

Received 31 May 2022, accepted 30 June 2022, date of publication 14 July 2022, date of current version 11 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3190851

RESEARCH ARTICLE

Region of Interest Encryption Based on Novel 2D Hyperchaotic Signal and Bagua Coding Algorithm

TZE-HAN CHEN¹ AND CHENG-HSIUNG YANG^{1,2} 

¹Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

²Center of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

Corresponding author: Cheng-Hsiung Yang (yangch@mail.ntust.edu.tw)

This work was supported by the Graduate Institute of Automation and Control of the National Taiwan University of Science and Technology.

ABSTRACT In recent years, image processing has attracted a lot of attention due to its high accuracy to detect and classify objects in images. Therefore, based on the encryption algorithm, this paper adds deep learning to construct an algorithm that uses a hyperchaotic system for the region of interest image encryption and explores its security. First, we design a novel two-dimensional hyperchaotic map, and for the first time, we use a coding architecture called Bagua coding. We combine the above two points to enhance the effect of the permutation process, and consequently, the complexity of our encryption scheme. The algorithm also uses the features extraction on the plaintext and SHA-256 to generate secret key, coupled with the advanced exclusive-or operation and bit shift calculation to encrypt the plaintext. Next, we import YoloV3 and UNet for object detection and selection. Users can automatically select the region of interest on the image and use an encryption algorithm to encrypt the selected part of the irregular region. Finally, we perform security analysis on ciphertext image. The security analysis results on the generated ciphertext image validate our proposed encryption framework against statistical and differential attack.

INDEX TERMS Hyperchaotic map, bagua coding, image encryption, region of interest, deep learning.


I. INTRODUCTION

Due to the recent developments of the Internet of Thing (IoT), public networks, and mobile devices, a large amount of data, such as audio and video, is transmitted through the internet. Sensitive information, for example military orders, medical files, and personal data, are susceptible of unauthorized access and use. Traditional encryption schemes, such as DES and AES, are not suitable for image encryption because of the diffusion of adjacent pixel values has a high degree of relevance [1].

Chaotic system shows interesting properties, for example, it generates pseudo-random sequence which is essential component for security applications, unpredictability, and has huge key space. Shannon proposed a permutation and diffusion encryption method in 1949, which laid the foundation for the encryption algorithm [2]. Later, Vladimir

Arnold proposed an image encryption based on chaotic map in 1967 [3]. Later, chaos-based encryption was proposed for image encryption in the 1990s [4]–[6]. Since the 2000s, more encryption schemes based on chaotic signal have been proposed [7]–[10]. Moreover, some studies use one-dimensional chaotic map, such as logistic map or sine map, to design multi-dimensional chaotic system [11], [12]. These researches showed that chaos-based encryption schemes have a great potential for security applications.

On the other hand, parallel DNA coding-based encryption is a distributive algorithm with high performance computing, low power consumption, and faster computation time [13], [14]. Some studies also include complementary rules and XOR operations in their encryption schemes [15], [16]. In recent years, some studies propose encryption schemes based on chaos theory in combination with crossover operations [17]–[19]. In this paper, we design a new coding framework according to the trigram in the Book of Changes called Bagua coding. In addition to the

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

most basic encoding process of converting pixel values into symbols, this encoding operation includes some steps, such as shuffling the Bagua coding operation [20], and the concept of double coding. The purpose is to increase the effect of hiding plaintext information.

Selective encryption shows an advantage due to its lower computational cost and improved performance. However, some proposed schemes the selection of the object of interest is not performed automatically [21], or the automated region of interest is not selected correctly [22]. In order to improve the computational efficiency and reduce the memory usage, this paper proposes an encryption scheme where only encrypt the region of interest is encrypted [21]–[23]. In this paper, we exploit the advantages of chaotic-based pseudo random number generation and Bagua cryptography algorithms to develop an efficient encryption scheme. Moreover, automatic object detection is also incorporated in our algorithm in order to obtain an improved performance.

The rest of this paper is organized as follow: In section 2, we introduce a new two dimensional Logistic-jointed-Trigonometric map (2D-LJTM) and its dynamical properties is investigated. The full description of the 2D-LJTM based image encryption algorithm (2D-LJTM IEA) is given in section 3. Section 4 describes the object detection in our proposed encryption scheme. In section 5, we perform security analysis in order to verify the efficacy of the proposed method. Finally, we conclude this paper in section 6.

II. TWO-DIMENSIONAL HYPER CHAOTIC MAP

The section introduces two-dimensional Logistic-jointed-Trigonometric map (2D-LJTM) and compares with two existing chaotic maps.

A. 2D LOGISTIC JOINTED TRIGONOMETRIC

In [11], [12], and [24]–[26], the author uses Logistic map [27] and trigonometric function to combine into a new chaotic system that exhibits highly random and unpredictable characteristics. Based on the above methods, we design a novel two-dimensional hyper chaotic map called 2D-LJTM which is defined as (1),

$$\begin{cases} x_{i+1} = \sin(\pi(3y_i + 1)x_i(1 - x_i)) \\ y_{i+1} = (\cos(3\pi ax_{i+1}(1 - x_{i+1})) + 3)y_i(1 - y_i) \end{cases} \quad (1)$$

where i is the number of iterations, a is the control parameter.

2D-LJTM combines one dimensional Logistic map and trigonometric function to expand into two-dimensional system which shows a larger range of chaos, better ergodicity and unpredictability. We set the initial values of x and y as 0.1, the control parameter is set as 0.01. We display the trajectories of 2D-LJTM in Fig. 1(a).

Fig. 1(b) and Fig. 1(c) show the trajectories of two-dimensional Sine Logistic modulation map (2D-SLMM) [11] and two-dimensional Logistic-modulated-Sine-coupling-Logistic chaotic map (LMSCL) [12]. It is found that our proposed 2D-LJTM has a more uniform degree of dispersion.

B. BIFURCATION DIAGRAM

Bifurcation diagram is the study of the mathematical theory of period doubling or topological structure, which occurs gradual based on the system parameters, which can be fixed points, periodic trajectories, or chaotic attractors. The at the moment when the parameters of the dynamic system change slightly [26]. It shows the values that tend to be bifurcation diagrams of the 2d chaotic maps are plotted in fig. 2. we set the control parameter range between $[0, 1]$, and we know that when any value of 2D-LJTM exhibit a hyper chaotic behavior.

C. LYAPUNOV EXPONENT

We use the Lyapunov exponent to analyze the chaotic behavior of dynamic systems [28]. The chaotic behavior can be described as unpredictable and sensitive to initial values. The Lyapunov exponent it used to quantify the degree of divergence of two similar trajectories of a dynamic system. In the phase plane, the initial conditions of the two trajectories differ by, and the rate of dissolution is defined as (2)

$$|\delta Z(t)| \approx e^{\lambda t} |\delta Z(0)| \quad (2)$$

among the definition, λ is the Lyapunov exponent, and according to different λ values, the system will have three different states. First, when $\lambda < 0$, the trajectory will converge to a fixed point or periodic orbit. Furthermore, when $\lambda = 0$ means that the system is conserved, we call it Lyapunov stability, and the trajectory will be in a certain steady-state mode. Finally, when $\lambda > 0$ the orbit is unstable and chaotic. No matter how close the initial conditions of the two trajectories are, the system will produce completely different trajectories after iteration.

The number of Lyapunov exponents is consistent with the system dimension. The largest λ indicates the predictability of the system. When the maximum λ is positive, the system is called a chaotic system. When more than one λ is positive, the system is called a hyperchaotic system. Hyperchaotic systems have higher complexity and trajectories, making them more difficult to predict. For discrete system. λ is defined in (3).

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \quad (3)$$

Fig. 3 show the Lyapunov exponents of the 2D chaotic maps. We can observe that our proposed 2D-LJTM has a larger range of chaotic behavior. In addition, for the control parameter of the 2D-LJTM, the value of λ at each point is greater than 0, indicating that 2D-LJTM is a hyper chaotic system within the range.

D. NIST SP 800 TEST

The test standard for random and pseudo-random number generators for cryptography applications [29] is the Federal Information Processing Standard (FIPS) issued by the National Institute of Standards and Technology (NIST). The NIST test consists of a total of 15 test items. These tests are used to test the randomness of the binary sequence generated

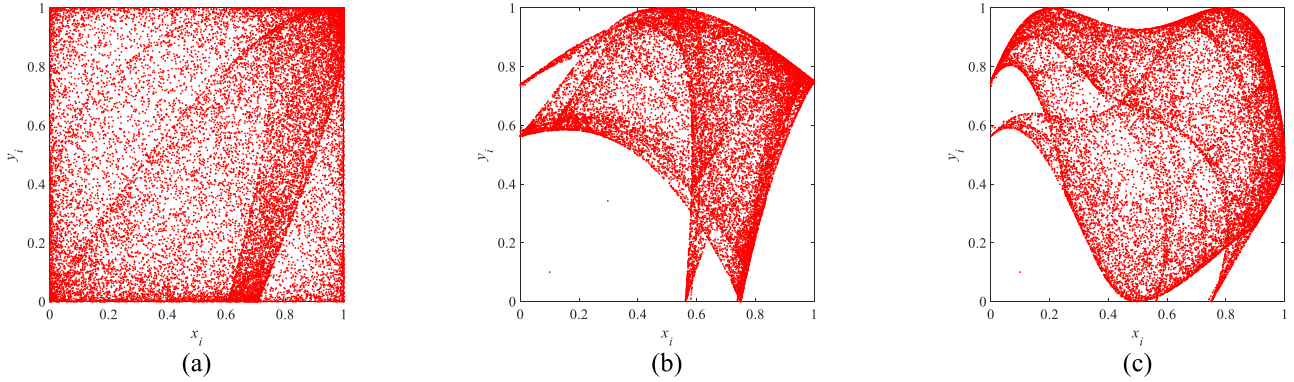


FIGURE 1. Trajectories of 2D chaotic maps: (a) 2D-LJTM with control parameter $\alpha = 0.01$; (b) 2D-SLMM with control parameter $\alpha = 1, \beta = 3$; (c) LSMCL with control parameter $\alpha = 0.75, \beta = 3$.

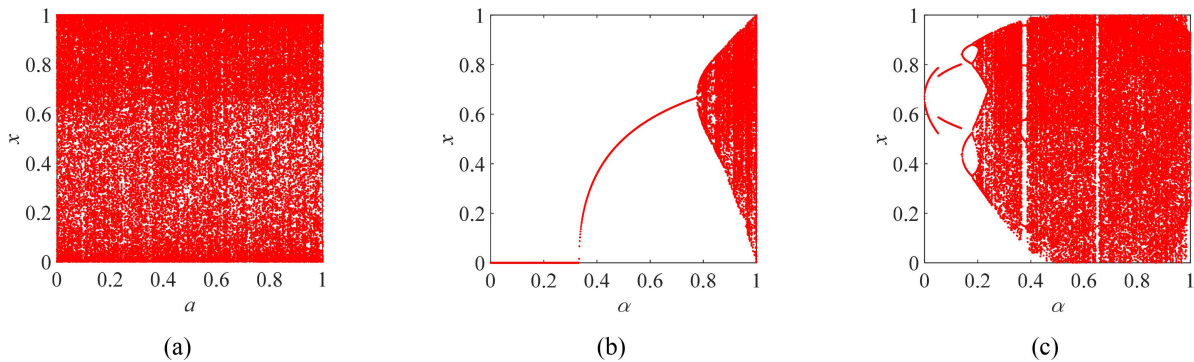


FIGURE 2. Bifurcation diagrams of 2D chaotic maps: (a) 2D-LJTM with control parameter $\alpha = 0.01$; (b) 2D-SLMM with control parameter $\beta = 3$; (c) LSMCL with control parameter $\beta = 3$.

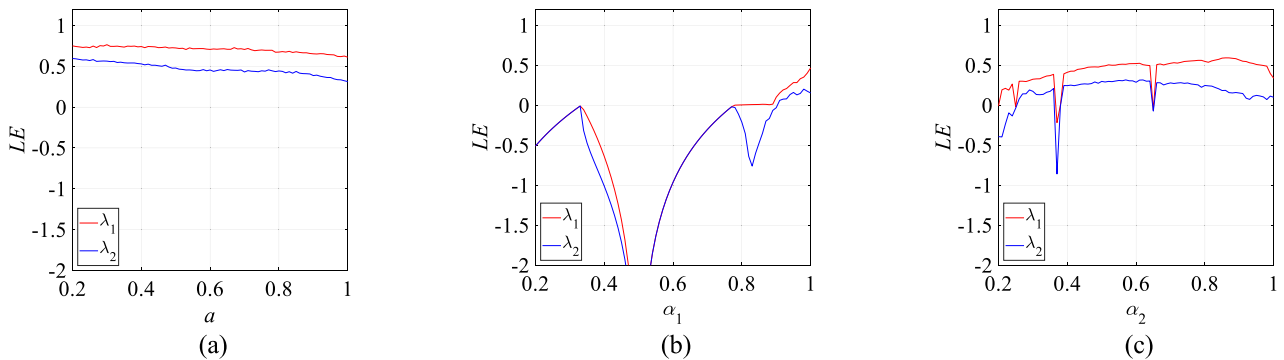


FIGURE 3. Lyapunov exponents of 2D chaotic maps (a) 2D-LJTM with control parameter $\alpha = 0.01$; (b) 2D-SLMM with control parameter $\beta = 3$; (c) LSMCL with control parameter $\beta = 3$.

by the random and pseudo-random generators. The statistics of the NIST test are used to test the P -value, which summarizes the strength of support for the null hypothesis. For these tests, each P -value is a perfect random number generator with less random probability than the sequence generated by the tested sequence. We can choose the level of significance α for the test. If $P > \alpha$, it means that the null hypothesis is accepted, and the sequence is random. If $P < \alpha$, means rejecting the null hypothesis and the sequence is non-random. For the entire result, the following two methods are used to explain the passing of the NIST test:

(1) The consistency of the P -value distribution.

(2) The proportion of the sequence that passed the statistical test. Use the following confidence interval to define the acceptable range of ratio as (4),

$$\left[\hat{P} - \sqrt{\hat{P}(1 - \hat{P})/n}, \hat{P} + \sqrt{\hat{P}(1 - \hat{P})/n} \right] \quad (4)$$

where $\hat{P} = 1 - \alpha$, n is the selected sample size. If the ratio is within the interval, there is evidence that the sequence is random.

To pass all NIST tests, the P -value needs to be greater than for the sequence to be considered random. Therefore, Table 1 shows our proposed 2D-LJTM passes all NIST tests.

TABLE 1. NIST test results for x sequence 2D-LJTM.

Test Name	The P-value	The proportion	Result
Frequency	0.907419	991/1000	Pass
Block-Frequency	0.947308	985/1000	Pass
Runs	0.741918	993/1000	Pass
Longest Run	0.593478	991/1000	Pass
Rank	0.723804	992/1000	Pass
FFT	0.382115	980/1000	Pass
Non-Overlapping	0.925287	993/1000	Pass
Overlapping	0.076658	986/1000	Pass
Universal	0.056426	987/1000	Pass
Linear Complexity	0.952152	985/1000	Pass
Serial	0.892036	991/1000	Pass
Approximate Entropy	0.832561	987/1000	Pass
Cumulative	0.767582	989/1000	Pass
Random Excursions	0.966721	640/648	Pass
Random Excursions Variant	0.902994	640/648	Pass

It means that the sequence generated by 2D-LJTM is random and suitable for encryption.

III. 2D-LJTM BASED IMAGE ENCRYPTION ALGORITHM

In this section, the 2D-LJTM IEA encryption scheme is proposed and discussed in detail. The encryption algorithm exploits the unique features of Bagua coding, which is a novel coding framework based on the trigram of the Book of Changes. The flow chart of the 2D-LJTM IEA is depicted in Fig. 4.

A. SECRET KEY GENERATOR

In cryptography, a brute-force attack means that an attacker can search for any possible secret key in the key space until the correct secret key is found to recover the ciphertext. The feasibility of a brute force attack depends on the total number of secret keys used in the encryption system. The key space of an encryption process based on chaos must be greater than 2^{100} [1] to prevent the attacker from brute force attacks. Our proposed secret key is composed of two binary sequences. Shannon entropy of a plain text is defined as follows,

$$En = \text{IEEE754}((-\sum_{i=0}^{255} p(s_i) \log_2 p(s_i))^{24}) \quad (5)$$

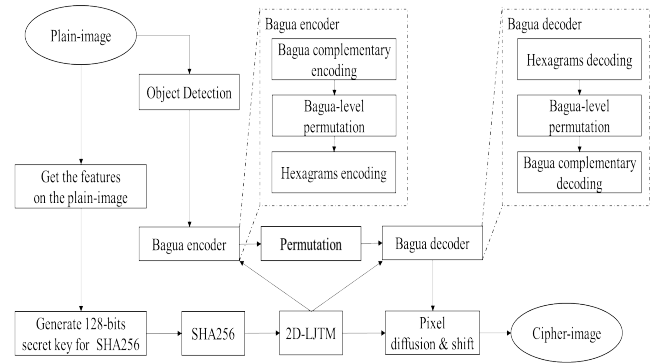


FIGURE 4. Flow chart of the 2D-LJTM IEA.

where $p(s_i)$ is the probability of the pixel value S and IEEE754 is the binary floating point arithmetic standard. The second sequence is the summation of weighted pixel values in the plaintext using the following equation,

$$S(m, n) = \text{IEEE754}((\sum_{m=1}^H \sum_{n=1}^W mn \times v(m, n))^{1.4}) \quad (6)$$

where H and W represent the length and width in the plaintext respectively, $v(m, n)$ is the pixel value at (m, n) position, and IEEE754 is the binary floating-point arithmetic standard. Therefore, the combination of these two sequences has a key space of 2^{128} , which is long enough to resist brute force attacks. The Secure Hash Algorithm 256-bits (SHA-256) [30] is incorporated in our secret key generator to improve the key sensitivity, and the final key space K is 2^{256} .

The generation of initial states of 2D-LJTM is shown in Algorithm 1 We define $K = \{x_0, y_0, a, \omega, \gamma\}$, where (x_0, y_0) are the initial value of the chaotic system with a size of 64 bits, a is the control parameter of 2D-LJTM with a size of 64 bits, and (ω, γ) are interference weight and parameter. Both have a size of 32 bits. The secret key K generate three different initial values $(x_0^1, y_0^1, a_1), (x_0^2, y_0^2, a_2)$ and (x_0^3, y_0^3, a_3) . With these initial values, the 2D-LJTM generates six distinct chaotic sequences which will be used in the following section.

B. BEGUA CODING AND PERMUTATION OPERATION

In the Book of Changes, there is the Taiji. Taiji refers to the state of chaos when the world is not yet being civilized. It generates two complementary modes. The two modes generate four forms. The four forms generate eight trigrams and then, these eight trigrams generate sixty-four hexagrams. We use these two modes to generate Yan and Yin. Yan and Yin are represented by the (☰) and (☷) symbols, respectively. The four forms represent spring, summer, autumn, and winter, which are derived from the intersection of Yan and Yin. Finally, the four forms add another line to become eight trigrams, namely Qian (☰), Dui (☱), Li (☲), Zhen (☳), Xun (☴), Kan (☵), Gen (☶), and Kun (☷). It is derived from the eight phenomena that can be represented in the universe. The phenomena represented are Qian to the heaven, Dui to

TABLE 2. Bagua complementary coding rules.

Rule	Q	D	L	Z	X	C	G	K
1	000	001	010	011	100	101	110	111
2	111	110	101	100	011	010	001	000

the lake, Li to the fire, Zhen to the thunder, Xun to the wind, Kan to the water, Gen to the mountain, and Kun to the ground.

Algorithm 1 The Generation of Initial States

Input: Secret key K with length of 256 bits.

Output: Initial states $(x_0^1, y_0^1, a_1), (x_0^2, y_0^2, a_2)$ and (x_0^3, y_0^3, a_3) .

1. $x_0 = \sum_{i=1}^{64} K[i] \times 2^{i-1} / 2^{64}$;
2. $y_0 = \sum_{i=65}^{128} K[i] \times 2^{i-65} / 2^{64}$;
3. $a_0 = \sum_{i=129}^{192} K[i] \times 2^{i-129} / 2^{64}$;
4. $\omega = \sum_{i=193}^{224} K[i] \times 2^{i-193} / 2^{32}$;
5. $\gamma = \sum_{i=225}^{256} K[i] \times 2^{i-225} / 2^{32}$;
6. $x_0^1 = (\omega \times \gamma / x_0) \bmod 1$;
7. $y_0^1 = (\omega \times \gamma / y_0) \bmod 1$;
8. $a_1 = (\omega \times \gamma / a_0) \bmod 1$;
9. for $i = 1$ to 2 do
10. $x_0^{i+1} = \text{LJTM}_x(x_0^i, y_0^i, a_i)$; {LJTM_x(x, y, a) : output x value of 2D-LJTM.}
11. $y_0^{i+1} = \text{LJTM}_y(x_0^i, y_0^i, a_i)$; {LJTM_y(x, y, a) : output y value of 2D-LJTM.}
12. $a_{i+1} = (a_i / (x_0^{i+1} \times y_0^{i+1})) \bmod 1$;
13. end for

In the Bagua coding rule, we set Qian to Q, Dui to D, Li to L, Zhen to Z, Xun to X, Kan to C, Gen to G, and Kun to K. Using these eight trigrams and the sixty-four hexagrams, combined with Bagua-level permutation, a new set of coding scheme is designed and proposed. We represent Yan and Yin as 0 and 1, respectively. If the three lines in the Bagua are presented in the form of three bits from bottom to top, there will be two complementary rules, as shown in Table 2. Among them, rule 1 regards the symbol of the unbroken line as 0 in binary, and rule 2 is its complementary relationship.

In order to enhance the encryption performance, algebraic operations [15], [16], such as XOR, are also included in our algorithm. The application of Bagua coding in encryption, referring to the transfer characteristics of waves in physics, the energy of the wave shifts the bit value as the medium, which is called Bagua-level permutation. It can be seen from Table 3 that there is a total of eight rules for Bagua-level permutation. The bit value of the Bagua coding ranges from 000 to 111, and it is 0 to 7 when converted to decimal. When the rule i is reached, the previous bit value plus i can get the current bit value after shifting, and the value of i is between 1 and 8. If the bit value after adding i is greater than 8, it will be passed back to the minimum value 1, and the operation will continue.

TABLE 3. Bagua-level permutation rules.

1	2	3	4	5	6	7	8
Q→D	Q→L	Q→Z	Q→X	Q→C	Q→G	Q→K	Q→Q
D→L	D→Z	D→X	D→C	D→G	D→K	D→Q	D→D
L→Z	L→X	L→C	L→G	L→K	L→Q	L→D	L→L
Z→X	Z→C	Z→G	Z→K	Z→Q	Z→D	Z→L	Z→Z
X→C	X→G	X→K	X→Q	X→D	X→L	X→Z	X→X
C→G	C→K	C→Q	C→D	C→L	C→Z	C→X	C→C
G→K	G→Q	G→D	G→L	G→Z	G→X	G→C	G→G
K→Q	K→D	K→L	K→Z	K→X	K→C	K→G	K→K

2 1	Q	D	L	Z	X	C	G	K
K	0C	2D	23	10	14	08	17	02
G	21	1F	38	3E	35	27	34	0F
C	06	2F	00	28	3B	1D	04	07
X	2C	1C	32	20	39	30	12	2E
Z	19	11	15	33	2A	03	1B	18
L	0D	31	1E	37	25	3F	16	24
D	0A	3A	26	36	3D	3C	29	13
Q	01	2B	0E	22	09	05	1A	0B

FIGURE 5. Corresponding square matrix of the eight trigrams and the sixty-four hexagrams.

The next step, the sixty-four hexagrams coding step will be carried out next. We use the Bagua complementary coding combine with sixty-four hexagrams coding to complete the coding twice. Compare with single encoding, this double encoding can achieve the effect of completely concealing the original signal.

Our sixty-four hexagram coding uses a similar architecture to the Substitution-box (S-Box). This is an important structure in encryption schemes [31], [32] since it is mainly used for the association between secret keys and ciphertext. In our application, we use the S-box as the connection between the Bagua complementary coding and the sixty-four hexagram coding. The coding method of the sixty-four hexagrams is the same as in the Book of Changes. They are all obtained by combining the eight trigrams in pairs, each contains six lines in total. Then, using the above-mentioned Bagua coding rules, the eight kinds of coding symbols are combined in pairs as input, and the output is 00 to 3F represented by hexadecimal, there are 64 kinds in total. The corresponding square matrix of the eight trigrams and the sixty-four hexagrams is shown in Fig. 5.

In order to illustrate our proposed algorithm, a simple numerical example is depicted in Fig. 6. First, we use 2D-LJTM to generate an array of pseudo-random values of the same size as the total number of pixels of a given image, and we combine these two arrays. We will refer to the array

generated by 2D-LJTM and the image array as M and I , respectively. All pixel values in I are converted into 8 bits, and the length of each element in M is 22 bits. What needs to know is that the longer the bit length after combining, the smaller the probability that each element is the same. Then, the combined elements will start encoding.

The encoding process is divided into three steps. The first step is the Bagua complementary encoding operation, the second one is the Bagua-level permutation operation, and the last is sixty-four hexagrams encoding operation. Among them, the Bagua complementary encoding and the Bagua-level permutation are all based on the chaotic system. In the Bagua complementary encoding operation, each position has a length of 30-bit, which is encoded once every three bits. The chaotic system is used to determine of Bagua elements B_{ij}^e . Then, the Bagua-level system is used to select and execute one of the eight Bagua-level permutation rules in Table 3. Finally, 10 groups of Bagua-level permutation elements C_{ij}^e , with 2 bits as a group, and converted into hexadecimal sequences S_{ij}^e according to Fig. 7. After finishing the Bagua encoding step, we use the S_{ij}^e as the input of the permutation operation to shuffle the pixel positions. The permutation step only moves the pixel position based on the S_{ij}^e and does not affect the pixel value.

So far, we still need to perform Bagua decoding and extract the last 8-bit pixel value as the output after permutation operation. The steps of Bagua decoding are similar to the steps of encoding. Among them, the Bagua complementary decoding step and the sixty-four hexagrams decoding step need to do the inverse operation, while the Bagua-level permutation step is not required. That is to say, the Bagua-level permutation operation is the same in the encoding and decoding steps. We present a numerical example as shown in Fig. 7. Suppose the output of sixty-four hexagrams decoding operation is C_{ij}^d , the output of Bagua-level permutation operation is B_{ij}^d , and the output of 8-bit pixel value is T .

C. DIFFUSION OPERATION

Every image encryption algorithm must include diffusion characteristics. The diffusion means that even small changes in the plaintext will produce very different ciphertext. As mentioned in [33], the diffusion property represents a one-bit change in the plaintext, which will cause each bit in the ciphertext to change with a probability of 50%. In this paper, we use two methods to change the pixel value of the image. First, the 8-bit pixel value output of the Bagua decoding operation is assumed to be T , and the chaotic sequence generated by 2D-LJTM is assumed to be S . Assuming that the length of the image is H and the width is W , the first diffusion operation is defined as (7),

$$Q_i = \begin{cases} T_i \oplus S_i & \text{if } i = 1 \\ Q_{i-1} \oplus T_i \oplus S_i & \text{if } i = [2, H \times W] \end{cases} \quad (7)$$

where Q is the result after the first diffusion operation, \oplus is the bitwise exclusive-OR (XOR) operation. As for the

decryption process of the first diffusion step is to do the inverse operation. It is defined as (8),

$$T_i = \begin{cases} Q_{i-1} \oplus Q_i \oplus S_i & \text{if } i = [2, H \times W] \\ Q_i \oplus S_i & \text{if } i = 1 \end{cases} \quad (8)$$

The second diffusion operation we use to change the pixel-value of the image is the bit shifting. According to the chaotic matrix generated by 2D-LJTM, the pixel value can be shifted from a minimum of 1 bit to the right to a maximum of 8 bits, and when the bit moves to the least significant bit, it will become the most significant bit. After diffusion operation, the entire encryption process is completed. The shifting operation is illustrated in Fig. 8.

IV. SELECTIVE IMAGE ENCRYPTION

In the literature, several encryption schemes have been proposed, most of them encrypt an entire image. However, in some cases, according to different senders and receivers, the areas that we want to encrypt and decrypt are not the same, so we tend to encrypt only the most important parts of the image. Based on the above purpose, we propose to use YoloV3 and UNet to recognize and select objects of a given image. In this paper, we will use the human body as an important area in the image to realize region of interest encryption, and the flowchart is shown in Fig. 9.

A. YOLOV3 AND UNET

You only look once (Yolo) is a real-time multi-target detection algorithm based on deep convolutional neural network [34]. The differences between its predecessor YoloV2 [35] and YoloV3 [36] are presented below. The main difference between YoloV3 and YoloV2 lies in the Darknet architecture. YoloV3 uses Darknet-53, with a total of 53 convolutional layers, replacing Darknet-19 used by YoloV2. This makes YoloV3 sacrifice speed compared to YoloV2, but it improves accuracy. YoloV3 can also predict on three different scales, each of which predicts three bounding boxes individually. UNet is a deep convolutional neural network for semantic segmentation. Neural network was originally used for biomedical image segmentation, but with its robustness and high performance, UNet can be a good tool for semantic segmentation [37].

We use YoloV3 in combination with UNet to detect and select object accurately. First, use YoloV3 to obtain the region of interest, and then use UNet to perform semantic segmentation for the region of interest obtained by YoloV3. Finally, the object contour is identified and encrypted. The encryption algorithm has been explained in the previous section.

B. SELECTIVE IMAGE ENCRYPTION

First, the first step is to read the images in the dataset, and then use YoloV3 to perform object detection, and select the region of the human body in the image, which is called the region of interest. The second step, following the action of the first step, will use the area selected by YoloV3 to perform semantic

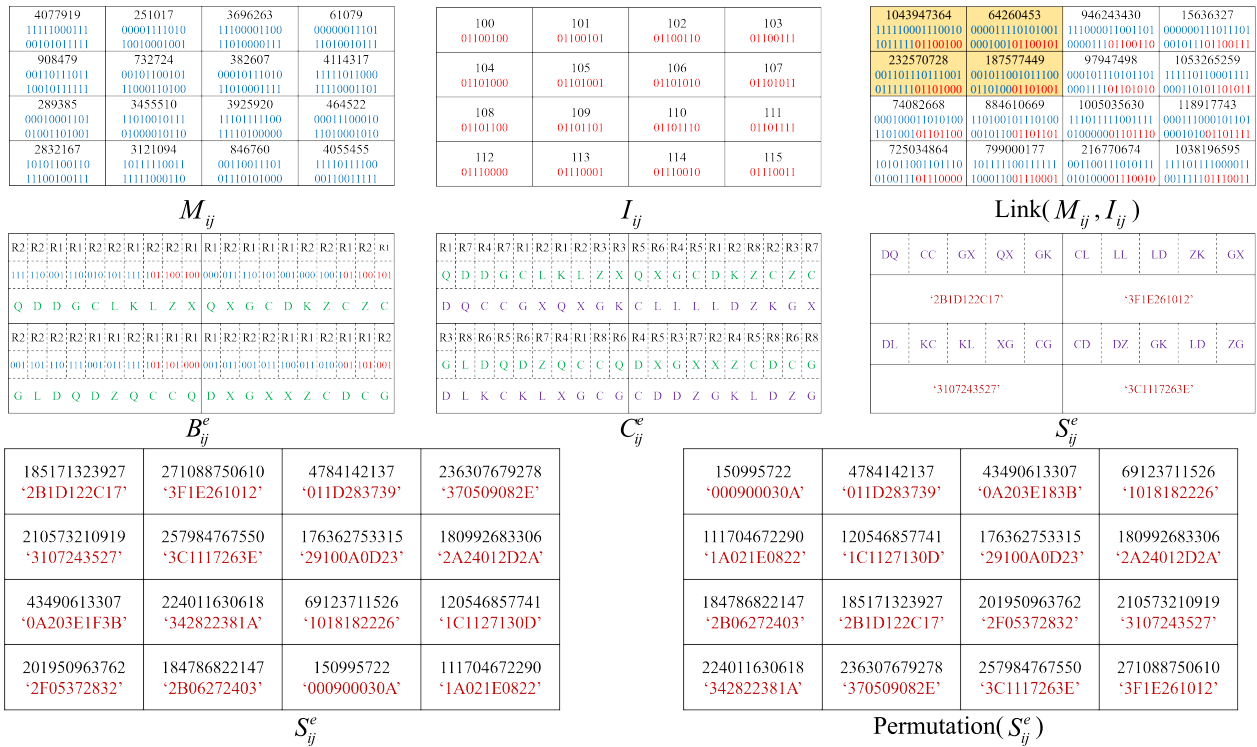


FIGURE 6. Numerical example of Bagua encoding and permutation.

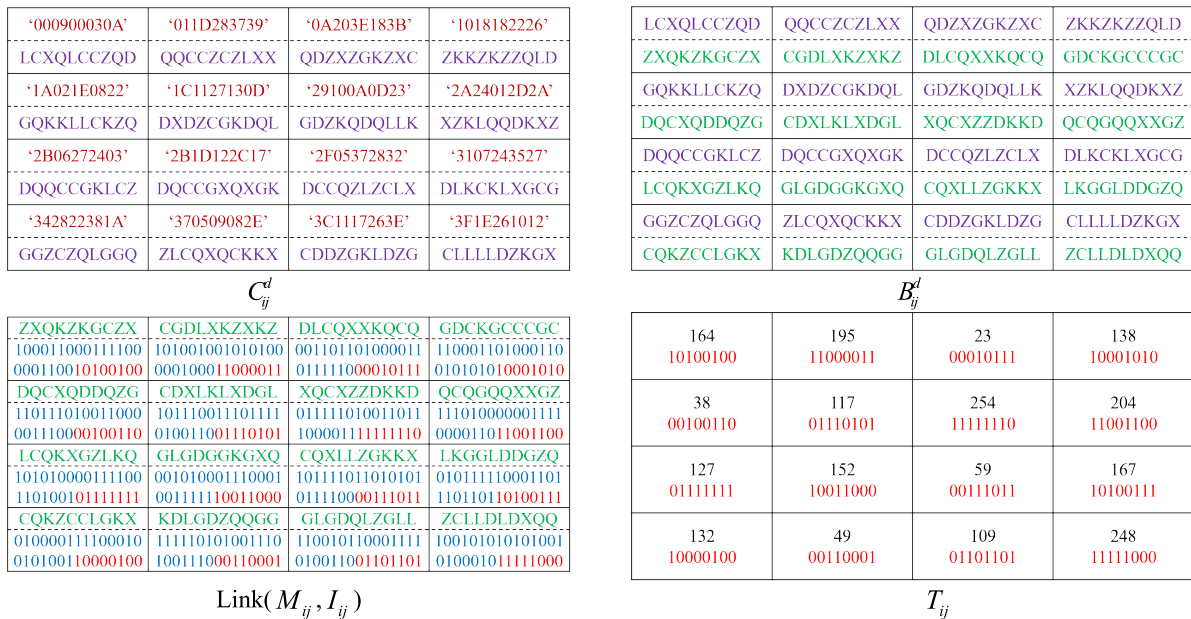


FIGURE 7. Numerical example of Bagua decoding.

segmentation using UNet to separate the human body and background in the rectangular area.

Next, after the human body is separated from the background, it still need to do image binarization. The background part that does not contain the human body is set to a pixel value of 0, which is black; and all parts that contain the human

body are set to a pixel value of 255, which is called foreground mask.

The last step is to encrypt the image that has been separated and binarized. For the pixel values position of the foreground mask part marked with the human body, encrypt it at the corresponding position of the plain-image, and then encryption

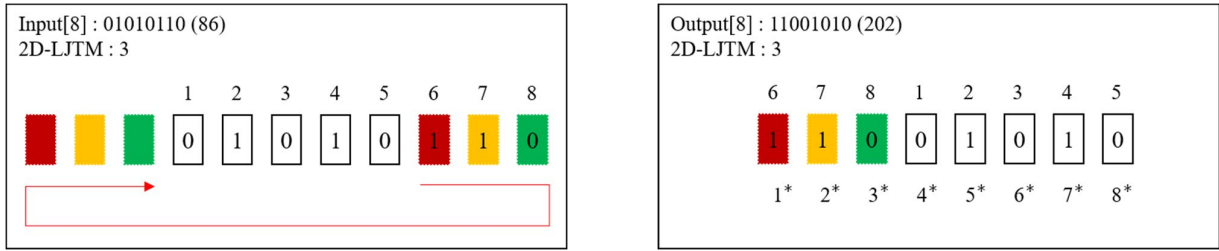


FIGURE 8. Bit shifting operation.

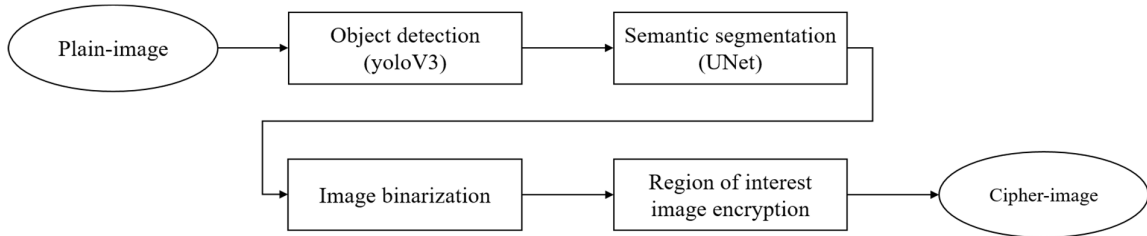


FIGURE 9. Flowchart of region of interest image encryption.

process can be completed in the region of interest. We use the images in the flicker8k dataset to display examples as shown in Fig. 10.

V. SECURITY ANALYSIS

In this section we will analyze the security of encryption algorithm. Due to the more complex chaotic behavior of 2D-LJTM and the Bagua coding proposed in this paper for the first time, the 2D-LJTM IEA process has the following advantage:

- (1) When the difference of initial conditions in the hyperchaotic system is very small, the results produced by 2D-LJTM are also very difficult to predict, which has very high security for encryption.
- (2) Different secret keys will be generated according to different images, rather than just using the same key, which makes the ciphertext very different even if there are only slight differences in the plaintext.
- (3) As it satisfies the principle of permutation and diffusion [2], therefore, each pixel in the plaintext can be replaced to any position, and small changes will spread to all pixels in the ciphertext.
- (4) It has good reliability against noise or data loss attacks. When there is noise in the ciphertext, 2D-LJTM IEA can nevertheless restore the plaintext to a high degree.

The methods used for verification include key analysis, chosen-plaintext and chosen-ciphertext attacks analysis, histogram analysis, correlation analysis, differential attack analysis, Shannon entropy analysis, robustness analysis. Most of the test images we analyze come from the USI-SIPI image data set.

A. KEY ANALYSIS

Key analysis can be divided into two parts. One is key space analysis, and the other is key sensitivity analysis. As mentioned in Section 3.1, the key space must be large enough to

resist brute force attacks. Our key space is 256 bits, which is large enough to meet the 100 bits required in [1]. On the other hand, key sensitivity means that the key must be sufficiently sensitive during encryption and decryption. In order to verify the key sensitivity, we select an image, and generate the correct key k_1 , then generate two more keys k_2 and k_3 which are one bit different with k_1 , the results are shown in fig. 11 and fig. 12, respectively. Fig. 11 denotes that the cipher-image encrypted by the key with only one-bit difference will also be completely different; fig. 12 illustrates that only under the correct key can the cipher-image be completely decrypted back to the plain-image.

B. CHOSEN-PLAINTEXT AND CHOSEN-CIPHERTEXT ATTACKS ANALYSIS

In cryptanalysis, the chosen-plaintext and chosen-ciphertext attacks are two extensively used security attack models. The chosen-plaintext attacks refer to the ability of the attackers to obtain the corresponding ciphertext through arbitrary plaintext, and the chosen-ciphertext attacks mean that the attackers can use a certain method to obtain the decrypted result through any ciphertext [38]. In order to resist the above chosen-plaintext attacks and s chosen-ciphertext attacks, we have designed the following architectures in the 2D-LJTM IEA in this paper:

- (1) The plaintext will be converted into a random hexadecimal sequence through the Bagua encoding process at the beginning. This mechanism can ensure that the input before the permutation process is random, thereby resisting the chosen-plaintext attacks initiated by the attackers
- (2) Use the key generator related to the plaintext to make different images have different keys. The key generated by a selected plaintext cannot be used to decrypt another ciphertext, thus resisting the chosen-ciphertext attacks launched by the attackers.

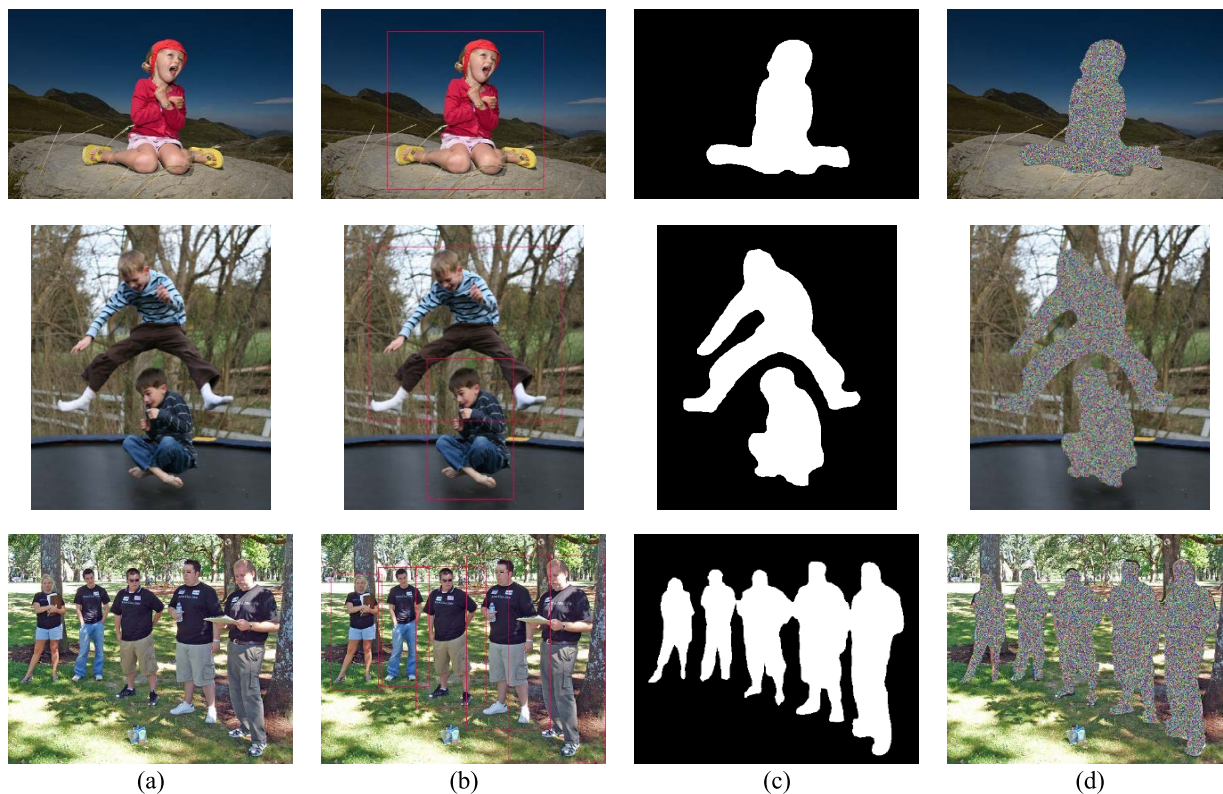


FIGURE 10. Examples of region of interest image encryption: (a) plain-image; (b) output after YoloV3 object detection; (c) Output after the UNet semantic segmentation; (d) cipher-image.

- (3) The operation of the permutation process can shuffle the pixel value in the plaintext to any position. The operation of the diffusion process can make any small changes in the plaintext be distributed in all the ciphertext. This makes it impossible for the attacker to deduce the ciphertext from the plaintext, nor to restore the plaintext from the ciphertext.

C. HISTOGRAM ANALYSIS

The image histogram is used to show the distribution of image pixel values [39]. Therefore, histogram analysis is often used to verify the encryption effect of image encryption. For an ideal encryption system, the ciphertext after the encryption process should have a consistent histogram representation. From Fig. 13, we show the histogram of images with different types after encryption. It is clear that the histogram of cipher-images reveals a high degree of consistency and is significantly different from the plain-images. Furthermore, the Chi-square test [40] can be used to quantitatively verify the uniformity of cipher-image, which is defined as (9),

$$X^2 = \sum_{i=1}^k \frac{(O_i - T_i)^2}{T_i} \tag{9}$$

where k is the value determined according to the gray level, O_i is the number of pixel values appearing in each gray level, T_i is the number of pixel values that should theoretically be

expected to appear in each gray level. For an 8-bit grayscale image, when the significance level $\alpha = 0.05$, the critical value $X_{0.05}^2 = 293.2478$. In other words, when $X^2 < X_{0.05}^2$, the number of pixel values appearing in each gray level of the cipher-image presents a uniform distribution. Table 4 shows that all grayscale images encrypted by 2D-LJTM IEA pass the Chi-square test

D. CORRELATION ANALYSIS

Correlation coefficient refers to the degree of similarity between the values of two adjacent pixels [41]. Generally, there is a high degree of correlation between adjacent pixel values of a plain-image. In contrast, the correlation coefficient of the image after encryption will be closer to 0, which means that the adjacent pixel values of the cipher-image are very different and unpredictable. We analyze the correlation coefficients in three directions, which are horizontal, vertical, and diagonal, the correlation coefficient is defined as (10),

$$CC_{xy} = \frac{\sum_{m=1}^M \sum_{n=1}^N (x_{m,n} - \bar{x})(y_{m,n} - \bar{y})}{\sqrt{\sum_{m=1}^M \sum_{n=1}^N (x_{m,n} - \bar{x})^2 \sum_{m=1}^M \sum_{n=1}^N (y_{m,n} - \bar{y})^2}} \tag{10}$$

where x is the measured pixel value of an image, y is the average value of \bar{x} and \bar{y} are average value of x and y , respectively. Fig. 14 shows the correlation coefficient diagram of

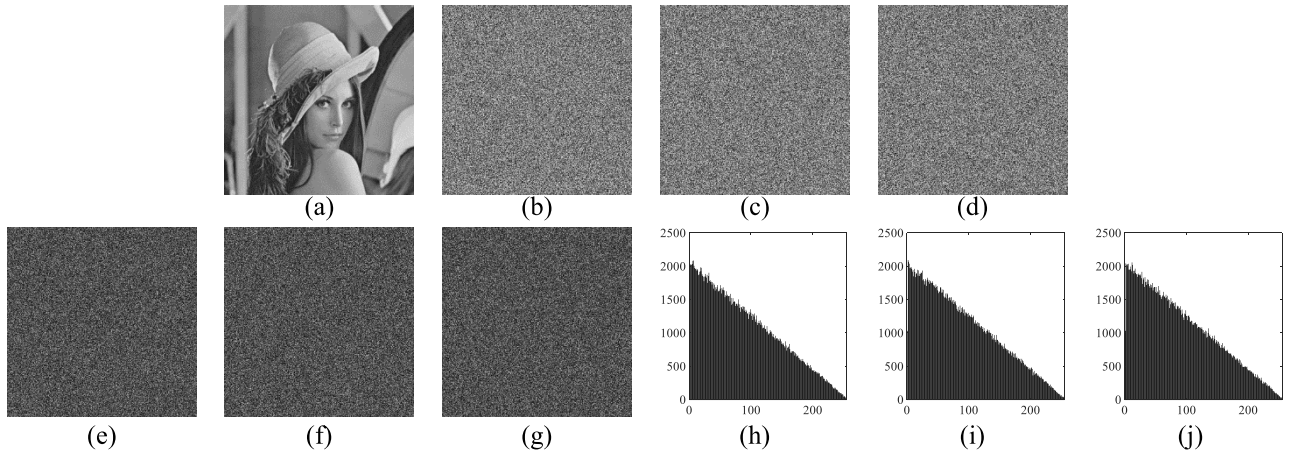


FIGURE 11. Key sensitivity analysis in the encryption process: (a) plain-image; (b) cipher-image (K_1); (c) cipher-image (K_2); (d) cipher-image (K_3); (e) |(b)-(c)|; (f) |(c)-(d)|; (g) |(b)-(d)|; (h) histogram of (e); (i) histogram of (f); (j) histogram of (g).

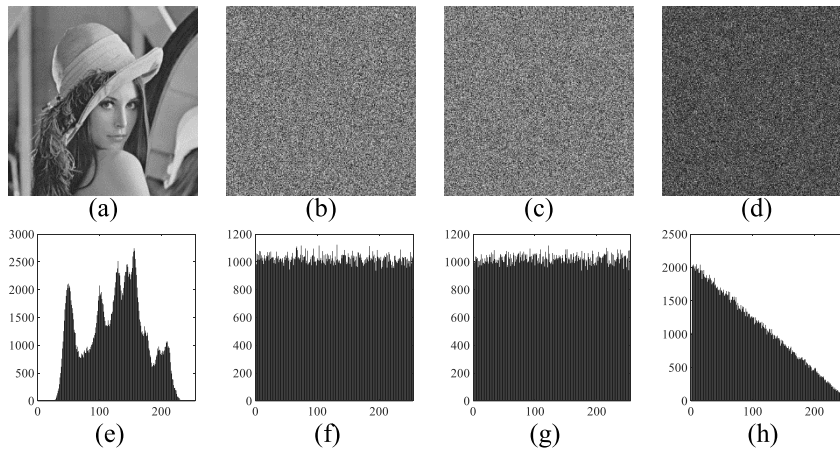


FIGURE 12. Key sensitivity analysis in the decryption process: (a) result of decryption (K_1); (b) result of decryption (K_2); (c) result of decryption (K_3); (d) |(b)-(c)|; (e) histogram of (a); (f) histogram of (b); (g) histogram of (c); (h) histogram of (d).

TABLE 4. The Chi-square test of encrypted grayscale image.

Cipher image	Lena	AirplaneU2	Airfield
χ^2 value	217.6504	225.5420	210.3960
$\chi^2_{0.05}$ value	293.2478	293.2478	293.2478
Result	Pass	Pass	Pass

the Lena grayscale image for three directions. From Table 5, we use different image encryption algorithms to encrypt the Lena grayscale image, and comparison results of 2D-LJTM IEA, [11], [12], [24], [42]. It can be found that the cipher-image presents good concealment, and the 2D-LJTM IEA has a smaller correlation coefficient in the horizontal, vertical, and average directions.

E. DIFFERENTIAL ATTACK ANALYSIS

which refers to Differential attack is one of the chosen-plaintext attacks, which refers to comparison and analysis

TABLE 5. Correlation coefficient of cipher-image with different encryption algorithms.

Grayscale image "Lena"	Horizontal	Vertical	Diagonal	Average
2D-LJTM IEA	3.3929×10^{-5}	1.4710×10^{-4}	1.3665×10^{-3}	5.1584×10^{-4}
Ref. [11]	1.4571×10^{-3}	1.0222×10^{-2}	5.5521×10^{-3}	5.7437×10^{-3}
Ref. [12]	2.2170×10^{-4}	1.3367×10^{-3}	5.6140×10^{-4}	7.0660×10^{-4}
Ref. [24]	1.2326×10^{-3}	1.2646×10^{-3}	5.9570×10^{-4}	1.0310×10^{-3}
Ref. [42]	4.6641×10^{-3}	1.2056×10^{-3}	4.0801×10^{-3}	3.3166×10^{-3}

changes of the plaintext with subtle differences after encryption [7]. An outstanding encryption algorithm must have strong resistance to differential attacks. We use two indicators to examine whether the image encryption algorithm is

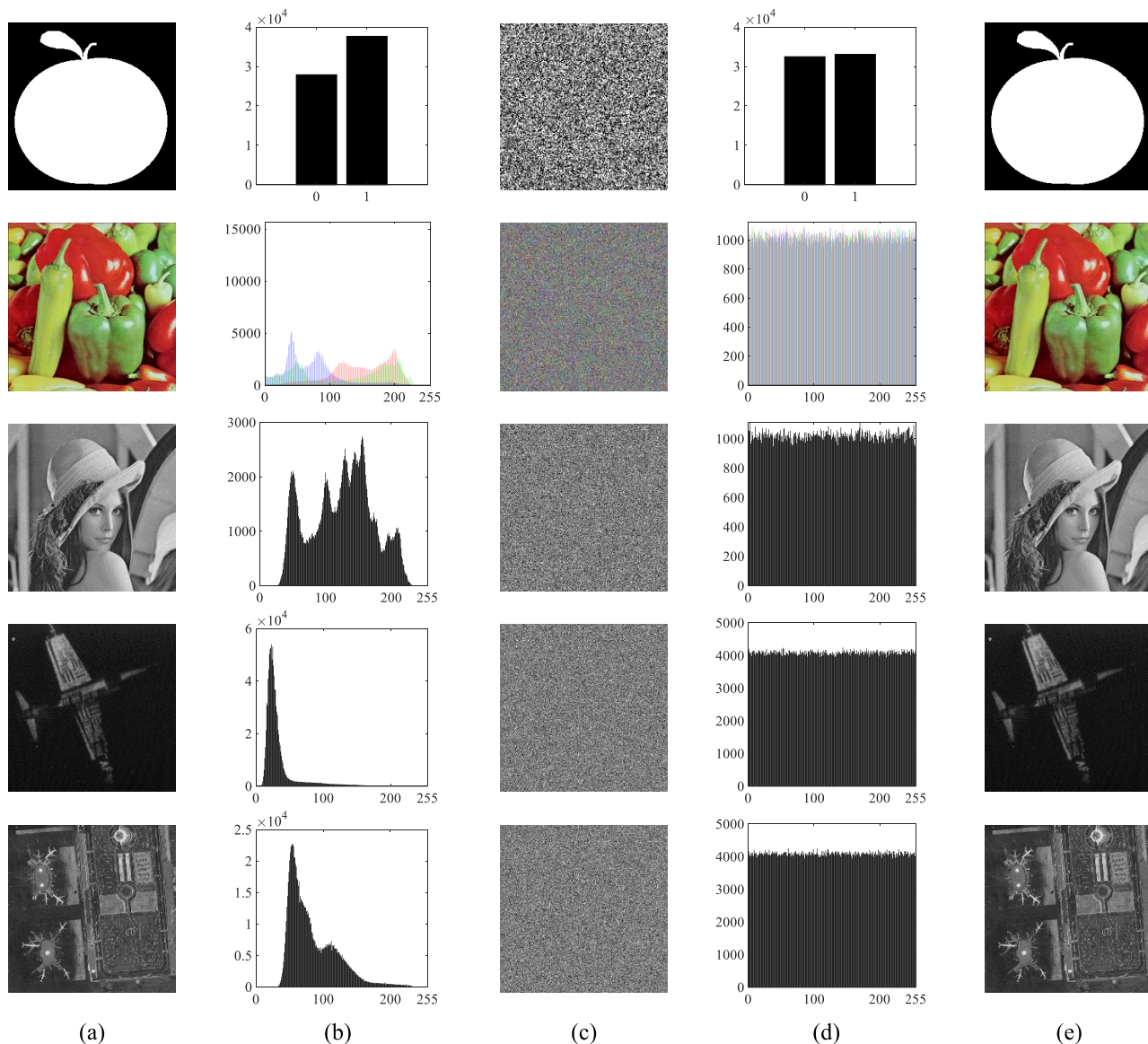


FIGURE 13. Histogram of images with different type: (a) plain-images; (b) histogram of plain-images; (c) cipher-images; (d) histogram of plain-images; (e) decrypted images.

capable of resisting differential attacks, namely, the number of pixel changing rate (NPCR) and the unified average changed intensity (UACI). First, select a plain-image P1, and randomly change the pixel value of one bit to obtain P2. C1 and C2 represent the two cipher-images encrypted from P1 and P2, respectively. NPCR and UACI are defined as following,

$$NPCR(C_1, C_2) = \frac{\sum_{i=1}^H \sum_{j=1}^W D(i, j)}{H \times W} \times 100\% \tag{11}$$

$$UACI(C_1, C_2) = \frac{1}{H \times W} \left[\sum_{i=1}^H \sum_{j=1}^W \frac{|C_1(i, j) - C_2(i, j)|}{L} \right] \times 100\% \tag{12}$$

where H and W represent the length and width of the cipher-image, i and j denote the position of the pixel value, L is the gray level of the image, and

$$D(i, j) = \begin{cases} 0, & C_1(i, j) = C_2(i, j) \\ 1, & C_1(i, j) \neq C_2(i, j) \end{cases} \tag{13}$$

Theoretically, as mentioned in [42], the ideal values of NPCR and UACI that are expected to resist differential attacks are expressed as following,

$$NPCR_* = \left(1 - \frac{1}{2^{\log_2 L}} \right) \times 100\% \tag{14}$$

$$UACI_* = \frac{1}{L^2} \left[\sum_{i=1}^{L-1} \frac{i(i+1)}{L-1} \right] \times 100\% \tag{15}$$

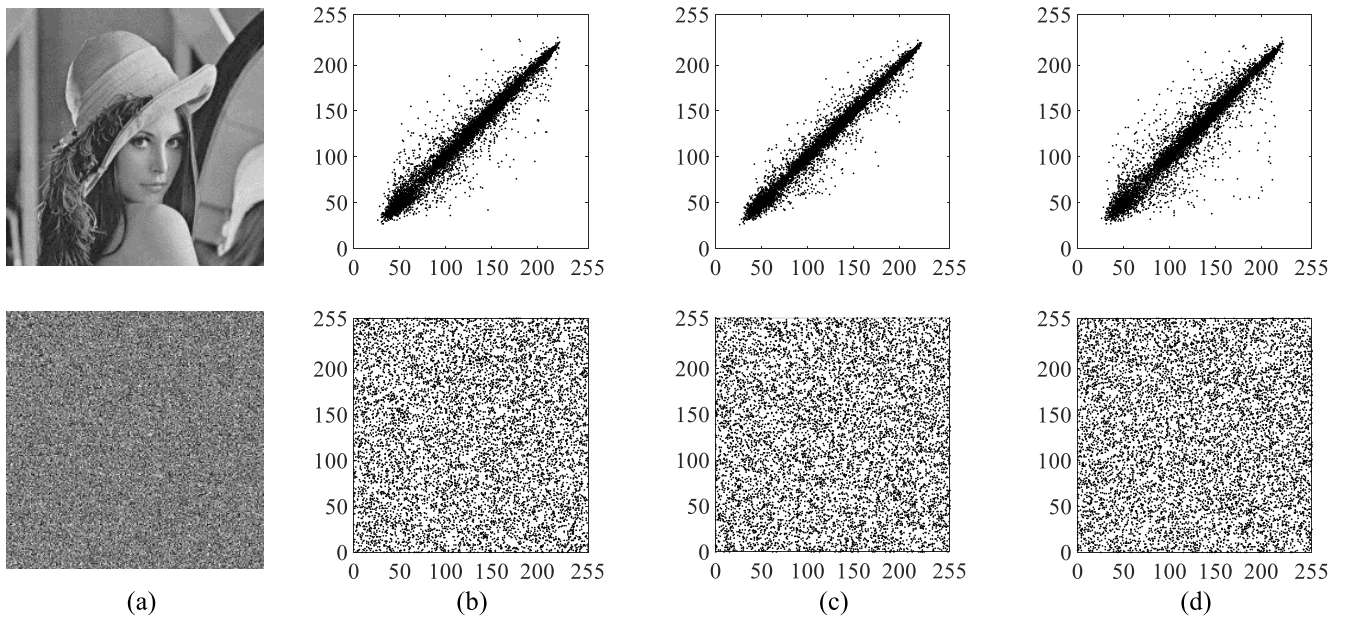


FIGURE 14. Correlation coefficient diagram of Lena grayscale image: (a) The image to be tested; (b) horizontal direction; (c) vertical direction; (d) diagonal direction.

When the NPCR and UACI values are closer to the theoretical values, it indicates that they have better resistance to differential attacks. But there is no standard to determine the range of NPCR and UACI values. So we use the results in [43] to verify the 2D-LJTM IEA. As mentioned in [43] that the critical values of NPCR and UACI are N_α and (u_α^-, u_α^+) , according to different image size and significance level α will have different critical values. If the NPCR and UACI values that we detected are less than N_α or not within the range of (u_α^-, u_α^+) , it means that the ciphertext lacks the ability to resist differential attacks. From Table 6, we can see that the NPCR and UACI average values of the USC-SIPI ‘Miscellaneous’ dataset are 99.6070% and 33.4795%, which are only 0.0024% and 0.016% away from the theoretical value mentioned in [43]. And according to [44], when the significance level α is 0.05, the NPCR and UACI values we verified are all within the critical value range.

F. HANNON ENTROPY ANALYSIS

Shannon entropy analysis can be divided into global Shannon entropy [2] and local Shannon entropy [45]. Both can be expressed as the randomness of pixel value distribution for an image. Based on the above characteristics, the randomness level of an image encryption system can be calculated and analyzed through Shannon entropy. The global Shannon entropy $H(s)$ is defined as (16),

$$H(s) = - \sum_{i=1}^L P(s_i) \log_2 P(s_i) \quad (16)$$

where $\{s_1, s_2, \dots, s_L\}$ represents the set of all elements $s_i \in S$, L represents the gray level of the image, and $P(s_i)$ represents the probability of occurrence of $S = s_i$. For an 8-bit

TABLE 6. NPCR and UACI results.

Image	Size	NPCR	UACI	Results
5.1.09	256×256	99.5850	33.5878	Pass
5.1.10	256×256	99.5957	33.5712	Pass
5.1.11	256×256	99.5956	33.5967	Pass
5.1.12	256×256	99.5987	33.4818	Pass
5.1.13	256×256	99.6185	33.4216	Pass
5.1.14	256×256	99.6292	33.5929	Pass
5.2.08	512×512	99.6101	33.4385	Pass
5.2.09	512×512	99.6037	33.4649	Pass
5.2.10	512×512	99.6067	33.4255	Pass
7.1.01	512×512	99.6078	33.4787	Pass
7.1.02	512×512	99.5999	33.4687	Pass
7.1.03	512×512	99.6223	33.4955	Pass
7.1.04	512×512	99.5930	33.4279	Pass
7.1.05	512×512	99.6090	33.4865	Pass
7.1.06	512×512	99.6150	33.4678	Pass
7.1.07	512×512	99.6037	33.4836	Pass
7.1.08	512×512	99.6094	33.4621	Pass
7.1.09	512×512	99.6078	33.4209	Pass
7.1.10	512×512	99.6090	33.3713	Pass
boat.512	512×512	99.6131	33.4017	Pass
gray21.512	512×512	99.6090	33.5086	Pass
ruler.512	512×512	99.6170	33.4941	Pass
5.3.01	1024×1024	99.6025	33.4805	Pass
5.3.02	1024×1024	99.6048	33.4956	Pass
7.2.01	1024×1024	99.6097	33.4620	Pass

grayscale image, there are a total of 2^8 possible states for the pixel value, when the number of all states are the same, the theoretical Shannon entropy is 8.

TABLE 7. Local Shannon entropy results.

Image	2D-LJTM IEA	Ref. [11]	Ref. [12]	Ref. [24]	Ref. [42]
5.1.09	7.9029823	7.9019537	7.9026161	7.9015718	7.9035711
5.1.10	7.9028661	7.9026794	7.9020332	7.9046965	7.9033718
5.1.11	7.9021433	7.9027762	7.9023787	7.9043636	7.9003172
5.1.12	7.9029545	7.9020605	7.9028313	7.9048929	7.9027386
5.1.13	7.9029192	7.9026766	7.9020172	7.9037709	7.9004188
5.1.14	7.9019015	7.9026756	7.9024421	7.9015737	7.9026265
5.2.08	7.9020071	7.9021010	7.8998169	7.9046512	7.9043155
5.2.09	7.9015445	7.9029031	7.9021513	7.9022999	7.9036303
5.2.10	7.9023297	7.9032474	7.9009431	7.9014831	7.9024444
7.1.01	7.9026800	7.9005814	7.9027466	7.9049225	7.9021978
7.1.02	7.9023413	7.9042115	7.8999371	7.9037566	7.9029369
7.1.03	7.9019626	7.9036169	7.9039913	7.9039016	7.9022684
7.1.04	7.9026431	7.9005136	7.9017415	7.9009347	7.9009099
7.1.05	7.9023706	7.9000206	7.9021942	7.9041305	7.9065666
7.1.06	7.9023613	7.9023032	7.9019996	7.9042892	7.9043870
7.1.07	7.9028997	7.9032941	7.9021436	7.9046288	7.9026480
7.1.08	7.9025984	7.9010992	7.9020712	7.9009333	7.9019985
7.1.09	7.9030373	7.9051132	7.9009463	7.9030724	7.9006977
7.1.10	7.9023894	7.9022365	7.9033234	7.9023371	7.9050126
boat.512	7.9021831	7.9023457	7.9023686	7.8997919	7.9035797
gray21.512	7.9021423	7.9018044	7.8989296	7.9024124	7.9028993
ruler.512	7.9023842	7.9010162	7.9029163	7.9014077	7.9040475
5.3.01	7.9023496	7.9034853	7.9029481	7.9047570	7.9032011
5.3.02	7.9022285	7.9012132	7.9024863	7.9014788	7.9029093
7.2.01	7.9029372	7.9016075	7.9021532	7.9056649	7.9042684
Mean	7.9024463	7.9023014	7.9020051	7.9031089	7.9029585
Std	0.0003953	0.0011932	0.0011300	0.0016271	0.0014534
Pass/All	24/25	11/25	17/25	3/25	10/25

However, there are some defects in the global Shannon entropy. For example, there is an unfair random comparison between images of different sizes, or when the number of image pixel values appear on average, but the distribution is not random. Thus we use the local Shannon entropy proposed in [45] to overcome problems in the global Shannon entropy. We need to randomly select k non-overlapping blocks in the ciphertext, where each block contains q pixel values, and calculate the average Shannon entropy of each non-overlapping block. The local Shannon entropy is defined as (17),

$$\bar{H}_{k,q}(S) = \sum_{i=1}^k \frac{H(s_i)}{k} \quad (17)$$

where $H(s_i)$ is global Shannon entropy. In this paper, we will randomly select 30 non-overlapping blocks, each block size is 44×44 , so that $(k, q) = (30, 1936)$. When the significance level α is 0.05, the ideal local Shannon entropy is 7.902469317, and when the local Shannon entropy falls within the interval (7.901901305, 7.903037329), it is considered to pass the test. Table 7 shows the local Shannon entropy of 8-bit grayscale images using the ‘Miscellaneous’ dataset in USI-SIPI for 2D-LJTM IEA and compares with [11], [12],

[24], and [42]. It can be seen that 2D-LJTM IEA has the highest local Shannon entropy pass rate, and the average value is 7.9024463, which is the closest to the ideal value, which means that 2D-LJTM IEA can encrypt different images into cipher-images with the highest randomness.

G. ROBUSTNESS ANALYSIS

In modern life, image information is often transmitted through the Internet or other media. During the transmission process, it is prone to noise interference or data loss, or attacker illegally obtained the authorization of the ciphertext and forged some image information to destroy the encryption and decryption process. Therefore, robustness analysis plays an important role in the security analysis of image encryption algorithms. Fig. 15 shows the results of decrypting the plain-image when the cipher-image has 6.25%, 25%, and 50% data loss, respectively. We can see that when the cipher-image has different sizes of data loss or noise interference, 2D-LJTM IEA can still restore to the plain-image. Although the decrypted image will have some noise according to the degree of data loss, most of the information in the plain-image can still be identified. This means that

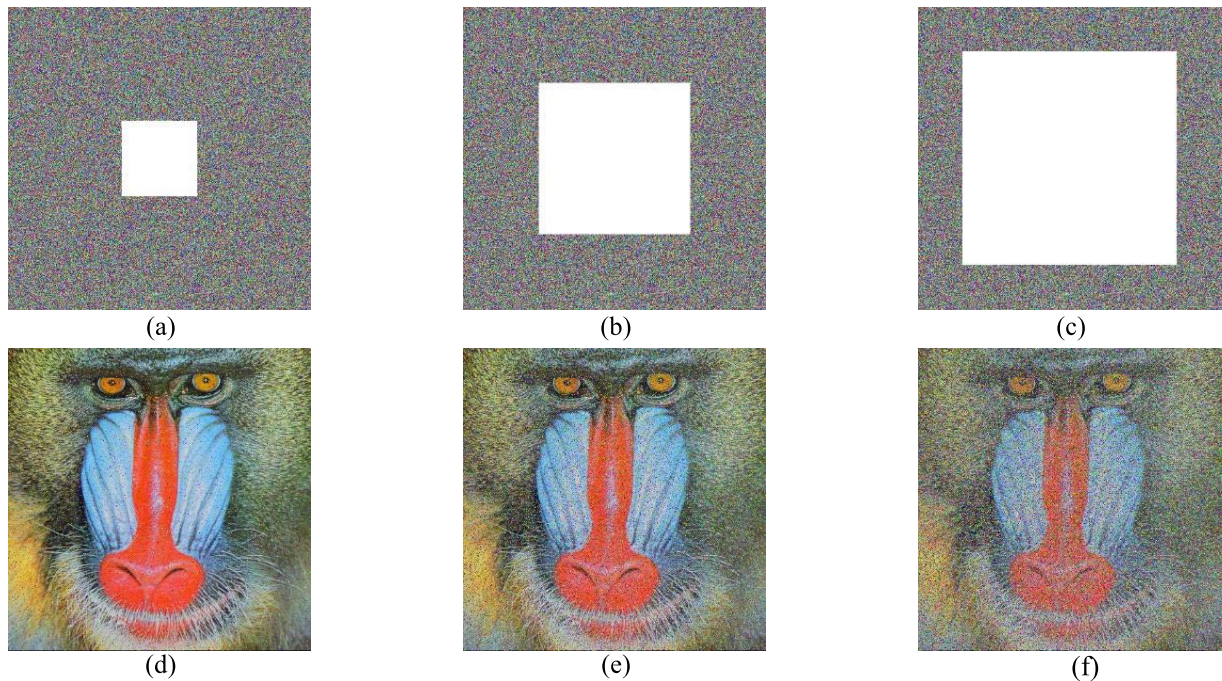


FIGURE 15. Robustness analysis (Baboon): (a), (b) and (c) are cipher-images with different sizes of data loss; (d), (e) and (f) are the effects of (a), (b) and (c) decrypted by 2D-LJTM IEA.

2D-LJTM IEA can evenly diffuse and distribute the pixel values in the plain-image to the cipher-image. So that even though there is data loss in the cipher-image, the data can also be successfully decrypted.

VI. CONCLUSION AND FUTURE WORK

In this paper, we design a novel two-dimensional hyperchaotic system 2D-LJTM for the region of interest image encryption. We use bifurcation diagram, Lyapunov exponent and NIST SP 800 test to verify its chaotic characteristics, we can find that 2D-LJTM exhibits a high degree of randomness in all tests. And we use two features on the plain-image to generate the key generator, and finally add SHA-256 to enhance the key sensitivity. Then in the encryption algorithm, we first proposed the Bagua coding architecture in this paper, which is a double coding form coupled with permutation step, which can make the plaintext completely concealed. At the end of the encryption algorithm, we use two diffusion methods, the first is the advanced exclusive-or operation, and the second is the pixel value bit shift operation to enhance the security of the encrypted ciphertext image. The above encryption algorithms are all based on 2D-LJTM.

In order to improve the efficiency of encryption, reduce the storage space spent in calculations, and different message senders and receivers have different areas of encryption and decryption that they want to encrypt and decrypt, we propose the region of interest image encryption. Using YoloV3 and UNet can automatically identify and extract the region to be encrypted, and the selected region can be irregular shape, then

encrypted by 2D-LJTM IEA. Finally, we use seven security analyses to verify 2D-LJTM IEA, and from the results, we can know that the encryption algorithm we designed based on the novel two-dimensional hyperchaotic system has sufficient security.

Bagua coding is the coding concept proposed for the first time in this paper, since it is the first time it appears, the details of this coding have not yet been optimized. For example, in the step of Bagua encoding, every three bits must be used as a unit for encoding. For an image with an 8-bit grayscale image, the pixel value must be expanded to a multiple of three to facilitate encoding. In the future, it is possible to increase the efficiency of Bagua coding by encoding multiple pixel values in one block, or to re-plan the Bagua-level permutation rule to improve while maintaining a certain degree of complexity.

ACKNOWLEDGMENT

The concept of the encryption algorithm is based on part of 2D logistic-modulated-sine-coupling-logistic chaotic map for image encryption of Hegui Zhu. Last, the application of the ROI encryption is based on part of Cy: Chaotic yolo for user intended image encryption and sharing in social media of Meysam Asgari-Chenaghlu.

REFERENCES

- [1] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, Aug. 2006.
- [2] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

- [3] A. S. Wightman, "New results in qualitative dynamics: Problèmes ergodiques de la mécanique classique. V. I. Arnold and A. Avez. Gauthier-Villars, Paris, 1967. IV + 243 pp., illus. Paper, 48 F. Monographies internationales de mathématiques modernes," *Science*, vol. 159, no. 3821, p. 1344, Mar. 1968.
- [4] R. Matthews, "On the derivation of a 'chaotic' encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, Jan. 1989.
- [5] T. Yang, C. W. Wu, and L. O. Chua, "Cryptography based on chaotic systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 5, pp. 469–472, May 1997.
- [6] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, Jun. 1998.
- [7] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, no. 3, pp. 746–761, Jul. 2004.
- [8] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, 2006.
- [9] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Process.*, vol. 92, no. 4, pp. 1101–1108, Apr. 2012.
- [10] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, no. 11, pp. 172–182, 2014.
- [11] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D sine logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.
- [12] H. Zhu, Y. Zhao, and Y. Song, "2D logistic-modulated-sine-coupling-logistic chaotic map for image encryption," *IEEE Access*, vol. 7, pp. 14081–14098, 2019.
- [13] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, Nov. 1994.
- [14] X. Zheng, J. Xu, and W. Li, "Parallel DNA arithmetic operation based on n -moduli set," *Appl. Math. Comput.*, vol. 212, no. 1, pp. 177–184, Jun. 2009.
- [15] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft. Comput.*, vol. 12, no. 5, pp. 1457–1466, 2012.
- [16] J. Wu, X. Liao, and B. Yang, "Image encryption using 2D Hénon-Sine map and DNA approach," *Signal Process.*, vol. 153, pp. 11–23, Dec. 2018.
- [17] Y.-Q. Zhang, X.-Y. Wang, J. Liu, and Z.-L. Chi, "An image encryption scheme based on the MLNCML system using DNA sequences," *Opt. Lasers Eng.*, vol. 82, pp. 95–103, Jul. 2016.
- [18] X. Y. Wang, H. L. Zhang, and X. M. Bao, "Color image encryption scheme using CML and DNA sequence operations," *Biosystems*, vol. 144, pp. 16–28, Jun. 2016.
- [19] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption," *Signal Process.*, vol. 142, pp. 340–353, Jan. 2018.
- [20] L. Liu, D. Wang, and Y. Lei, "An image encryption scheme based on hyperchaotic system and DNA with fixed secret keys," *IEEE Access*, vol. 8, pp. 46400–46416, 2020.
- [21] D. Xiao, Q. Fu, T. Xiang, and Y. Zhang, "Chaotic image encryption of regions of interest," *Int. J. Bifurcation Chaos*, vol. 26, no. 11, Oct. 2016, Art. no. 1650193.
- [22] H.-W. Xue, J. Du, S.-L. Li, and W.-J. Ma, "Region of interest encryption for color images based on a hyperchaotic system with three positive Lyapunov exponents," *Opt. Laser Technol.*, vol. 106, pp. 506–516, Oct. 2018.
- [23] M. Asgari-Chenaghlu, M.-R. Feizi-Derakhshi, N. Nikzad-Khasmaki, A.-R. Feizi-Derakhshi, M. Ramezani, Z. Jahanbakhsh-Nagadeh, T. Rahkar-Farshi, E. Zafarani-Moattar, M. Ranjbar-Khadivi, and M.-A. Balafar, "C_y: Chaotic Yolo for user intended image encryption and sharing in social media," *Inf. Sci.*, vol. 542, pp. 212–227, Jan. 2021.
- [24] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.
- [25] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Process.*, vol. 149, pp. 148–161, Aug. 2018.
- [26] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Inf. Sci.*, vol. 546, pp. 1063–1083, Feb. 2021.
- [27] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, pp. 459–467, Jun. 1976.
- [28] H. F. von Bremen, F. E. Udawadia, and W. Proskurowski, "An efficient QR based method for the computation of Lyapunov exponents," *Phys. D, Nonlinear Phenomena*, vol. 101, nos. 1–2, pp. 1–16, Feb. 1997.
- [29] A. Rukhin *et al.*, "A statistical test suite for random and pseudo-random number generators for cryptographic applications," NIST Special Publication 800-22, Apr. 2010. [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final>
- [30] X. Chai, Y. Chen, and L. Brody, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.
- [31] G. Tang, X. Liao, and Y. Chen, "A novel method for designing S-boxes based on chaotic maps," *Chaos, Solitons Fractals*, vol. 23, no. 2, pp. 413–419, 2005.
- [32] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation-substitution-based encryption," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 337–361, Jan. 2017.
- [33] D. Xiao, X. Liao, and S. Deng, "One-way hash function construction based on the chaotic map with changeable-parameter," *Chaos, Solitons Fractals*, vol. 24, no. 1, pp. 65–71, Apr. 2005.
- [34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [35] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
- [36] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [37] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. Cham, Switzerland: Springer*, Oct. 2015, pp. 234–241.
- [38] Y. Zhang, D. Xiao, W. Wen, and M. Li, "Breaking an image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Nonlinear Dyn.*, vol. 76, no. 3, pp. 1645–1650, May 2014.
- [39] H. Kaur and N. Sohi, "A study for applications of histogram in image enhancement," *Int. J. Eng. Sci.*, vol. 6, no. 6, pp. 59–63, Jun. 2017.
- [40] Y. Yang, L. Wang, S. Duan, and L. Luo, "Dynamical analysis and image encryption application of a novel memristive hyperchaotic system," *Opt. Laser Technol.*, vol. 133, Jan. 2021, Art. no. 106553.
- [41] A. G. Asuero, A. Sayago, and A. González, "The correlation coefficient: An overview," *Crit. Rev. Anal. Chem.*, vol. 36, no. 1, pp. 41–59, 2006.
- [42] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "An image encryption algorithm based on compound homogeneous hyper-chaotic system," *Nonlinear Dyn.*, vol. 89, no. 1, pp. 61–79, Jul. 2017.
- [43] C. Fu, J.-J. Chen, H. Zou, W.-H. Meng, Y.-F. Zhan, and Y.-W. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Opt. Exp.*, vol. 20, no. 3, pp. 2363–2378, 2012.
- [44] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Select. Areas Telecomm.*, vol. 1, pp. 31–38, Apr. 2011.
- [45] Y. Wu, Y. Zhou, G. Saveriadis, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Inf. Sci.*, vol. 222, pp. 323–342, Feb. 2013.

•••