Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network

Xingyuan Wang[1, 2, *], Suo Gao[1]

(1 School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China)

(2 School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China)

Abstract: In this paper, a chaotic image encryption algorithm based on the matrix semi-tensor product (STP) with a compound secret key is designed. First, a new scrambling method is designed. The pixels of the initial plaintext image are randomly divided into four blocks. The pixels in each block are then subjected to different numbers of rounds of Arnold transformation, and the four blocks are combined to generate a scrambled image. Then, a compound secret key is designed. A set of pseudosecret keys is given and filtered through a synchronously updating Boolean network to generate the real secret key. This secret key is used as the initial value of the mixed linear-nonlinear coupled map lattice (MLNCML) system to generate a chaotic sequence. Finally, the STP operation is applied to the chaotic sequences and the scrambled image to generate an encrypted image. Compared with other encryption algorithms, the algorithm proposed in this paper is more secure and effective, and it is also suitable for color image encryption.

Keywords: Matrix semi-tensor product; Boolean network; Compound key; Spatiotemporal chaos; Image encryption

1 Introduction

Currently, with the rapid development of mobile Internet technology, images are gradually becoming important carriers of information in social communication [11, 26, 28, 47]. Images and image processing algorithms are widely encountered in various fields, such as computed tomography (CT) images in medicine, maps in the military field, and facial recognition for various technological applications. However, when information is transmitted over a network, it is sometimes desired that no one except the recipient should be able to see the transmitted information [1, 24]. Therefore, solving the problem of how to protect information security during the transmission process has become an important challenge [31, 33, 42]. According to chaos theory, chaotic systems have characteristics such as pseudorandomness, initial value sensitivity, parameter sensitivity, ergodicity, and unpredictability [2, 30, 48]. Therefore, concepts related to chaos are widely used in the field of image security [22, 32].

With the development of increasingly chaotic systems and the cross-application of chaotic systems with other disciplines, many new chaotic image encryption algorithms have been presented. Guan used frequency-domain DNA encoding to propose a new color image encryption algorithm [9]. Gong proposed a new encryption

*Corresponding authors. E-mail addresses: xywang@dlmu.edu.cn (X. Wang), 1418159118@qq.com (S. Gao)

algorithm combined with diffractive imaging [8]. Wang applied a neural network for image encryption [29]. Zhou proposed a series of quantum image encryptions combined with quantum theory [46, 49].

In this paper, a new image encryption scheme based on the semi-tensor product (STP) is proposed. The STP was proposed by Cheng [6,7]. In recent years, the STP has been widely used in Boolean network control and synchronization, graph coloring, game theory and other areas [17, 20, 21, 44]. However, it has not been applied in the field of image encryption [19]. The STP has reversible properties and thus is consistent with the characteristics of symmetric encryption [25, 34, 45]; hence, it is very suitable for image encryption. In this paper, we use the STP in image diffusion. The experimental results show that the STP can be effectively used for image encryption.

In common encryption algorithms, the secret key design is simple and cannot resist certain plaintext attacks. For example, the algorithm proposed by Zhang Q [39] was cracked by Zhang Y [43], and Zhang [41] cracked the algorithm proposed by Zhu [50]. These algorithms were cracked by means of plaintext attacks. To address this shortcoming, many methods of generating secret keys have been proposed. Kaur [15] proposed multimodal biometric keys. Khan [16] proposed a DNA key. Wang [27] proposed a quantum key and used it for quantum image encryption. In this paper, based on Boolean network theory, an encryption algorithm with a compound secret key is proposed. The secret key is generated in a Boolean network and related to the original plaintext image, allowing it to resist chosen-plaintext attacks. The complex structure of the Boolean network also makes it more difficult to crack the algorithm.

A Boolean network is a kind of genetic network. The Boolean network model is a discrete system that was originally proposed by Kauffman [14]. In a Boolean network, each node represents a genetic state [13, 40]. Each gene has two possible states, “on” and “off”, which are represented by the binary values 1 and 0 and can be simply understood as indicating whether the corresponding gene is expressed or not [18, 35, 36]. Depending on the update rules applied, Boolean networks can be divided into synchronously updating Boolean networks and asynchronously updating Boolean networks. In a synchronously updating Boolean network, every node in the network changes with each update [38]. In an asynchronously updating Boolean network, there are two possible states in the network, updating and not updating, meaning that each node may or may not be updated at each time [37]. Because a Boolean network has a symmetric form, its expression is reversible, and a Boolean network also has a complex structure, making it very suitable for the generation of secret keys [5].

The experimental results presented in this paper show that compared with other algorithms, the algorithm proposed here has a good encryption effect, and it can resist all kinds of common attacks.

2 Relevant knowledge

This section introduces background knowledge on Boolean networks and presents an example of a synchronously updating Boolean network as well as an introduction to the mixed linear-nonlinear coupled map lattice (MLNCML) system.

2.1 Boolean networks and the semi-tensor product (STP)

This paper focuses on the use of a synchronously updating Boolean network to generate secret keys. The equations for a synchronously updating Boolean network are shown in Eq. (1) [38]:

$$\begin{cases} y_1(t+1) = f_1(y_1(t), y_2(t), \cdots, y_n(t)) \\ y_2(t+1) = f_2(y_1(t), y_2(t), \cdots, y_n(t)) \\ \vdots \\ y_n(t+1) = f_n(y_1(t), y_2(t), \cdots, y_n(t)) \end{cases}. \tag{1}$$

In Eq. (1), $y_i(t)$ represents the state of gene node i at time t. The functions f_i, i=1, 2, 3, ..., n, are logical operation functions. They are also update calculation rules for the state of each gene node. The logical operators used in these operation functions include $\neg X$, $X \wedge Y$, $X \vee Y$, $X \oplus Y$, $X \leftrightarrow Y$, and $X \to Y$.

Because a Boolean network is limited by its logical expression, it is difficult to analyze its dynamic behavior with traditional mathematical tools. To overcome this challenge, Cheng proposed the STP, which solves this problem very well [7]. The STP theory for matrices can be described as follows.

Definition 1 [6] Suppose that a is a row vector of kl dimensions and that b is an l-dimensional column vector. We decompose a into l blocks of the same size, denoted by $a^1, a^2, a^3, \cdots, a^l$, where the size of a^i is $1 \times k$. The STP is calculated as follows:

$$\begin{cases} a \ltimes b = \sum_{i=1}^{l} a^i b_i \in \mathrm{R}^k \\ b^{\mathrm{T}} \ltimes a^{\mathrm{T}} = \sum_{i=1}^{l} b_i (a^i)^{\mathrm{T}} \in \mathrm{R}^k \end{cases}. \tag{2}$$

In Eq. (1), "$\ltimes$" is the symbol representing the STP.

Definition 2 [6] Suppose that $P \in M_{r \times l}$ and $Q \in M_{s \times t}$. When l is divisible by s and $kl = s$, we say that $P \prec_k Q$; when s is divisible by l, we say that $P \succ_k Q$. The STP operation between P and Q may be defined as $W = P \ltimes Q$, where the matrix W consists of $r \times t$ blocks, each defined as follows:

$$W^{ij} = P^i \ltimes Q_j \ (i = 1, 2, \cdots, r, \ j = 1, 2, \cdots, t). \tag{3}$$

In Eq. (3), P^i represents the i-th row of the matrix P, and Q_j represents the j-th column of the matrix Q.

Theorem 1 [6] Given $X \in M_{m \times np}$ and $Y \in M_{p \times q}$, we define

$$X \ltimes Y = X(Y \otimes I_n). \tag{4}$$

In Eq. (4), "$\otimes$" is the symbol representing the matrix tensor product, and I_n is a unit matrix of order n.

Theorem 2 Suppose that $Z = X_{np \times np} \ltimes Y_{p \times p}$. If $|Y \otimes I_n| \neq 0$, then

$$X = Z(Y \otimes I_n)^{-1}. \tag{5}$$

Proof: Because $Z = X \ltimes Y = X(Y \otimes I_n)$ and $\left|Y \otimes I_n\right| \neq 0$, $(Y \otimes I_n)^{-1}$ exists.

$Z(Y \otimes I_n)^{-1} = X(Y \otimes I_n)(Y \otimes I_n)^{-1}$, and $X = Z(Y \otimes I_n)^{-1}$.

Theorem 3 [10] $\det(Y \otimes I_n) = (\det Y)^p (\det I_n)^n = (\det Y)^p$, where Y has dimensions of $p \times p$.

Example 1 Let $X=\begin{bmatrix}1 & 1 & 0 & 2\\ 3 & 1 & 0 & 4\\ 4 & 2 & 2 & 1\end{bmatrix}$ and $Y=\begin{bmatrix}1 & 3\\ 2 & 1\end{bmatrix}$.

By applying Eq. (2), we can obtain the following:

$$X \ltimes Y = \begin{bmatrix}(1 \quad 1)\times 1+(0 \quad 2)\times 2 & (1 \quad 1)\times 3+(0 \quad 2)\times 1\\ (3 \quad 1)\times 1+(0 \quad 4)\times 2 & (3 \quad 1)\times 3+(0 \quad 4)\times 1\\ (4 \quad 2)\times 1+(2 \quad 1)\times 2 & (4 \quad 2)\times 3+(2 \quad 1)\times 1\end{bmatrix}$$

$$= \begin{bmatrix}(1 \quad 5) & (3 \quad 5)\\ (3 \quad 9) & (9 \quad 7)\\ (8 \quad 4) & (14 \quad 7)\end{bmatrix} = \begin{bmatrix}1 & 5 & 3 & 5\\ 3 & 9 & 9 & 7\\ 8 & 4 & 14 & 7\end{bmatrix}.$$

Example 2 Let $Z=\begin{bmatrix}1 & 5 & 3 & 5\\ 3 & 9 & 9 & 7\\ 8 & 4 & 14 & 7\end{bmatrix}$ and $Y=\begin{bmatrix}1 & 3\\ 2 & 1\end{bmatrix}$.

We can obtain X from Eq. (5), $X = Z(Y \otimes I_2)^{-1}$, as follows:

$$X = \begin{bmatrix}1 & 5 & 3 & 5\\ 3 & 9 & 9 & 7\\ 8 & 4 & 14 & 7\end{bmatrix} \times \left(\begin{bmatrix}1 & 3\\ 2 & 1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right)^{-1}$$

$$= \begin{bmatrix}1 & 5 & 3 & 5\\ 3 & 9 & 9 & 7\\ 8 & 4 & 14 & 7\end{bmatrix} \times \begin{bmatrix}1 & 0 & 3 & 0\\ 0 & 1 & 0 & 3\\ 2 & 0 & 1 & 0\\ 0 & 2 & 0 & 1\end{bmatrix}^{-1}$$

$$= \begin{bmatrix}1 & 5 & 3 & 5\\ 3 & 9 & 9 & 7\\ 8 & 4 & 14 & 7\end{bmatrix} \times \begin{bmatrix}-0.2 & 0 & 0.6 & 0\\ 0 & -0.2 & 0 & 0.6\\ 0.4 & 0 & -0.2 & 0\\ 0 & 0.4 & 0 & -0.2\end{bmatrix}$$

$$= \begin{bmatrix}1 & 1 & 0 & 2\\ 3 & 1 & 0 & 4\\ 4 & 2 & 2 & 1\end{bmatrix}.$$

Other definitions related to the STP are as follows:

(1) δ_n^r represents the r-th column of the n-dimensional unit matrix I_n. In a binary Boolean network, $T = 1 \sim \delta_2^1 = [1\ 0]^{\mathrm{T}}$ and $F = 0 \sim \delta_2^2 = [0\ 1]^{\mathrm{T}}$.

(2) A structure matrix is defined in the following form [38]:

$$\begin{aligned} M_{\neg} &:= M_n = \delta_2[2,1] \\ M_{\wedge} &:= M_c = \delta_2[1,2,2,2] \\ M_{\vee} &:= M_d = \delta_2[1,1,1,2] \\ M_{\rightarrow} &:= M_i = \delta_2[1,2,1,1] \\ M_{\oplus} &:= M_x = \delta_2[2,1,1,2] \\ M_{\leftrightarrow} &:= M_e = \delta_2[1,2,2,1] \end{aligned}. \tag{6}$$

In Eq. (6), if the logic matrix is $\Lambda \in [\delta_n^{i_1}, \delta_n^{i_2}, \cdots, \delta_n^{i_m}]$, then we can simplify $\Lambda \in [\delta_n^{i_1}, \delta_n^{i_2}, \cdots, \delta_n^{i_m}]$ as $\Lambda \in \delta_n[i_1, i_2, \cdots, i_m]$.

For convenience of operation, a subtraction parameter matrix is defined. It has the following properties:

$$x^2 = M_r x .$$

(3) For two column vectors, $X \in R^m$ and $Y \in R^n$, an exchange matrix $W_{[m,n]}$ is defined; then, we can obtain

$$W_{[m,n]} \ltimes X \ltimes Y = Y \ltimes X. \tag{7}$$

In Eq. (7), $W_{[m,n]}$ is an $mn \times mn$-dimensional matrix.

The column indices are $(11, 12, \cdots 1n, \cdots, m1, m2, \cdots mn)$. The row indices are $(11, 21, \cdots, m1, \cdots, 1n, 2n, \cdots, mn)$. Accordingly, the value of each variable in the matrix is

$$w_{(I,J),(i,j)} = \begin{cases} 1, & I = i, J = j \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

When $m = n$, we can obtain $W_{[m]} = W_{[m,m]}$ [38].

Example 3 When $m = 2$ and $n = 2$, the exchange matrix $W_{[2,2]}$ is defined as

$$W_{[2,2]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

When $m = 2$ and $n = 3$, the exchange matrix $W_{[2,3]}$ is defined as

$$W_{[2,3]} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Based on the above foundation, the STP theory is established as follows.

Lemma 1 [6] The logical function $M(P_1, P_2, \cdots, P_r)$ with elements $P_1, P_2, \cdots, P_r$ can be expressed linearly as follows:

$$M(P_1, P_2, ..., P_r) = L_M P_1 P_2 \cdots P_r . \tag{9}$$

In Eq. (9), the matrix L_M is the structure matrix of the logic function M.

Unless otherwise noted, all multiplication operations in this paper are STP operations, and the "⋉" symbol is omitted.

Example 4 Consider the following synchronously updating logical Boolean network:

$$\begin{cases} x_1(t+1) = x_2(t) \wedge x_3(t) \\ x_2(t+1) = x_3(t) \\ x_3(t+1) = x_1(t) \wedge (x_2(t) \leftrightarrow x_3(t)) \end{cases}. \tag{10}$$

In accordance with logical rules, Eq. (10) can be converted into the form of Eq. (11):

$$\begin{cases} x_1(t+1) = M_c x_2(t) x_3(t) \\ x_2(t+1) = x_3(t) \\ x_3(t+1) = M_c x_1(t) M_e x_2(t) x_3(t) \end{cases}. \tag{11}$$

Suppose that $x(t) = x_1(t) \ltimes x_2(t) \ltimes x_3(t)$; then, Eq. (12) can be obtained in accordance with Eq. (4), Eq. (5), and Lemma 1.

$$\begin{aligned} x(t+1) &= x_1(t+1)x_2(t+1)x_3(t+1) \\ &= M_c x_2(t)x_3(t)x_3(t)M_c x_1(t)M_e x_2(t)x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)x_2(t)x_3(t)x_3(t)x_1(t)x_2(t)x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)x_2(t)M_r x_3(t)x_1(t)x_2(t)x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)x_2(t)x_3(t)x_1(t)x_2(t)x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}x_1(t)x_2(t)x_3(t)x_2(t)x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}x_1(t)x_2(t)W_{[2]}x_2(t)x_3(t)x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}(I_4 \otimes W_{[2]})x_1(t)M_r x_2(t)M_r x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)x_1(t)x_2(t)M_r x_3(t) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r)x_1(t)x_2(t)x_3(t) \end{aligned}. \tag{12}$$

If the initial node state $x(0)$ is given, the node states at any time may be obtained using Eq. (12).

2.2 Mixed linear-nonlinear coupled map lattice (MLNCML) system

The MLNCML system was proposed by Zhang [42]. It is an improved version of the coupled map lattice (CML) system obtained by adding nonlinear coupling to the CML system. It has a smaller periodic window than the CML system; therefore, it is more suitable for image encryption. The expression for the MLNCML system is as follows:

$$y_{n+1}(i) = (1-\theta)f[y_n(i)] + (1-\eta)\frac{\theta}{2}\{f[y_n(i+1)] + f[y_n(i-1)]\} + \eta\frac{\theta}{2}\{f[y_n(j)] + f[y_n(k)]\}. \tag{13}$$

In Eq. (13), n ($n = 1, 2, 3...$) is the time index, and $\theta\,(0 \le \theta \le 1)$ and $\eta\,(0 \le \eta \le 1)$ are coupling parameters. $f(y) = \mu y(1-y)$ ($3.57 < \mu \le 4$) is the logistic map. $i-1$ and $i+1$ are the lattice points adjacent to i. i, j, and k represent different lattice points in space, and the relationship between them is determined by an Arnold mapping, which is expressed as

$$\begin{bmatrix} j \\ k \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} i \\ i \end{bmatrix} (\mathrm{mod}\, L). \tag{14}$$

In Eq. (14), p and q are the parameters of the Arnold mapping, and L is the number of lattice points in the

MLNCML system.

3 The proposed image encryption algorithm

The encryption method proposed in this paper is a process consisting of scrambling followed by diffusion. In this section, the generation of compound secret keys and the processes of scrambling and diffusion are introduced. The encryption algorithm proposed in this paper is designed for an image size of $M \times M$. If the original plaintext matrix does not meet this requirement, then we adopt the method of zero padding to expand it to a square matrix. The encryption process is described below.

3.1 Generation of the compound secret key

In this paper, a Boolean network model is used to generate the secret key.

The pseudosecret key is defined by the encryptor, and no separate secret key generator is required to produce it. The pseudosecret key is a simple vector. The Boolean network that serves as the secret key generator is stored by both the sender and the receiver. For the transmission process, the sender needs to send only the pseudosecret key, and the receiver then obtains the true secret key through the secret key generator. The time required to generate the pseudosecret key and the true secret key is not included in the encryption time. The length of the pseudosecret key is different from that of the true secret key in order to ensure that the pseudosecret key is different from the true secret key.

We provide a set of pseudosecret keys, denoted by k_1, k_2, and k_3, where the value of k_i is either 0 or 1. We define k_i as follows:

$$x_i(0) = \begin{cases} \delta_2^1, & k_i = 1 \\ \delta_2^2, & k_i = 0 \end{cases}.$$

$x_i(0)$ is an initial value introduced into Eq. (11); then, $x_i(1)$ is obtained by updating. Subsequently, $x_i(1)$ is taken as an initial value introduced into Eq. (12) to be updated, and thus, $x_i(2)$ is obtained.

Definition 3 A function $\mathrm{g}(x_1, x_2, \cdots, x_n)$ is defined to concatenate the vectors $x_1, x_2, \cdots, x_n$ from beginning to end.

Example 5 Suppose that $x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$; then, through Definition 3, we obtain $\mathrm{g}(x_1, x_2) = [1,0,0,1]^{\mathrm{T}}$.

Definition 4 A function $h(x)$ is defined to convert a vector x into a decimal value.

Example 6 Suppose that $x = [1,0,0,1]^{\mathrm{T}}$; then, through Definition 4, we obtain $h(x) = 0.1001$.

The initial values of the MLNCML system are given below:

$$y_1 = h(g(x_1(0), x_2(0), x_3(0), x_1(1), x_2(1), x_3(1))),$$
$$y_2 = h(g(x_1(0), x_2(0), x_3(0), x_1(2), x_2(2), x_3(2))),$$
$$y_3 = h(g(x_1(1), x_2(1), x_3(1), x_1(2), x_2(2), x_3(2))),$$
$$y_4 = h(g(x_1(1), x_2(1), x_3(1), x_1(0), x_2(0), x_3(0))),$$

$$y_5 = h(g(x_1(2), x_2(2), x_3(2), x_1(0), x_2(0), x_3(0))) .$$

$x_1(0),\ x_2(0),\ \text{and}\ x_3(0)$ are generated from $k_1,\ k_2,\ \text{and}\ k_3$; then, $x_1(1),\ x_2(1),\ \text{and}\ x_3(1)$ as well as $x_1(2),\ x_2(2),\ \text{and}\ x_3(2)$ are generated by this Boolean network:

$$\begin{cases} x_1(t+1) = x_2(t) \wedge x_3(t) \\ x_2(t+1) = x_3(t) \\ x_3(t+1) = x_1(t) \wedge (x_2(t) \leftrightarrow x_3(t)) \end{cases} .$$

According to Eq. (12),

$$\begin{aligned} x(1) &= x_1(1)x_2(1)x_3(1) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r)x_1(0)x_2(0)x_3(0) \\ x(2) &= x_1(2)x_2(2)x_3(2) \\ &= M_c(I_8 \otimes M_c)(I_{16} \otimes M_e)(I_2 \otimes M_r)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r)x_1(1)x_2(1)x_3(1) \end{aligned} .$$

The Boolean network that serves as the secret key generation tool is stored by both the sender and the receiver. During transmission, the sender needs to transmit only a set of pseudosecret keys. The receiver then generates the real secret key using the secret key generator and decrypts the image. The space occupied by this set of pseudosecret keys is smaller than the space of the true secret key, thereby saving transmission capacity and improving transmission efficiency. Even if an attacker intercepts the pseudosecret keys, he or she cannot crack the secret key without the secret key generator. The attacker cannot crack the secret key through a brute-force attack because the attacker cannot determine the length of the true secret key from the pseudosecret keys. Therefore, the proposed method of secret key generation using a Boolean network makes the encryption system more secure.

3.2 Combination scrambling

In this paper, a method of combination scrambling is proposed. First, the pixels of the original image are divided into four blocks in accordance with a chaotic sequence; then, each block is scrambled through different numbers of rounds of Arnold mapping, and finally, the scrambled image blocks are combined.

Based on the chaotic initial value y_1 generated as described in Section 3.1, the chaotic sequence Y_1 is generated using Eq. (13). Then, Y_1 is sorted as follows: $[b, R] = sort(Y_1)$, where b is the sorted vector and R is each item in b that corresponds to the index of items in Y_1. In this way, we obtain an unduplicated array R with dimensions of $1 \times MM$. Any plaintext image of the same size will produce the same array R. Therefore, we can generate the array R in advance, meaning that the time needed to generate the array is not included in the total encryption time, thus improving the encryption efficiency.

The steps of the scrambling process are described as follows.

Step 1: R is divided into four parts, defined as follows:

$$\begin{aligned} R_1 &= R(1 : MM/4) , \\ R_2 &= R(1 + MM/4 : MM/2) , \\ R_3 &= R(1 + MM/2 : 3MM/4) , \\ R_4 &= R(1 + 3MM/4 : MM) . \end{aligned}$$

The pixels of the plaintext image P are numbered in line-first order as $1\sim MM$. First, the pixel numbered R_1 in P is found and used to generate the matrix P_1 in row-priority order, where the size of P_1 is $M/2\times M/2$. This process is then repeated to generate the matrices $P_2, P_3,$ and P_4.

Step 2: The matrices $P_1, P_2, P_3,$ and P_4 generated in Step 1 are individually scrambled via different numbers of rounds of Arnold mapping and are finally recombined to form the scrambled image S.

The Arnold mapping process is expressed as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} (\mathrm{mod}\, L).$$

This formula can be rewritten as

$$\begin{cases} x = (((i-1)+p\times(j-1)) \bmod L)+1 \\ y = ((q\times(i-1)+(pq+1)\times(j-1)) \bmod L)+1 \end{cases}.$$

In Figs. 1~3, an example is given to illustrate the method of combination scrambling proposed in this paper. Each number in these figures represents the position number of the corresponding pixel, not the pixel value. In this example, we assume that the parameters of the Arnold mapping are $p=3$ and $q=5$. The numbers of rounds of scrambling are $l_1=2$, $l_2=3$, $l_3=4$ and $l_4=5$.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Fig. 1 Pixel position numbers in the original image

Figs. 2(a)~(d) shows the block information obtained from Fig. 1 based on a chaotic sequence. Figs. 2(e)~(h) show the results of different numbers of rounds of Arnold scrambling.

30	11	16	57
47	51	39	1
12	43	63	37
61	45	31	14

(a) P_1

55	28	19	9
21	17	49	41
33	3	5	35
26	23	53	59

(b) P_2

7	58	6	52
24	25	48	34
22	8	40	4
27	2	60	54

(c) P_3

32	13	36	18
20	64	44	62
42	38	56	50
46	15	10	29

(d) P_4

30	1	63	45
57	39	43	61
16	51	12	14
11	47	37	31

(e) P_1 after scrambling

55	9	19	28
26	59	53	23
33	35	5	3
21	41	49	17

(f) P_2 after scrambling

7	27	22	24
2	8	25	58
40	48	6	60
34	52	54	4

(g) P_3 after scrambling

32	15	56	62
13	10	50	20
36	29	42	64
18	46	38	44

(h) P_4 after scrambling

Fig. 2 Block operation and different numbers of rounds of Arnold scrambling

Fig. 3 shows the final scrambled image reassembled from Figs. 2(e)~(h). After the block scrambling process, the STP diffusion operation is carried out.

30	1	63	45	55	9	19	28
57	39	43	61	26	59	53	23
16	51	12	14	33	35	5	3
11	47	37	31	21	41	49	17
7	27	22	24	32	15	56	62
2	8	25	58	13	10	50	20
40	48	6	60	36	29	42	64
34	52	54	4	18	46	38	44

Fig. 3 Figs. 2(e)~(h) combined to form the scrambled image

The advantages of combination scrambling are as follows. First, because the image is divided into four blocks, the computer can process all four blocks simultaneously. Thus, compared with traditional scrambling, the efficiency of combination scrambling is higher. Second, because a different number of rounds of Arnold mapping is carried out for each block, even if an attacker breaks the Arnold mapping, the whole scrambled image cannot be obtained, thereby increasing the security of the scrambling process.

3.3 STP diffusion

STP diffusion is used in the proposed image encryption process. STP diffusion is similar to a chemical reaction. The scrambled image acts as one reactant. Another reactant is formed by the MLNCML system, whose size is different from that of the original image. The matrix STP operation is performed on these two reactants to achieve the effect of diffusion.

Usually, an attacker searches a database of known encryption algorithms to crack a particular algorithm by referring to known algorithms. However, STP diffusion is an unknown algorithm, so an attacker will not be able to find a matching algorithm to crack it. This improves the security of the algorithm. In addition, compared with other algorithms, the algorithm proposed in this paper is inherently more secure.

Chaotic initial values y_2, y_3, y_4, and y_5 generated as described in Section 3.1 are input into Eq. (13) to generate chaotic sequences Y_2, Y_3, Y_4, and Y_5.

Definition 5 A function $W(x)$ is defined that outputs the last four digits of a positive integer x.

Example 7 Suppose that $x = 987654$; then, through Definition 5, we obtain $W(x) = 7654$.

For a plaintext image $P_{M\times M}$, we define *n* as follows:

$$n = W((\sum_{i=1}^{M}\sum_{j=1}^{M} P(i,j))^2). \tag{15}$$

Then, we define A_i as follows:

$$\begin{cases} A_1 = \text{reshape}(Y_2(n:\frac{M}{16}\times\frac{M}{16}+n-1),\ \frac{M}{16},\ \frac{M}{16}) \\ A_2 = \text{reshape}(Y_3(n:\frac{M}{8}\times\frac{M}{8}+n-1),\ \frac{M}{8},\ \frac{M}{8}) \\ A_3 = \text{reshape}(Y_4(n:\frac{M}{4}\times\frac{M}{4}+n-1),\ \frac{M}{4},\ \frac{M}{4}) \\ A_4 = \text{reshape}(Y_5(n:\frac{M}{2}\times\frac{M}{2}+n-1),\ \frac{M}{2},\ \frac{M}{2}) \end{cases}. \tag{16}$$

In Eq. (16), from the *n*th element of each chaotic sequence, we extract a chaotic sequence of fixed length. The notation $A = \text{reshape}(P,m,n)$ represents the transformation of P into a matrix A with dimensions of $m\times n$.

A_i has dimensions of $\frac{M}{2^{5-i}}\times\frac{M}{2^{5-i}}$ ($i = 1,2,3,4$). If $|A_i| = 0$, we use the following formula to continue to look for an A_i such that $|A_i| \neq 0$ (according to Theorem 3, if $|A_i| \neq 0$, then $|A_i \otimes I_{2^{5-i}}| \neq 0$):

$$A_i = \text{reshape}(Y_{i+1}(n+100:\frac{M}{2^{5-i}}\times\frac{M}{2^{5-i}}+n+99),\ \frac{M}{2^{5-i}},\ \frac{M}{2^{5-i}}).$$

In Example 6, the initial values of the MLNCML system generated by the Boolean network can avoid periodicity of the generated chaos. The resulting sequences are all chaotic. The row vectors of the chaotic sequence matrix are linearly independent, so A_i is reversible. Generally, only one A_i can satisfy $|A_i| \neq 0$.

The STP diffusion process is carried out using the following formulas:

$$C = S \ltimes A_1,\ \text{if mod}(n,4) = 0, \tag{17}$$

$$C = S \ltimes A_2, \text{ if } \mathrm{mod}(n,4) = 1, \tag{18}$$

$$C = S \ltimes A_3, \text{ if } \mathrm{mod}(n,4) = 2, \tag{19}$$

$$C = S \ltimes A_4, \text{ if } \mathrm{mod}(n,4) = 3, \tag{20}$$

$$D = \mathrm{mod}(\mathrm{floor}(C), 256). \tag{21}$$

The secret key is calculated as follows:

$$\begin{aligned} C_1 &= \mathrm{floor}(\mathrm{C}/256) \\ C_2 &= C - \mathrm{floor}(C) \end{aligned}. \tag{22}$$

In Eq. (21) and Eq. (22), mod(x) is the modulus function; floor(x) is the floor function, which returns the next smaller integer than the argument; and C_1 and C_2 are the secret keys to be decrypted. We can then obtain C from C_1 and C_2:

$$C = C_1 \times 256 + D + C_2. \tag{23}$$

In this way, the ciphertext image D is generated. The flow of the encryption process is shown in Fig. 4.

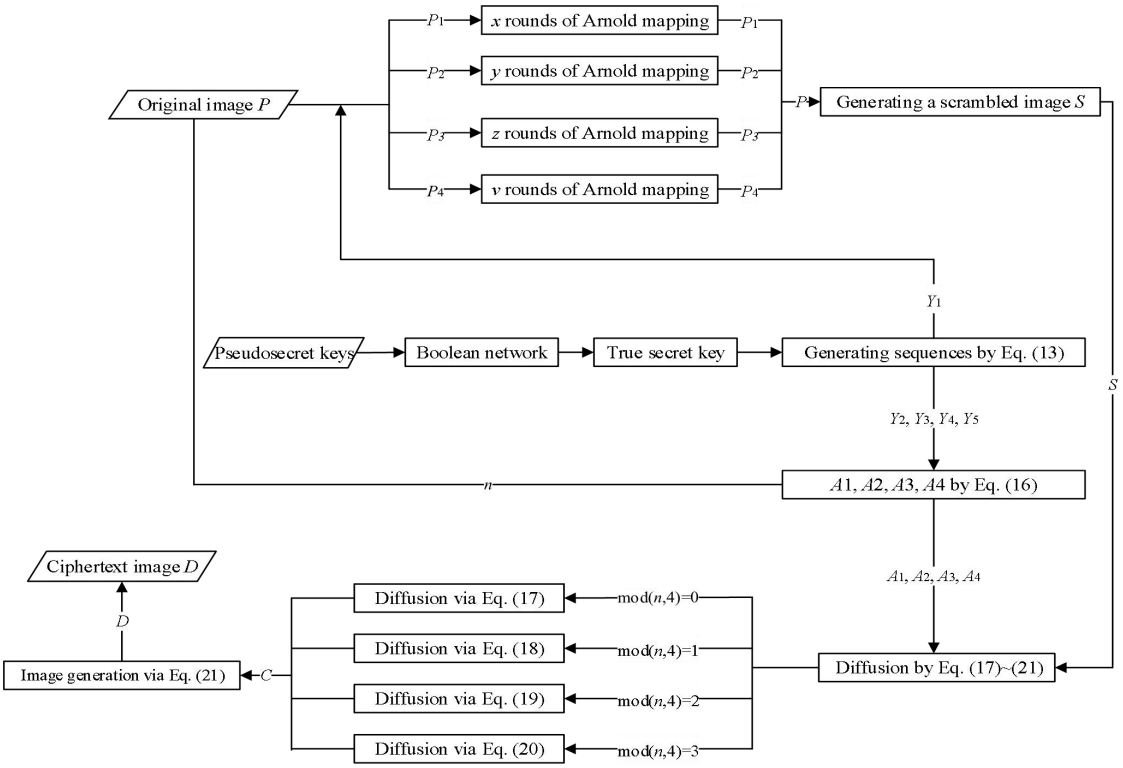


Fig. 4 Encryption flow chart

3.4 Decryption process

The encryption algorithm proposed in this paper is symmetric, and each step is reversible; therefore, the decryption process is the inverse of the encryption process, as shown in Fig. 5.

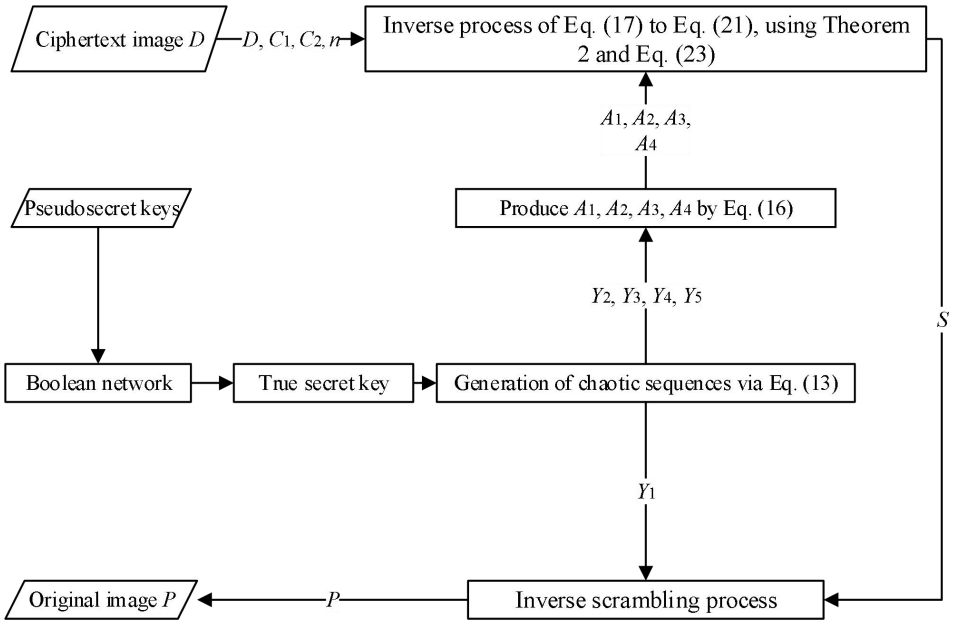


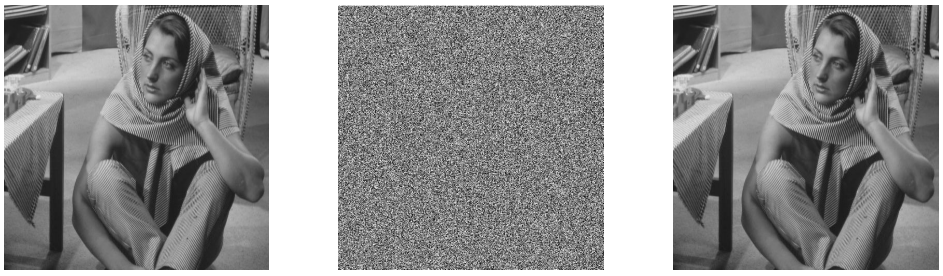
Fig. 5 Decryption flow chart

During decryption, n is a 4-bit integer produced by Eq. (15). C_1 is the integer part of the secret key, and C_2 is the decimal part of the secret key; therefore, C_1 and C_2 can be merged into a single secret key, $CK = C_1 + C_2$, which can then be transmitted as an image. Notably, our algorithm is primarily designed for application to small images, which take up little storage space, and current technology allows a secret key of the same size to be carried as ciphertext for transmission. In practice, existing compression technology can be used to compress the secret key in order to enhance the transmission efficiency.

4 Performance analysis

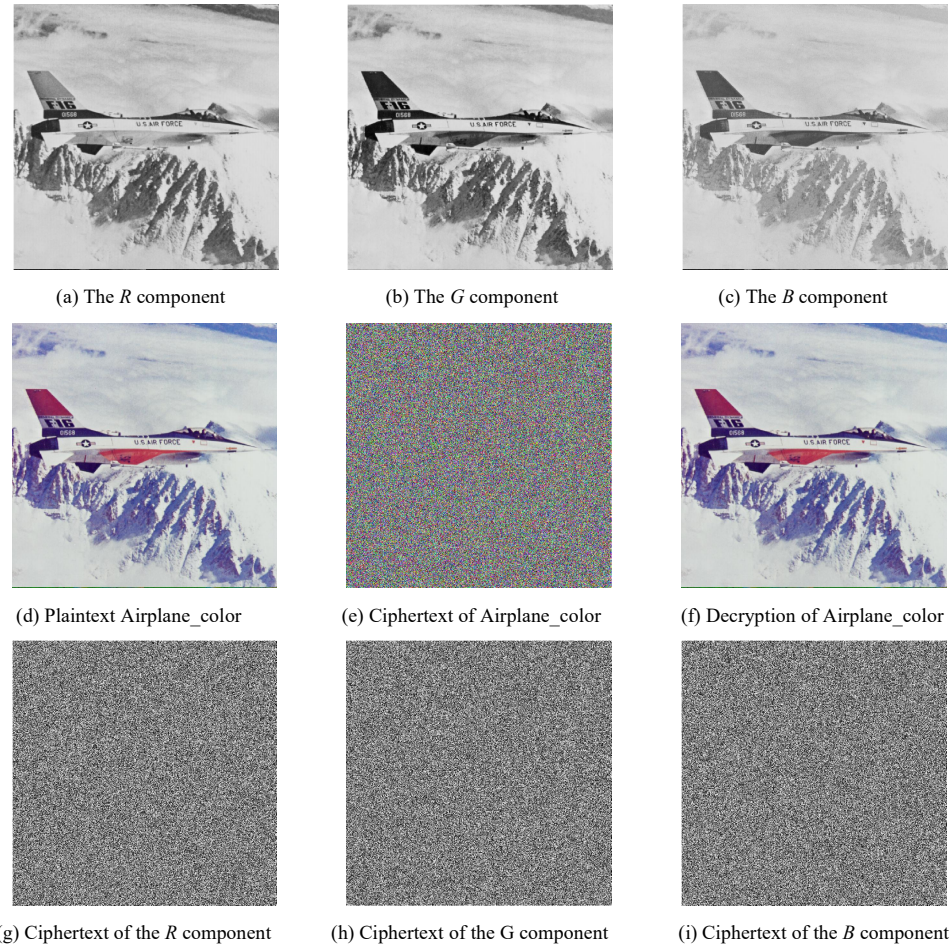
4.1 Simulation experiment

In this section, the encryption and decryption results for Barbara_gray and Airplane_color are shown in Fig. 6 and Fig. 7. Color images can be divided into three channels, red (R), green (G) and blue (B). In the encryption algorithm proposed in this paper, we encrypt each of these three channels separately and then synthesize them to obtain the ciphertext image. The size of each encrypted image is 512×512.



(a) Plaintext of Barbara_gray (b) Ciphertext of Barbara_gray (c) Decryption of Barbara_gray

Fig. 6 Encryption and decryption of Barbara_gray



(a) The *R* component (b) The *G* component (c) The *B* component

(d) Plaintext Airplane_color (e) Ciphertext of Airplane_color (f) Decryption of Airplane_color

(g) Ciphertext of the *R* component (h) Ciphertext of the G component (i) Ciphertext of the *B* component

Fig. 7 Encryption and decryption of Airplane_color

4.2 Encryption and decryption of Lena_gray

Step 1: Generation of chaotic sequences

First, we specify a set of pseudosecret keys, $k_1 = 1,\ k_2 = 0,$ and $k_3 = 0$, and use the k_i to generate the initial values of the chaotic system through a Boolean network. The other parameters of the MLNCML system are given as $\theta = 0.453,\ \eta = 0.77,\ \mu = 3.99,\ p = 12,$ and $q = 7$. Finally, chaotic sequences $Y_1,\ Y_2,\ Y_3,\ Y_4,$ and Y_5 are generated via Eq. (13).

Step 2: Scrambling

The chaotic sequence Y_1 is used to divide the pixels of the plaintext image into blocks, and the Arnold mapping of each block is carried out to form the scrambled image, as shown in Fig. 8.

(a) Lena_gray (f) Scrambling of Lena_gray composed of (b) ~ (e)

(b) P_1 subblock of Lena_gray (c) P_2 subblock of Lena_gray (d) P_3 subblock of Lena_gray (e) P_4 subblock of Lena_gray

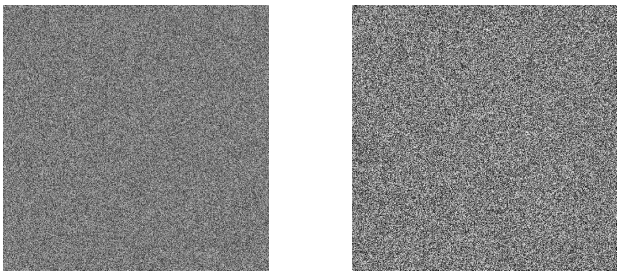
Fig. 8 Combination scrambling

Step 3: Diffusion

$n_{\text{Lena}} = 7721$ is calculated from the pixel values of the plaintext image. Because mod $(n, 4) = 1$, we use

$$C = S \ltimes A_2, \text{ if } \operatorname{mod}(n,4) = 1,$$
$$D = \operatorname{mod}(\operatorname{floor}(C), 256)$$

to generate the ciphertext D, where S is the scrambled image. Fig. 9 shows the results of the process from scrambling to diffusion.



(a) Scrambling of Lena_gray (b) STP diffusion of Lena_gray

Fig. 9 STP diffusion

Step 4: Decryption

The secret key C is generated during encryption. From the formulas

$$C_1 = \operatorname{floor}(\mathrm{C}/256)$$
$$C_2 = C - \operatorname{floor}(C)$$

and the ciphertext D, we calculate

$$C = C_1 \times 256 + D + C_2,$$

and then,

$$S = C(A_2 \otimes I_8)^{-1}.$$

The original image P can be obtained through inverse scrambling.

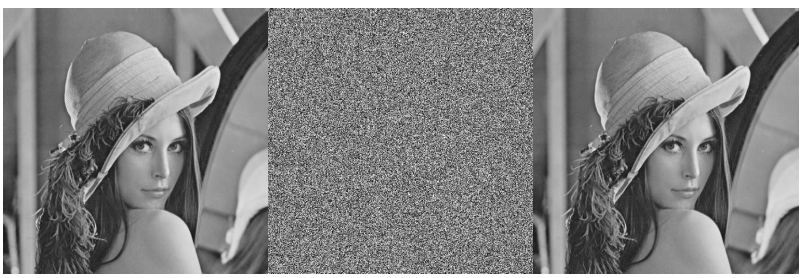


Fig. 10 Encryption and decryption of Lena_gray

4.3 Secret key space analysis

The secret keys considered in this article are

$$key = (k_1, k_2, k_3, \theta, \eta, \mu, p, q, C_1, C_2, n, l_1, l_2, l_3, l_4) ,$$

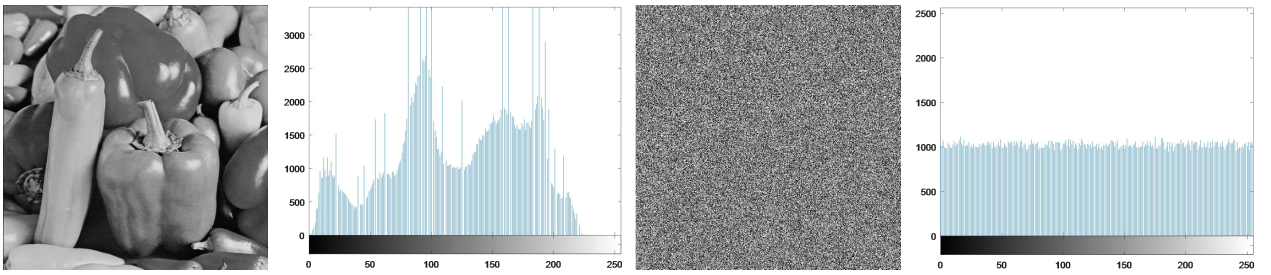
where k_1, k_2, and k_3 are the pseudosecret keys; θ, η, μ, p, and q are the parameters of the MLNCML system; C_1 and C_2 are produced via Eq. (21) and Eq. (22); n is generated from the plaintext image via Eq. (15); and l_1, l_2, l_3, and l_4 represent different numbers of rounds of Arnold scrambling.

The encryption algorithm presented in this paper is designed on the basis of a set of pseudosecret keys. During transmission, an attacker may intercept these secret keys, but because they are pseudosecret keys, the attacker cannot use them to crack the algorithm. Therefore, the algorithm proposed in this paper has a good ability to resist attacks, and the method of secret key cracking is invalid for this algorithm.

4.4 Histogram analysis

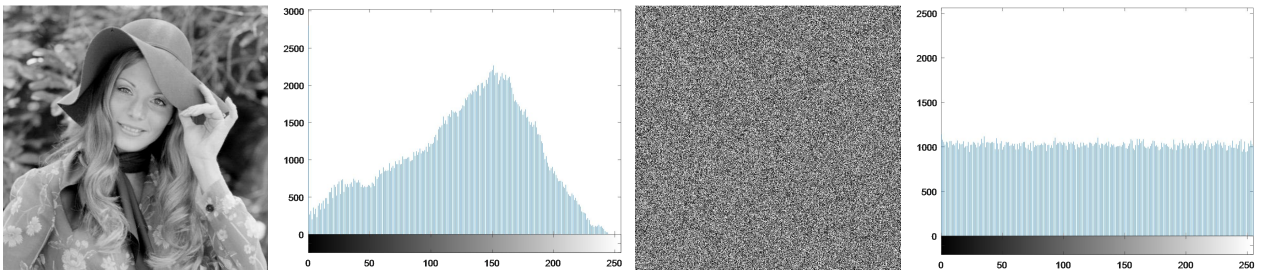
A histogram is a functional image formed by counting the numbers of points with the same pixel value, which reflect the distribution of the pixel values. Usually, the pixel value distribution of the original image will be uneven [28].

In Fig. 11 and 12, histograms of the pixel value distributions of the plaintext and ciphertext versions of two images, Peppers_gray and Hat_gray, are shown.



(a) Plaintext of Peppers_gray (b) Histogram of Peppers_gray (c) Ciphertext of Peppers_gray (d) Histogram of ciphertext Peppers_gray

Fig. 11 Histograms of plaintext Peppers_gray and ciphertext Peppers_gray



(a) Plaintext of Hat_gray (b) Histogram of Hat_gray (c) Ciphertext of Hat_gray (d) Histogram of ciphertext Hat_gray

Fig. 12 Histograms of plaintext Hat_gray and ciphertext Hat_gray

The pixel value distributions of the ciphertext images obtained using the encryption algorithm proposed in this paper are very uniform. Therefore, the proposed algorithm has a good ability to resist statistical analysis.

4.5 χ^2 test

Here, we quantitatively verify that the pixel values of the ciphertext images are evenly distributed. The most common test for this purpose is the χ^2 test. The calculation formula is [28]

$$\chi^2 = \sum_{i=0}^{255} \frac{(w_i - w_0)^2}{w_0}. \tag{24}$$

In Eq. (24), i represents a pixel value, w_i represents the number of times that i appears in the image, and $w_0 = (M \times N)/256$. The χ^2 values of the plaintext images and ciphertext images are shown in Table 1.

Table 1 χ^2 test results

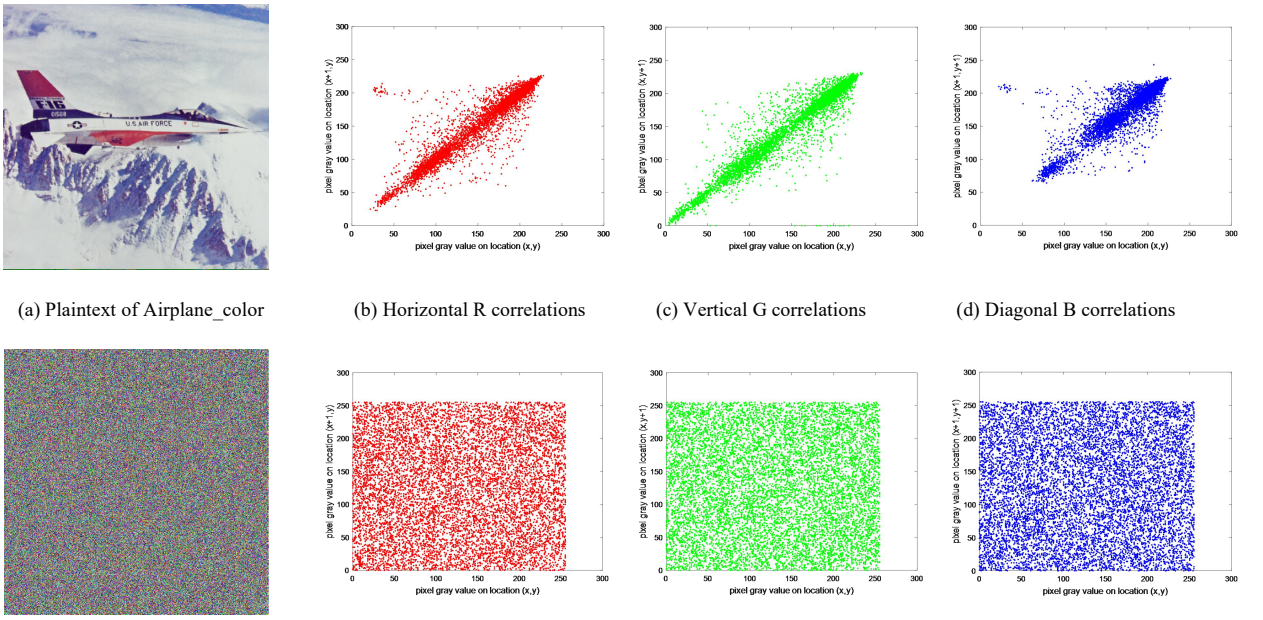
Image	Lena_gray	Barbara_gray	Peppers_gray	Hat_gray	Airplane_color_R	Airplane_color_G	Airplane_color_B
Plaintext	157665	144101	204333	102801	678425	677708	1107794
Ciphertext	283.2188	255.3281	266.4121	285.1211	246.5117	248.9355	265.3945

From Table 1, we can see that the χ^2 values of the ciphertext images are very small; specifically, they are all lower than 290. Thus, we can prove that the pixel value distribution generated by the proposed encryption method is uniform.

4.6 Correlation analysis

Another way to perform a statistical attack is to analyze the correlations between adjacent pixel values in a ciphertext image in order to find a pattern that can be used to crack the encryption algorithm.

In this section, 10000 pixels are randomly selected from the plaintext and ciphertext versions of Airplane_color. Fig. 13 shows the correlations between adjacent pixels in these images.



(a) Plaintext of Airplane_color (b) Horizontal R correlations (c) Vertical G correlations (d) Diagonal B correlations

(e) Ciphertext of Airplane_color (f) Horizontal ciphered R correlations (g) Vertical ciphered G correlations (h) Diagonal ciphered G correlations

Fig. 13 Correlation coefficients of Airplane_color

We use Eq. (25) and Eq. (26) to calculate the correlations of adjacent pixels in the plaintext and ciphertext images in three directions [3].

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{25}$$

$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \quad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \quad E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i. \tag{26}$$

The calculation results are shown in Table 2, and Table 3 presents a comparison with the results obtained using the encryption methods of Refs. [1, 2, 22, 24, 32, 33].

Table 2 Correlation coefficients of images

Image	Plaintext			Proposed encryption		
	Horizontal	Vertica	Diagonal	Horizontal	Vertical	Diagonal
Lena_gray	0.9719	0.9850	0.9593	0.0002	0.0022	− 0.0015
Barbara_gray	0.8594	0.9591	0.8417	0.0016	− 0.0011	0.0002
Peppers_gray	0.9806	0.9823	0.9710	− 0.0036	0.0023	0.0022
Hat_gray	0.9878	0.9864	0.9772	− 0.0005	− 0.0009	− 0.0006
Airplane_color_R	0.9725	0.9592	0.9371	0.0026	0.0008	− 0.0023
Airplane_color_G	0.9708	0.9669	0.9454	0.00002	− 0.0021	− 0.0025
Airplane_color_B	0.9635	0.9352	0.9154	− 0.0047	0.0046	0.0021

Table 3 Comparison of correlation coefficients

Image	Lena_gray	Ref. [33]	Ref. [2]	Ref. [22]	Ref. [32]	Ref. [24]	Ref. [1]
H	0.0002	− 0.0015	− 0.0084	− 0.0025	0.0053	− 0.0016	− 0.0067
V	0.0022	0.0041	− 0.0017	− 0.0029	− 0.0067	− 0.0026	− 0.0021
D	− 0.0015	0.0069	− 0.0019	− 0.0016	− 0.0022	0.0116	− 0.0027

As seen from the comparative experiments presented in Tables 2 and 3, through the encryption method proposed in this paper, the correlations between adjacent pixel values become very low. Compared with some other representative methods, the method proposed in this paper results in less correlation; therefore, the proposed encryption algorithm has a better ability to resist statistical analysis.

4.7 Information entropy analysis

The information entropy, which represents the randomness of the pixel value distribution, can be obtained as shown in Eq. (27):

$$H(a) = \sum_{i=0}^{2^L-1} p(a_i)\log_2\frac{1}{p(a_i)}. \tag{27}$$

In Eq. (27), $p(a_i)$ represents the probability of a_i. Theoretically, an information entropy that is closer to 8 indicates that the distribution of the pixel values is more chaotic.

Table 4 Information entropy of images

Image	Lena_gray	Barbara_gray	Peppers_gray	Hat_gray	Airplane_color_R	Airplane_color_G	Airplane_color_B
Plain	7.4474	7.4664	7.3967	7.6778	6.7178	6.8055	6.2140
Proposed	7.9992	7.9993	7.9993	7.9992	7.9993	7.9993	7.9993

Table 5 Information entropy comparison

Image	Average	Ref. [33]	Ref. [2]	Ref. [22]	Ref. [32]	Ref. [24]	Ref. [1]
Information entropy	7.9993	7.9935	7.9974	7.9993	7.9992	7.9972	7.9971

Table 4 shows the information entropy results for the plaintext and ciphertext images, and Table 5 presents a comparison with the results obtained using the encryption methods of Refs. [1, 2, 22, 24, 32, 33]. After the application of the encryption algorithm proposed in this paper, the information entropy of the ciphertext approaches 8; therefore, the proposed algorithm has high security.

4.8 Differential attack

By transforming one or more pixel values in a plaintext image, an attacker can obtain a new decrypted image and observe the differences in the pixel values between the two encrypted images to find a pattern that can be used to crack the encryption algorithm.

Two important indicators for a differential attack are the number of pixels change rate (NPCR) and the unified average changing intensity (UACI), the values of which are obtained as shown in Eq. (29) and Eq. (30) [28]:

$$NPCR = \frac{\sum_{i,j} E(i,j)}{A \times B} \times 100\%, \tag{29}$$

$$UACI = \frac{1}{A \times B}[\sum_{i,j} \frac{|m_1(i,j) - m_2(i,j)|}{255}] \times 100\%. \tag{30}$$

In Eq. (29) and Eq. (30), A and B represent the image width and height, respectively, and m_1 and m_2 are the two ciphertext images obtained after changing a pixel value in the original plaintext image. If $m_1(i,j) \neq m_2(i,j)$, $E(i,j) = 1$; otherwise, $E(i,j) = 0$.

Theoretically, the closer the values of the NPCR and UACI (between the two encrypted images) are to 99.6093% and 33.4635%, respectively, the better the encryption effect. Table 7 compares the NPCR and UACI results of the proposed algorithm with those of the encryption algorithms of Refs. [1, 2, 22, 24, 32, 33].

Table 6 Average NPCR and UACI values between two ciphertexts

Image	Lena_gray	Barbara_gray	Peppers_gray	Hat_gray	Airplane_color_R	Airplane_color_G	Airplane_color_B
NPCR (%)	99.6040	99.5998	99.5838	99.5960	99.6101	99.6056	99.6014
UACI (%)	33.4736	33.4305	33.5194	33.4988	33.4403	33.4778	33.4699

Table 7 NPCR and UACI comparison with other algorithms

Image	Average	Ref. [33]	Ref. [2]	Ref. [22]	Ref. [32]	Ref. [24]	Ref. [1]
NPCR (%)	99.6001	99.6098	99.6180	99.6100	99.6181	99.5982	99.5800
UACI (%)	33.4729	33.4697	33.4960	33.4600	33.4991	33.4396	30.5840

As seen from Table 6 and Table 7, the NPCR and UACI values between two ciphertexts generated using the proposed algorithm are very close to the theoretical ideal values, indicating that it is difficult for attackers to crack this algorithm through differential attacks. Therefore, the algorithm proposed in this paper has a good ability to resist differential attacks.

Similarly, when we calculate the NPCR and UACI values that reflect the differences between corresponding plaintext and ciphertext images, the experimental results are as shown in Table 8.

Table 8 NPCR and UACI values between a plaintext image and the corresponding ciphertext image

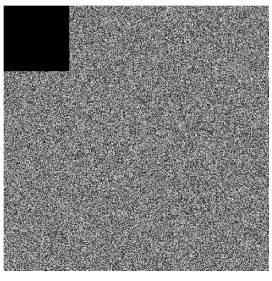
Image	Lena_gray	Barbara_gray	Peppers_gray	Hat_gray	Airplane_color_R	Airplane_color_G	Airplane_color_B
NPCR (%)	99.6033	99.6010	99.6025	99.6342	99.6006	99.5937	99.6109
UACI (%)	28.6797	28.8584	29.6301	28.8339	29.9714	28.6369	31.2619

Table 8 shows that the NPCR between corresponding plaintext and ciphertext images is approximately 99.60% and that the UACI is approximately 30%. These findings indicate that there are large differences between the plaintext and ciphertext images. Therefore, it is impossible for an attacker to obtain the plaintext image from the ciphertext image.

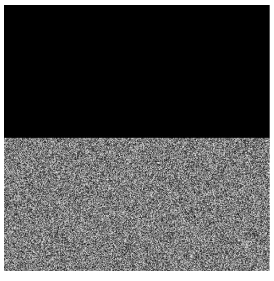
4.9 Robustness evaluation

Images may lose some information after they are encrypted, or they may be affected by noise. Therefore, it is very important for an encryption algorithm to exhibit good robustness.

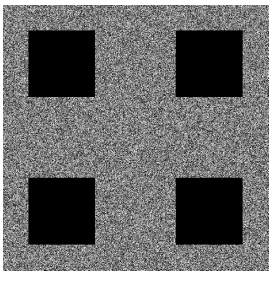
In this section, the robustness of the proposed algorithm against clipping and noise attacks is tested. Fig. 14 shows the results of various ciphertext clipping attacks and subsequent restoration. The experimental results show that the proposed algorithm has a good ability to resist clipping attacks. Even if some of the ciphertext information is lost, the receiver can approximately restore the information of the plaintext image through the decryption algorithm.



(a) Full encryption, 1/16 block attack



(b) Full encryption, 1/2 block attack

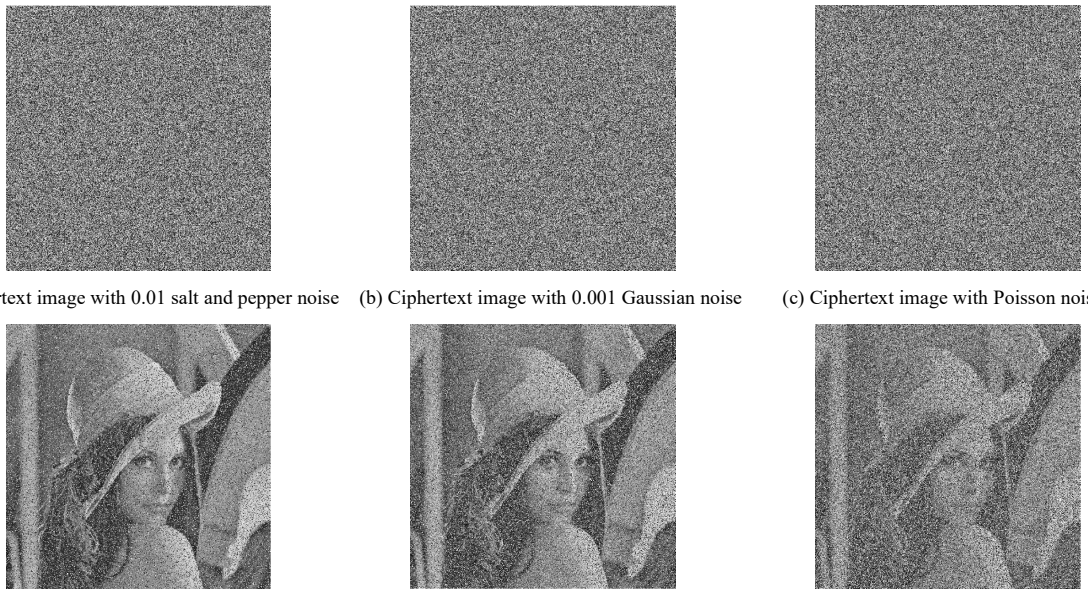


(c) Full encryption, 1/4 block attack

(d) Decrypted image corresponding to (a) (e) Decrypted image corresponding to (b) (f) Decrypted image corresponding to (c)

Fig. 14 Clipping attacks on Lena_gray

Fig. 15 shows the ability of the proposed algorithm to resist noise attacks. The experimental results show that this algorithm can resist all kinds of noise attacks. Even if the ciphertext is affected by noise, the receiver can restore a large portion of the original image information using the decryption algorithm.

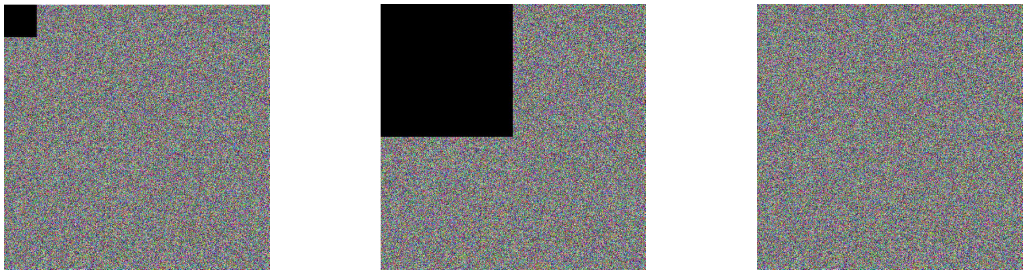


(a) Ciphertext image with 0.01 salt and pepper noise (b) Ciphertext image with 0.001 Gaussian noise (c) Ciphertext image with Poisson noise

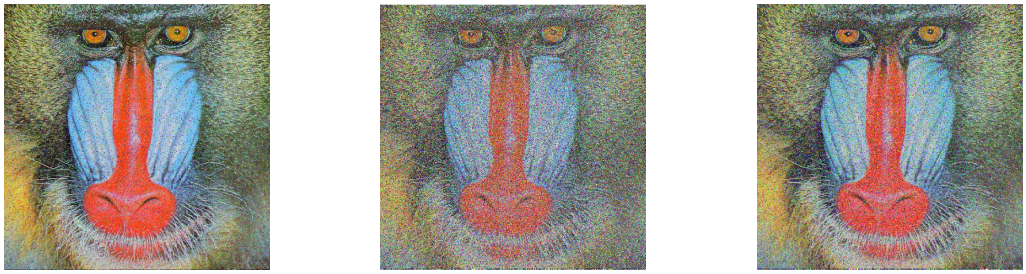
(d) Decrypted image corresponding to (a) (e) Decrypted image corresponding to (b) (f) Decrypted image corresponding to (c)

Fig. 15 Various noise attacks on Lena_gray

The robustness was also tested on a color image, and the results are shown in Fig. 16.



(a) Full encryption, 1/64 block attack (b) Full encryption, 1/4 block attack (c) Ciphertext image with 0.01 salt and pepper noise



(d) Decrypted image corresponding to ((e) Decrypted image corresponding to (b) (f) Decrypted image corresponding to (c)

Fig. 16 Robustness evaluation on Baboon_color

The results show that even if the ciphertext loses some information or is affected by noise, some of the plaintext information can be retrieved through the decryption algorithm. Therefore, the proposed algorithm shows good robustness, not only for grayscale images but also for color images.

4.10 Classic types of attacks

There are four classic types of attacks: (1) ciphertext-only attacks, (2) known-plaintext attacks, (3) chosen-plaintext attacks, and (4) chosen-ciphertext attacks.

The chosen-plaintext attack mode is known to be the strongest; therefore, if an algorithm can resist chosen-plaintext attacks, then it can resist all of the above attack modes [30].

The algorithm proposed in this paper uses plaintext information in the diffusion stage. Based on the plaintext information, different chaotic sequences are generated, which are then applied during diffusion. The encryption algorithm proposed in this paper can be regarded as a one-time-secret encryption algorithm. Therefore, it can resist chosen-plaintext attacks. It can also resist all four of the above attack types. In addition, a special set of pseudosecret keys is designed in this paper. Even if these pseudosecret keys are cracked, it is impossible for an attacker to use them to crack the algorithm. Moreover, because of the novelty of the STP diffusion method proposed in this paper, an attacker cannot find an analogous encryption algorithm in a database of known algorithms; thus, the security of the proposed encryption algorithm is improved.

4.11 Time efficiency analysis

The encryption time of an algorithm is an important index for testing its performance, as an algorithm with an encryption time that is too long is not suitable for practical application. The encryption algorithm proposed in this paper uses matrix multiplication in the diffusion phase. MATLAB uses highly optimized libraries for matrix multiplication; hence, ordinary MATLAB-based matrix multiplication is very fast. The proposed encryption algorithm was tested on a computer running Windows 10 with an i5 processor, 8 GB of memory, and MATLAB R2017a. For comparison with other encryption algorithms, we encrypted the Lena image 100 times and obtained the average value. The results are shown in Table 9.

Table 9 Encryption time comparison (s)

Image size	Ours	Ref. [4]	Ref. [12]	Ref. [23]	Ref. [3]
256×256	0.16	0.58	0.0949	0.3440	0.9810
512×512	0.62	-	0.4010	1.3357	3.8539

The experimental results show that the time efficiency of the proposed algorithm is better than that of most other algorithms, indicating that this algorithm can be widely used.

5 Conclusion

In this paper, a compound secret key encryption algorithm based on STP theory is proposed, which consists of scrambling followed by diffusion. First, the pixels of the original plaintext image are randomly divided into four blocks, and a different number of rounds of Arnold transformation is carried out on each block. Then, a set of pseudosecret keys is provided, and the real secret key is obtained by filtering with a synchronously updating Boolean network. Finally, in combination with the MLNCML system, the STP technique is used for diffusion to obtain the encrypted image. Experimental results show that the proposed algorithm offers good security and can be used not only on grayscale images but also on color images.

In future work, we plan to create a more complex secret key generator. In the design of the new secret key generator, we will incorporate additional types of networks beyond merely Boolean networks: probabilistic Boolean networks (a probabilistic Boolean network is more complex than a Boolean network; it can be regarded as a combination of Boolean networks subject to a certain probability distribution) and multivalued logical networks (as a natural generalization of a Boolean network, a multivalued logical network has more than two possible values per node; consequently, it is better able to characterize the dynamic behavior of genes in cells). The structures of the networks discussed above are more complex than that of a Boolean network, but they can still be expressed as sets of mathematical equations; therefore, in theory, they should be well suited to be used as secret key generators.

Acknowledgments: This research was supported by the National Natural Science Foundation of China (No. 61672124), the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund (No. MMJJ20170203), a Project of the Liaoning Province Science and Technology Innovation Leading Talents Program (No. XLYC1802013), the Key R&D Projects of Liaoning Province (No. 2019JH2/10300057), and the Jinan City ‘20 Universities’ Funding Projects Introducing Innovation Team Program (No. 2019GXRC031), "Double First-rate" Construction Project ("Innovation Project") (No. SSCXXM012).

References

[1] B. Abd-El-Atty, A.A. Abd El-Latif, S.E. Venegas-Andraca. An encryption protocol for NEQR images based on one-particle quantum walks on a circle. Quantum Inf. Process., 18 (9) (2019), p. 272.

[2] M. Alawida, A. Samsudin, J. Sen Teh, R.S. Alkhawaldeh. A new hybrid digital chaotic system with applications in image encryption. Signal Process., 160 (2019), pp. 45-58.

[3] X.L. Chai, Y.R. Chen, L. Broyde. A novel chaos-based image encryption algorithm using DNA sequence operations. Opt. Lasers Eng., 88 (2017), pp. 197-213.

[4] X.L. Chai, X.Y. Zheng, Z.H. Gan, D.J. Han, Y.R. Chen. An image encryption algorithm based on chaotic system and compressive sensing. Signal Process., 148 (2018), pp. 124-144.

[5] D.Z. Cheng, H.S. Qi. A linear representation of dynamics of Boolean networks. IEEE Trans. Autom. Control, 55 (10) (2010), pp. 2251-2258.

[6] D.Z. Cheng, H.S. Qi, Z.Q. Li. Analysis and control of Boolean networks: a semi-tensor product approach. Springer Science & Business Media, London, 2010.

[7] D.Z. Cheng, Y. Zhao. Semi-tensor product of matrices-A convenient new tool. Chin. Sci. Bull., 56 (32) (2011), pp. 2664-2674.

[8] Q. Gong, H.J. Wang, Y. Qin, Z.P. Wang. Modified diffractive-imaging-based image encryption. Opt. Lasers Eng., 121 (2019), pp. 66-73.

[9] M.M. Guan, X.L. Yang, W.S. Hu. Chaotic image encryption algorithm using frequency-domain DNA encoding. IET Image Process., 13 (9) (2019), pp. 1535-1539.

[10] T. Hilberdink. Quasi Kronecker products and a determinant formula. Linear Alg. Appl., 536 (2018), pp. 87-102.

[11] Z.Y. Hua, Y.C. Zhou. Image encryption using 2D Logistic-adjusted-Sine map. Inf. Sci., 339 (2016), pp. 237-253.

[12] Z.Y. Hua, Y.C. Zhou, H.J. Huang. Cosine-transform-based chaotic system for image encryption. Inf. Sci., 480 (2019), pp. 403-419.

[13] C. Huang, J.Q. Lu, D.W.C. Ho, G.S. Zhai, J.D. Cao. Stabilization of probabilistic Boolean networks via pinning control strategy. Inf. Sci., 510 (2020), pp. 205-217.

[14] S. Kauffman, C. Peterson, B. Samuelsson, C. Troein. Genetic networks with canalyzing Boolean rules are always stable. Proc. Natl. Acad. Sci. U. S. A., 101 (49) (2004), pp. 17102-17107.

[15] J. Kaur, N. Jindal. A secure image encryption algorithm based on fractional transforms and scrambling in combination with multimodal biometric keys. Multimed. Tools Appl., 78 (9) (2019), pp. 11585-11606.

[16] J.S. Khan, J. Ahmad, S.S. Ahmed, H.A. Siddiqa, S.F. Abbasi, S.K. Kayhan. DNA key based visual chaotic image encryption. J. Intell. Fuzzy Syst., 37 (2) (2019), pp. 2549-2561.

[17] X.S. Kong, S.L. Wang, H.T. Li, F.E. Alsaad. New developments in control design techniques of logical control networks. Front. Inform. Technol. Elect. Eng., 21 (2) (2020), pp. 220-233.

[18] B.W. Li, J.Q. Lu, J. Zhong, Y. Liu. Fast-Time Stability of Temporal Boolean Networks. IEEE Trans. Neural Netw. Learn. Syst., 30 (8) (2019), pp. 2285-2294

[19] H.T. Li, G.D. Zhao, M. Meng, J. Feng. A survey on applications of semi-tensor product method in engineering. Sci. China-Inf. Sci., 61 (1) (2018), p. 010202.

[20] H.C. Liu, Y. Liu, Y.Y. Li, Z. Wang, F.E. Alsaadi. Observability of Boolean networks via STP and graph methods. IET Contr. Theory Appl., 13 (7) (2018), pp. 1031-1037.

[21] J.Q. Lu, M.L. Li, T.W. Huang, Y. Liu, J.D. Cao. The transformation between the Galois NLFSRs and the Fibonacci NLFSRs via semi-tensor product of matrices. Automatica, 96 (2018), pp. 393-397.

[22] E.G. Nepomuceno, L.G. Nardo, J. Arias-Garcia, D.N. Butusov, A. Tutueva. Image encryption based on the pseudo-orbits from 1D chaotic map. Chaos, 29 (6) (2019), p. 061101.

[23] P. Ping, F. Xu, Y.C. Mao, Z.J. Wang. Designing permutation–substitution image encryption networks with Henon map. Neurocomputing, 283 (2018), pp. 53-63.

[24] D. Ravichandran, P. Praveenkumar, J.B.B. Rayappan, R. Amirtharajan. DNA chaos blend to secure medical privacy. IEEE Trans. Nanobiosci., 16 (8) (2017), pp. 850-858.

[25] L.Z. Sun, B.D. Zheng, Y.M. Wei, C.J. Bu. Generalized inverses of tensors via a general product of tensors. Front. Math. China, 13 (4) (2018), pp. 893-911.

[26] N. Tsafack, J. Kengne, B. Abd-El-Atty, A.M. Iliyasu, K. Hirota, A.A. Abd El-Latif. Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption. Inf. Sci., 515 (2020), pp. 191-217.

[27] J. Wang, Y.C. Geng, L. Han, J.Q. Liu. Quantum image encryption algorithm based on quantum key image. Int. J. Theor. Phys., 58 (1) (2019), pp. 308-322.

[28] X.Y. Wang, L. Feng, H.Y. Zhao. Fast image encryption algorithm based on parallel computing system. Inf. Sci., 486 (2019), pp. 340-358.

[29] X.Y. Wang, Z.M. Li. A color image encryption algorithm based on Hopfield chaotic neural network. Opt. Lasers Eng., 115 (2019), pp. 107-118.

[30] X.Y. Wang, L. Teng, X. Qin. A novel colour image encryption algorithm based on chaos. Signal Process., 92 (4) (2012), pp. 1101-1108.

[31] S.P. Wen, Z.G. Zeng, T.W. Huang, Q.G. Meng, W. Yao. Lag synchronization of switched neural networks via neural activation function and applications in image encryption. IEEE Trans. Neural Netw. Learn. Syst., 26 (7) (2015), pp. 1493-1502.

[32] M. Xu, Z.H. Tian. A novel image cipher based on 3D bit matrix and latin cubes. Inf. Sci., 478 (2019), pp. 1-14.

[33] Q.Y. Xu, K.H. Sun, C. Cao, C.X. Zhu. A fast image encryption algorithm based on compressive sensing and hyperchaotic map. Opt. Lasers Eng.,121 (2019), pp. 203-214.

[34] J. Yao, J.E. Feng, M. Meng. On solutions of the matrix equation AX=B with respect to semi-tensor product. J. Frankl. Inst.-Eng. Appl. Math, 353 (5) (2016), pp. 1109-1131.

[35] Y.Y. Yu, J.E. Feng, J.F. Pan, D.Z. Cheng. Block decoupling of Boolean control networks. IEEE Trans. Autom. Control, 64 (8) (2019), pp. 3129-3140.

[36] Y.Y. Yu, M. Meng, J.E. Feng, Y. Gao. An adjoint network approach to design stabilizable switching signals of switched Boolean networks. Appl. Math. Comput., 357 (2019), pp. 12-22.

[37] H. Zhang, X.Y. Wang, X.H. Lin. Synchronization of Boolean networks with different update schemes. IEEE-ACM Trans. Comput. Biol. Bioinform., 11 (5) (2014), pp. 965-972

[38] H. Zhang, X.Y. Wang, X.H. Lin. Synchronization of asynchronous switched Boolean network. IEEE-ACM Trans. Comput.

Biol. Bioinform., 12 (6) (2015), pp. 1449-1456.

[39] Q. Zhang, L. Guo, X.P. Wei. A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik, 124 (18) (2013), pp. 3596-3600.

[40] X.H. Zhang, H.X. Han, Z.J. Sun, W.D. Zhang. Alternative approach to calculate the structure matrix of Boolean network with semi-tensor product. Contr. Theory Appl., 11 (13) (2017), pp. 2048-2057

[41] Y.Q. Zhang, X.Y. Wang. Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation. Nonlinear Dyn., 77 (3) (2014), pp. 687-698.

[42] Y.Q. Zhang, X.Y. Wang. Spatiotemporal chaos in mixed linear–nonlinear coupled logistic map lattice. Physica A, 402 (2014), pp. 104-118.

[43] Y.S. Zhang, W.Y. Wen, M.T. Su, M. Li. Cryptanalyzing a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik, 125 (4) (2014), pp. 1562-1564.

[44] G.D. Zhao, H.T. Li, P.Y. Duan, F.E. Alsaadi. Survey on applications of semi-tensor product method in networked evolutionary games. J. Appl. Anal. Comput., 10 (1) (2019), pp. 32-54.

[45] J. Zhong, Y. Liu, K.I. Kou, L.J. Sun, J.D. Cao. On the ensemble controllability of Boolean control networks using STP method. Appl. Math. Comput., 358 (2019), pp. 51-62.

[46] N.R. Zhou, W.W. Chen, X.Y. Yan, Y.Q. Wang. Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system. Quantum Inf. Process., 17 (6) (2018), p. 137.

[47] N.R. Zhou, Y.Q. Hu, L.H. Gong, G.Y. Li. Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations. Quantum Inf. Process., 16 (6) (2017), p. 164.

[48] N.R. Zhou, S.M. Pan, S. Cheng, Z.H. Zhou. Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing. Opt. Laser Technol., 82 (2016), pp. 121-133.

[49] N.R. Zhou, X.Y. Yan, H.R. Liang, X.Y. Tao, G.Y. Li. Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system. Quantum Inf. Process., 17 (12) (2018), p. 338.

[50] Z.L. Zhu, W. Zhang, K.W. Wong, H. Yu. A chaos-based symmetric image encryption scheme using a bit-level permutation. Inf. Sci., 181 (6) (2011), pp. 1171-1186.