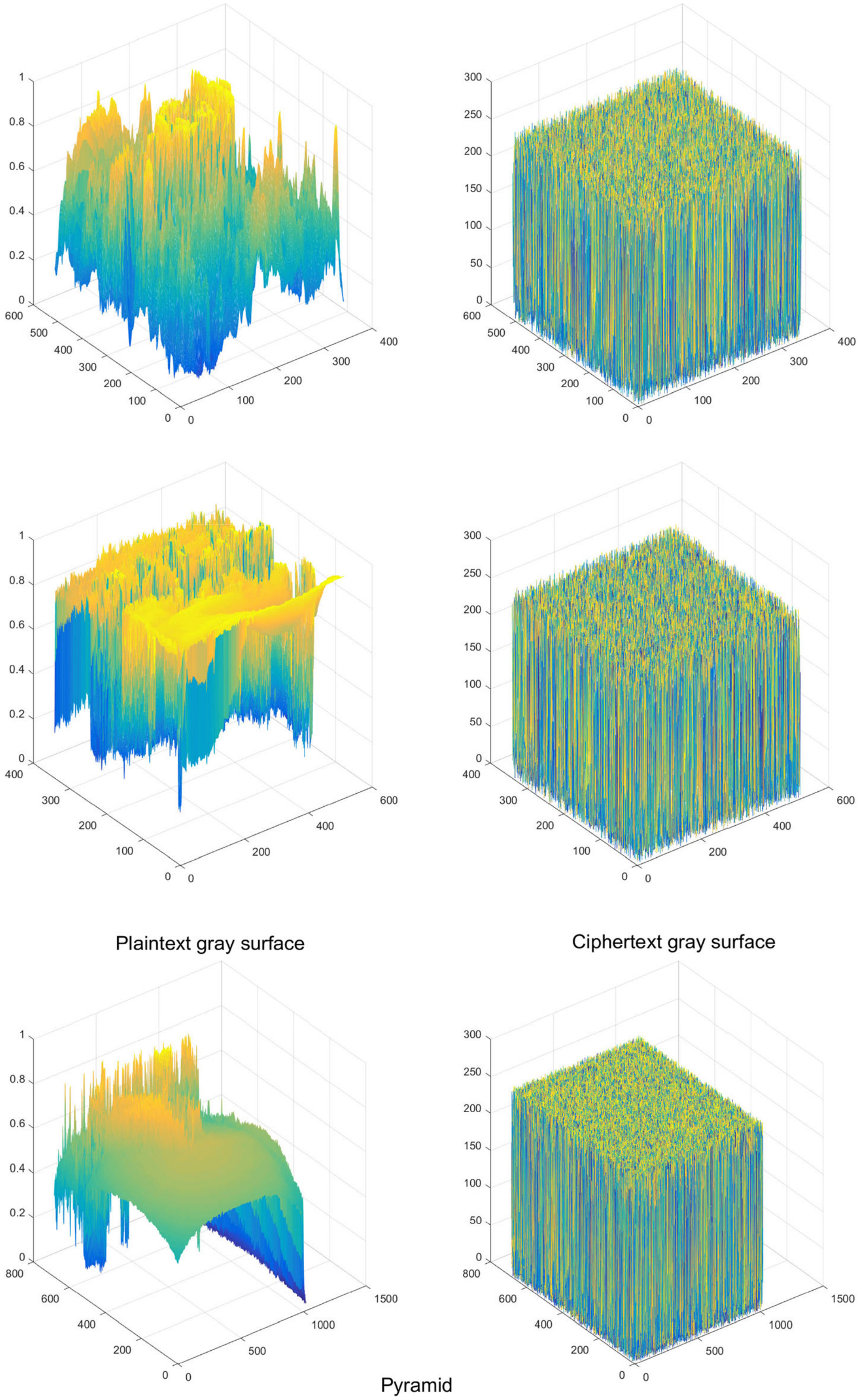


Fig. 8 Gray histograms of plaintext and cipher images

ciphertexts are very close to 8. Thus, it is inferred that the proposed algorithm is robustness and strong enough to resist statistical attack. Meanwhile, the data also proves that the proposed algorithm is superior to that mentioned in [20] in the sense of information entropy.


Plaintext gray surface

Ciphertext gray surface

Pyramid

Fig. 9 Gray surfaces of plaintext and cipher images

Table 4 Correlations of the plaintext and cipher images in the three directions

Direction	Horizontal	Vertical	Diagonal
Plaintext	0.9531	0.9569	0.9345
Ciphertext	0.0094	0.0081	0.0105

7.4 Resistance to differential attack and plaintext sensitivity

Under the same condition of keys, plaintext sensitivity focuses on the influence extent of the quantitative changes in plaintext on ciphertext. Using $P_0, P_0 + \Delta P, C_{P_0}$, and $C_{P_0+\Delta P}$,

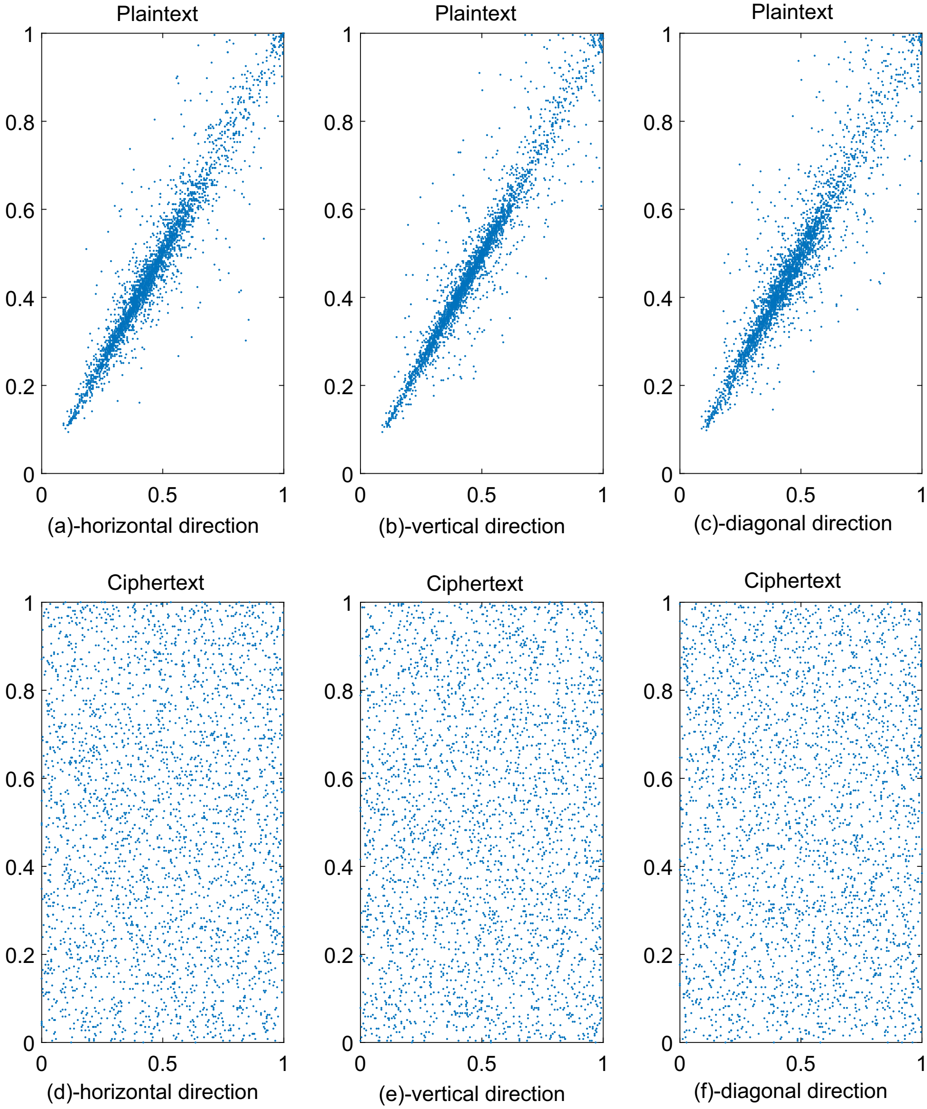


Fig. 10 The plaintext and ciphertext pixels distributions of the image Tiger

Table 5 Entropies of the plaintext and cipher images

Image	Tiger	Pilots	Pyramid
Size (Pixels)	342 × 512	512 × 342	684 × 1024
The IE of the plain image in this paper	7.4109	7.1581	7.3069
The IE of the cipher image in this paper	7.9973	7.9969	7.9980
The IE of the cipher image in [20]	7.4109	7.1581	7.3069

respectively to represent the plaintexts and the corresponding ciphertexts before and after the variation, the plaintext sensitivity analysis is to find the differences between C_{P_0} and $C_{P_0+\varDelta P}$ when $\varDelta P$ is taken as a small numerical value. If the difference is great, then the algorithm is considered to be sensitive to the plaintext.

Respectively increasing the gray values of the three experimental images by 10^{-10}, we calculate the correlation coefficients, SSIM, NPCR, UACI, and BACI of the two cipher images before and after the change. The results are shown in Table 6. The data state clearly that the correlations and SSIM of the ciphertexts are very close to zero. Meanwhile, the values of NPCR, UACI, and BACI are very close to the ideal values, respectively. This demonstrates that the proposed algorithm is sensitive to plaintext and can effectively defend against differential attacks.

7.5 Resistance to chose-plaintext attack

The practice of image encryption has proved that pure scrambling encryption or sequence encryption is insecure for chose-plaintext attack. Compared with these simple encryption methods, the proposed scheme adopts such an encryption pattern of diffusion- scrambling-nonlinear transformation to improve the security for chose-plaintext attacked. From the differential attack test, we find that even a light change in plain image should cause a huge change in the cipher image. This fully shows that the algorithm is very sensitive to the change of plaintext. And what's more, as mentioned in the Section 2.2, the generating of Lagrange interpolation cipher sequence is based on the two important factors, that are random components and plaintext relation. This feature guarantees that the algorithm is still secure against chose-plaintext attacks even when other part of the key is cracked.

7.6 Comparison with other schemes

In order to evaluate the algorithm more objectively, we compare some performance indicators of our scheme with that of two other image encryption methods published in recent years, namely, Refs. [5, 12, 32]. The indexes are listed in Table 7, which are obtained based on the image Tiger.

Table 6 Correlation, SSIM, NPCR, UACI, and BACI of the ciphertexts before and after the variation of the plaintexts

Indexes	Tiger	Pilots	Pyramid	Ideal values
correlation	−0.0022	-9.5294×10^{-4}	-1.8331×10^{-4}	0.0000
SSIM	0.0034	0.0048	0.0056	0.0000
NPCR	99.5951%	99.6174%	99.6161%	99.6094%
UACI	33.3744%	33.3580%	33.3509%	33.4636%
BACI	26.6951%	26.5986%	26.6841%	26.7712%

Table 7 Performance comparison of the proposed algorithm with other schemes

		Ref. [5]	Ref. [32]	Ref. [12]	Our algorithm	Ideal value
Key space		144 bit	259 bit	449bit	451 bit	Less than 128 bit
Information entropy	–	7.9968	7.9992	7.9970	7.9973	8.0000
Correlation analysis	Horizontal	−0.0048	0.0183	0.0178	0.0094	0
	Vertical	−0.0039	0.0170	0.0086	0.0081	0
	Diagonal	−0.0098	0.0217	0.0201	0.0105	0
Differential analysis	NPCR	99.62%	99.59%	–	99.5951%	99.6094%
	UACI	33.46%	33.45%	–	33.3744%	33.4636%

From this table, we can see that the key space of our algorithm is the biggest among the four schemes. This reveals that our method is optimal for resisting brute force attack. The data in line 2 show that the information entropy values of the three schemes are all very close to 8.0. Although the value of our scheme is slightly smaller than that of Ref. [12, 32], it has little influence since which can guarantee no information leak of cipher image. For correlation indicators in horizontal, vertical and diagonal direction, our method is superior to Ref. [12, 32] and inferior to Ref. [5], but this slight difference has little influence since they are all extremely close to zero. As for differential analysis, the values of NPCR and UACI of the three schemes are all very close to the ideal values, so all the four schemes can resist differential attack effectively. In contrast, our algorithm takes advantages of mathematical methods not commonly used in cryptography, such as non-linear interpolation, irreversible chaotic system, non-linear transformation, etc., which can achieve high security with simple structure. Thus, the proposed scheme demonstrates some advantages and is expected to be applied to image encryption practice.

7.7 Performance evaluation on large scale datasets

Only the three experimental images can not reveal all the truth. In order to strongly prove the effectiveness and security of our method, we use two large scale image datasets to evaluate the proposed algorithm. The Cifar-10 [7] dataset contains images of objects belonging to 10 categories, with 6000 images per category. Another image processing dataset BSD500 [8] includes 800 images belong to various species and scenes. For performance evaluation, we randomly pickup 5 images per category from Cifar-10 and other 50 images from BSD500 respectively, which forms a testing subset of 100 images. We encrypt all the 100 images using the proposed algorithm and calculate the main performance data. Limited to the space, the

Table 8 Statistical results of the indexes for performance and security

Index	Sub-index	Distribution interval	Mean	Variance	Ideal value
Information entropy	–	[7.7811, 7.9974]	7.9013	0.0098	8.0000
Correlation analysis	Horizontal	[0.0024, 0.0651]	0.0272	0.0004	0
	Vertical	[0.0030, 0.0904]	0.0300	0.0008	0
	Diagonal	[0.0008, 0.0540]	0.0203	0.0003	0
Differential analysis	CORL	[−0.0435, 0.0829]	0.0031	0.0006	0
	SSIM	[−0.0413, 0.0852]	0.0089	0.0006	0
	NPCR	[99.3164, 99.8047]	99.6047	0.0166	99.6094%
	UACI	[32.0293, 34.0648]	33.2427	0.2446	33.4636%
	BACI	[25.6772, 27.5603]	26.6648	0.1665	26.7712%

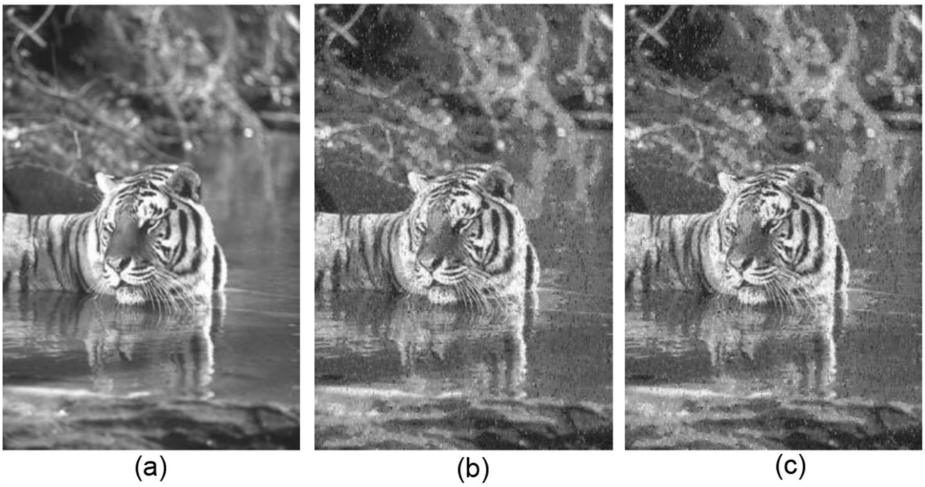


Fig. 11 Decrypted results of the cropped cipher for the experimental image Tiger

encrypted images and detailed data of the various indexes are omitted here. The statistical results of the indexes are shown in Table 8. In this table, the sub-index CORL of differential analysis means the correlation coefficients of the two cipher images before and after the plain image variation. The data demonstrate that the mean of every indicator is very close to the ideal value and the distribution of the data is reasonable, uniform, and stable. So, our encryption scheme is effective and safe for large scale image datasets.

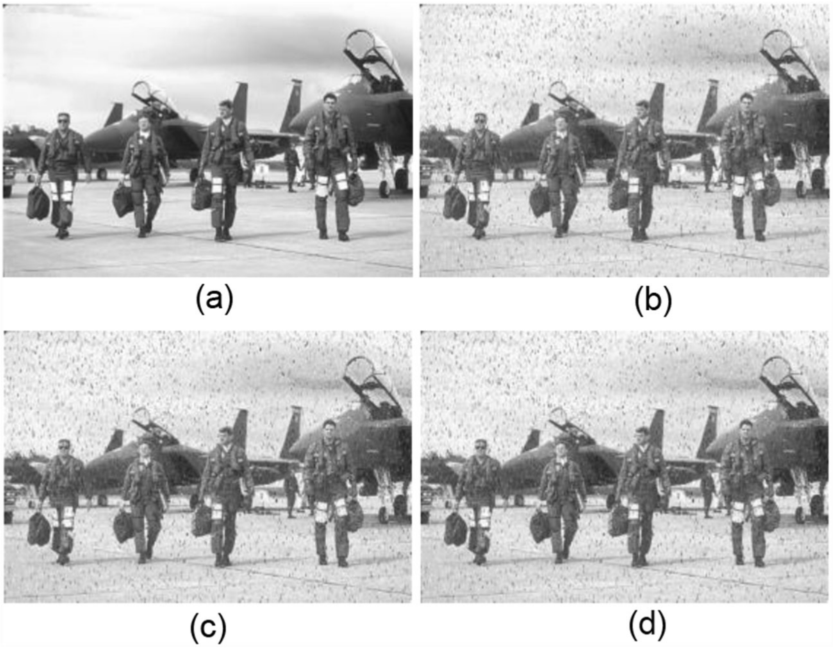


Fig. 12 Decrypted results of the noised cipher for the experimental image Pilots

7.8 Decryption of cropped, noised, and compressed cipher image

In this subsection, we discuss whether a cipher can be correctly de decrypted if it is cropped, noised, of compressed. For the three images used in Section 6, we carry out the decryption experiment. The results are exhibited in Figs. 11, 12, and 13, respectively.

In Fig. 11, (a) is the initial image of Tiger, (b) is the decrypted result of the cipher in which 80,000 random pixels are replaced with the Matlab constant ‘Nan’, (c) is the decrypted image of the cipher in which an image block is substituted by ‘Nan’. The block ranges from the 10000th pixels to the 40000th pixel.

In Fig. 12, (a) is the initial image of Pilots, (b), (c), and (d), respectively are the decrypted results in three different noised ciphers. These ciphers include those in which 40,000 random pixels, a block of pixels, and all the pixels, are added with random noise, respectively.

In Fig. 13, (a) is the initial image of Pyramid, (b) is the decrypted result of the cipher after Huffman compression and decompression, and (c) is the decrypted image of the cipher after DCT compression and decompression.

These figures show that, for lossless compressed ciphertext, the result after decompression and decryption is exactly the same as the plain image. Even for cropped, noised, and loss compressed ciphertext, the decrypted results are extremely close to the plain image.

8 Conclusions

In this paper, the plaintext related image hybrid encryption scheme is detailed discussed based on multiple theories and techniques such as Lagrange interpolation, generalized chaotic map, matrix nonlinear transformation, random matrix, and sequence rearrangement, etc. The encryption scheme is implemented in the mode of diffusion-scrambling-transformation. Using the operation of XOR and the first cipher matrix, the image diffusion is carried out. Combing Lagrange interpolation cipher with random matrices and sequence rearrangement, the image scrambling is performed. Utilizing the point operation and rounding operation, the compound transformation encryption is executed. Accordingly, the decryption process is fulfilled by the inverse operations in the contrary order. In our algorithm, the cryptographic mechanism related to plain image is fully complied through nonlinear transformations and operations. The results of the experimental simulation show that the proposed algorithm is feasible and effective. And the performance evaluation reveals that the scheme is quite secure to resist different attacks, such as statistical attack, chosen-plaintext attack, brute force attack, and differential attack, etc. Compared with other algorithms, the characteristics of our method are demonstrated in the paper. So it can be regarded as a candidate for practical image encryption.

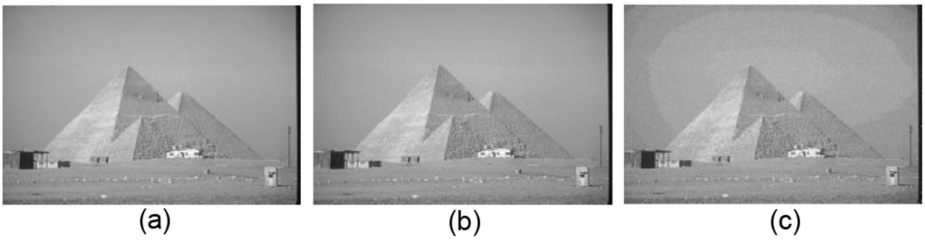


Fig. 13 Decrypted results of the compressed cipher for the experimental image Pyramid

Acknowledgments Without institutional funding, the authors carry out the research on image encryption solely due to personal interest. The authors express their sincere gratitude to those who have provided information and suggestions for this work.

Compliance with ethical standards

Ethical statements We solemnly declare that all authors are well aware of the academic ethics of the journal of Nonlinear Dynamics and are committed to unconditionally complying with these norms.

We certify that this manuscript is original and has not been published and will not be submitted elsewhere for publication while being considered by the journal of Multimedia Tools and Applications. And the study is not split up into several parts to increase the quantity of submissions and submitted to various journals or to one journal over time. No data have been fabricated or manipulated (including images) to support our conclusions. No data, text, or theories by others are presented as if they were our own.

The submission has been received explicitly from all co-authors. And authors whose names appear on the submission have contributed sufficiently to the scientific work and therefore share collective responsibility and accountability for the results.

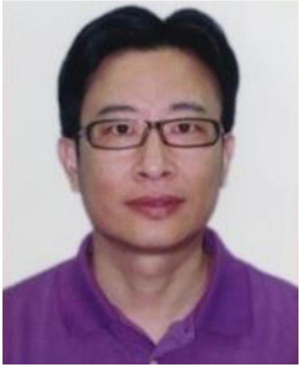
The authors declare that they have no conflict of interest. This article does not contain any studies with human participants or animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

References

1. 101_ObjectCategories, The Caltech 101 dataset[DB/OL] (2019) http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html#Download
2. Akif OZ (2012) Image encryption technique using Lagrange interpolation. Ibn Al-Haitham Journal for Pure and Applied Science 25(1):478–493
3. Amalarethinam DIG, Geetha JS (2015) Image encryption and decryption in public key cryptography based on MR. IEEE international conference on computing and communications technologies, Cairo, Egypt. Springer
4. Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213
5. Chen YC, Ye RS (2017) A novel image encryption algorithm based on improved standard mapping. Computer Science and Application 7(8):753–773
6. Cheng P, Yang H, Wei P, Zhang W (2015) A fast image encryption algorithm based on chaotic and lookup table. Nonlinear Dynamics 79(3):2121–2131
7. CIFAR-10 Matlab version, The CIFAR-10 dataset[DB/OL] (2019) http://www.cs.toronto.edu/~kriz/cifar.html
8. Contour Detection and Image Segmentation Resources[DB/OL] (2018) https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html
9. Fu C, Chen J, Zou H et al (2017) A selective compression -encryption of images based on SPIHT coding and Chirikov standard map. Signal Process 131:514–526
10. Ganesan K, Murali K (2014) Image encryption using eight dimensional chaotic cat map. The European Physical Journal Special Topics 223(8):1611–1622
11. Guo FM, Tu L (2015) Application of chaos theory in cryptography[M]. Beijing Institute of Technology Press, Beijing
12. Jin X, Yin S, Liu N, Li X, Zhao G, Ge S (2018) Color image encryption in non-RGB color spaces. Mult Tools Appl (MTAP) 77(12):15851–15873
13. Kanso A, Ghebleh M (2012) A novel image encryption algorithm based on a 3D chaotic map. Commun Nonlinear Sci Numer Simul 17:2943–2959
14. Kanso A, Ghebleh M (2015) An efficient and robust image encryption scheme for medical applications. Commun Nonlinear Sci Numer Simul 24(1–3):98–116
15. Liu CL (2017) Image processing with Matlab[M]. Tsinghua university press, Beijing
16. Liu Q, Li P, Zhang M, Sui Y, Yang H (2015) A novel image encryption algorithm based on chaos maps with Markov properties. Commun Nonlinear Sci Numer Simul 20(2):506–515
17. Liu W, Sun K, Zhu C (2016) A fast image encryption algorithm based on chaotic map. Opt Lasers Eng 84:26–36

18. Miller JE, Moursund DG, Duris CS (2011) Elementary theory and application of numerical analysis [M]. Dover Publications, New York
19. Norouzi B, Seyedzadeh SM, Mirzakuchaki S et al (2014) A novel image encryption based on hash function with only two-round diffusion process. Multimedia Systems 20(1):45–64
20. Prasad M, Sudha KL (2011) Chaos image encryption using pixel shuffling[M]. In: Wyld C et al (eds) CCSEA, CS &IT 02, pp 169–179
21. Ritwik MG, Krishna D, Sahoo A (2017) Cryptanalysis of image encryption using traditional encryption techniques. Image Vis Comput 23(5):89–97
22. Shamir A (1979) How to share secret [J]. Commun ACM 24(11):612–613
23. Sun XH (2013) Image encryption algorithms and practices with implentations in C#[M]. Science Press, Beijing
24. Wang X, Xu D (2015) A novel image encryption scheme using chaos and Langton's ant cellular automaton. Nonlinear Dynamics 79(4):2449–2456
25. Wei XP, Guo L, Zhang Q et al (2012) A novel gray image encryption algorithm based on DNA sequence operation and hyper-chaotic system[J]. J Syst Softw 85(2):290–299
26. Xu L, Guo X, Li Z et al (2017) A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. Opt Lasers Eng 91:41–52
27. Zhang Y (2014) Plaintext related image encryption scheme using chaotic map. TELKOMNIKA 12(1):635–643
28. Zhang Y (2014) A chaotic system based image encryption algorithm using plaintext-related confusion. TELKOMNIKA 12(11):7952–7962
29. Zhang Y (2016) Chaotic digital image cryptosystem[M]. Tsinghua university press, Beijing
30. Zhang W, Wong KW, Yu H et al (2013) An image encryption scheme using reverse 2-dimensinal chaotic map and dependant diffusion. Commun Nonlinear Sci Numer Simul 18:2066–2080
31. Zhang LY, Hu X, Liu Y et al (2014) A chaotic image encryption scheme owning temp-value feedback. Commun Nonlinear Sci Numer Simul 19(10):3653–3659
32. Zhen P, Zhao G, Min L, Jin X (2016) Chaos-based image encryption scheme combining DNA coding and entropy. Multimed Tools Appl (MTAP) 75(11):6303–6319
33. Zhu ZL, Zhang W, Wong KW et al (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation[J]. Inf Sci 181:1171–1186

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Xikun Liang , the corresponding author, received the B.S. degree from Huaibei Normal University in 1989, and received the Ph. D. degree from Hefei University of Technology in 2003, respectively. He has served as an associate professor in College of Information Science and Technology, and Hangzhou Institute of Service Engineering successively, Hangzhou Normal University since 2004. His main research interests include computer graphics and image processing, numerical algorithm, and intelligent information processing, in particular, image encryption and information hiding.

Xiao Tan , received the B.S. and M.S. degrees from FudanUniversity, in 2007 and 2010, and received the Ph. D. degree from City University of Hong Kong in 2013, respectively. He has served as a lecturer in Hangzhou Institute of Service Engineering, Hangzhou Normal University since 2014. His main research interests include applied cryptography and information security, in particular, cloud security, fair exchange of digital signatures, authenticated key agreement protocols.

Limin Tao , received the M.S. degree from Jiangnan University, in 2004, and received the Ph. D. degree from Zhejiang University of Technology in 2013, respectively. He has served as a lecturer in the Institute of Service Engineering, Hangzhou Normal University since 2004. His main research interests include service computing and cloud computing.

Affiliations

Xikun Liang[1,2] • **Xiao Tan**[1,2] • **Limin Tao**[1,2]

Xiao Tan
xiaotan_cs@163.com

Limin Tao
tlm5460@163.com

[1] College of Information Science and Engineering, Hangzhou Normal University, Hangzhou 311121, China

[2] Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou 311121, China