Colour image encryption based on logistic mapping and double random-phase encoding

Huiqing Huang\textsuperscript{1,2}, Shouzhi Yang\textsuperscript{1} \E-mail: szyang@stu.edu.cn

\textsuperscript{1}Department of Mathematics, Shantou University, Shantou 515063, Guangdong, People’s Republic of China
\textsuperscript{2}School of Mathematics, Jiaying University, Meizhou 514015, Guangdong, People’s Republic of China

Abstract: In this study, the authors propose a novel method to encrypt a colour image by use of Logistic mapping and double random-phase encoding. Firstly, use Logistic mapping to diffuse the colour image, then the red, green and blue components of the result are scrambled by replacement matrices generated by Logistic mapping. Secondly, by utilising double random-phase encoding to encrypt the three scrambled images into one encrypted image. Experiment results reveal the fact that the proposed method not only can achieve good encryption result, but also that the key space is large enough to resist against common attack.

1 Introduction

With the rapid development of networks, people’s economic life is increasingly dependent on the network, so network security has become increasingly important in recent years. Data communications have become largely networking in nature with imaging capabilities being embedded in a myriad of portable devices, such as smart phones and tablet computers. Simultaneously, communication channels, such as internet and wireless networks, these techniques bring great convenience to our life, but it also brings new challenges to the privacy. To meet this challenge, various encryption algorithms have been designed [1–9]. Such as image encryption based on pixel permutation, image encryption based on double random-phase encoding, image encryption technology based on modern cryptography [10, 11], image encryption technology based on chaotic and so on. Among them, due to chaotic have shown some exceptionally good properties that are desirable for the cipher system, pseudo-randomness, extremely sensitive for initial value, ergodicity and a period and so on, more and more researchers pay attention to chaotic encryption algorithm. In [12], Socek et al. proposed an enhanced one-dimensional (1D) chaotic key-based algorithm for image encryption. A novel block cryptosystem based on iterating a chaotic map has been proposed in [13]. Later in 2007, Li et al. [14] pointed out the encryption scheme presented in [13] is not only insecure against chosen-plaintext attack, but also insecure against a differential known-plaintext attack. Subsequently, a new block based image shuffling is proposed to achieve good shuffling effect using two chaotic maps and the encryption of the shuffled image is performed using a third chaotic map to enforce the security of the proposed encryption process [15].

In recent years, plenty of colour image encryption approaches have been proposed. In some encryption algorithms, colour image is decomposed into the components of red, green and blue (RGB). The corresponding colour components are encoded by an encryption method of grey-level image. The RGB components of colour image are encrypted by chaotic systems. Chaos-based image encryption algorithm to encrypt colour images by using coupled non-linear chaotic map [16] has been reported. In [17], Liu et al. proposed a colour image encryption by using the rotation of colour vector in Hartley transform domains. Later in 2011, Liu et al. [18] presented a colour image encryption algorithm is designed by use of Arnold transform and discrete cosine transform. Subsequently, Chen et al. have reported a colour image encryption based on the affine transform and the gyrator transform [19]. Liu et al. [20] proposed a novel colour image hiding scheme with three channels of cascaded Fresnel domain phase-only filtering.

In this paper, we propose a colour image encryption by use of Logistic mapping and phase encoding. Firstly, diffuse the colour image by using Logistic mapping, then the RGB components of the result are scrambled and encoded by Logistic mapping and double random-phase encoding. Moreover, finally the numerical results are given to illustrate the feasibility and effectiveness of the proposed algorithm.

2 Preliminaries for proposed technique

2.1 Logistic chaotic mapping

Logistic mapping is also known as pest model, it is a typical mathematical model for describing the evolution of species, which can be shown with a non-linear repeated equation as follows [21]:

\[ x_{k+1} = \mu x_k (1 - x_k), \]

where \( \mu \) is the branch parameter, and \( x_k \in (0, 1) \). Logistic mapping is in chaotic state when \( \mu \in (3.5699456, 4) \) [22]. Logistic mapping can be generalised to the 2D logistic mapping, which is defined as [23]

\[
\begin{align*}
\dot{x}_{k+1} &= x_k + \lambda (x_k - x_k^2 + y_k), \\
\dot{y}_{k+1} &= y_k + \lambda (y_k - y_k^2 + x_k),
\end{align*}
\]

When \( 0.6 < \lambda \leq 0.686 \), the system of (2) is in chaotic state. The bifurcation graph shown in Fig. 1 with the initial point (0.4, 0.5) and \( 0.49 \leq \lambda \leq 0.686 \). For more detailed analysis of the complex dynamics of the system, please see relative reference [23].

2.2 Double random-phase encoding

In 1995, Refregier and Javidi [24] proposed the double random-phase encoding technique. The encoded image is obtained by random-phase encoding in both the input and the Fourier planes. If two random-phase masks are used to encrypt the image in the input and Fourier planes, respectively, the input image is transformed into a complex-amplitude stationary white noise.

Let \( f(x, y) \) denote the image to be encoded and \( g(x, y) \) denote the encoded image. Let \( \phi(x, y) \) and \( \psi(u, v) \) denote two independent white sequences uniformly distributed in \([0; 1]\), \((x, y)\) and \((u, v)\) respectively.
This section presents the proposed scheme for colour image encryption by using Logistic and phase encoding. Assume that the vector $H = \{x_i, y_i, z_i\}$ denote the spatial plane and the Fourier plane coordinates, respectively. The encoding and decoding procedures are shown as follows:

$$
\begin{align*}
    g(x, y) &= \mathcal{F}^{-1}\{\mathcal{F}\{f(x, y)\exp[2\pi i \phi(x, y)]\}\exp[2\pi i \phi(u, v)]\}, \\
    f(x, y) &= \mathcal{F}^{-1}\{\mathcal{F}\{g(x, y)\exp[-2\pi i \phi(u, v)]\}\exp[-2\pi i \phi(u, v)]\},
\end{align*}
$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Fourier transform and the inverse Fourier transform, respectively.

3 Proposed colour image encryption scheme

This section presents the proposed scheme for colour image encryption by using Logistic and phase encoding. Assume that the size of original colour image $I$ is $N \times N$. The schematic diagram of the proposed encryption is illustrated in Fig. 2, and the proposed image encryption algorithm consists of the following steps:

Step 1: Set the values of $x_0, y_0$, and $\lambda$.

Step 2: For the size $(N \times N)$ of the colour image, after iterating the Logistic mapping of (2) for $L$ times, we iterate the Logistic mapping continuously. For each iteration, we can get two values $x_i$ and $y_i$. These decimal values are preprocessed first to get four decimal sequences $U, V, W,$ and $Z$, as shown in follows:

$$
\begin{align*}
    U &= \{u_1, u_2, \ldots, u_{2^N}\}, \\
    V &= \{v_1, v_2, \ldots, v_{2^N}\}, \\
    W &= \{w_1, w_2, \ldots, w_{2^N}\}, \\
    Z &= \{z_1, z_2, \ldots, z_{2^N}\},
\end{align*}
$$

where

$$
\begin{align*}
    u_i &= 10^2 x_i - \text{floor}(10^2 x_i), \\
    v_i &= 10^3 y_i - \text{floor}(10^3 y_i), \\
    w_i &= \text{floor}(10^4 x_i) - \text{floor}(10^4 x_i)(10^4) \mod 256, \\
    z_i &= \text{floor}(10^4 y_i) - \text{floor}(10^4 y_i)(10^4) \mod 256.
\end{align*}
$$

Here floor(x) returns the nearest integer less than or equal to x, and mod returns the remainder after division.

Step 3: By taking $N$ successive elements of sequence $U$ or $V$, a vector $H = (u_{i_1}, \ldots, u_{i_k})$ is obtained, where $0 \leq k \leq 3N^2 - N$. Then we reorder vector $H$ from small to large order to get new vector $H’$. Via $H$ and $H’$ we can construct a scrambling matrix $A$, which satisfy

$$
H = H'A. 
$$

So, by this way we can construct six scrambling matrices $A_x, A_y, A_z, A_{x'}, A_{y'},$ and $A_{z'}$.

Step 4: By taking $N \times N$ successive elements of sequence $U$, and we convert it into a random matrix $Q_x$ of size $N \times N$. Similarly, we can obtain another random matrix $Q_y$ by using sequence $V$.

Step 5: Let $M_x(x, y) = \exp(2\pi i Q_x)$ and $M_y(x, y) = \exp(2\pi i Q_y)$ present the two random phase masks.

Step 6: Reshape the sequences $W$ and $Z$, respectively, we obtain the 3D matrices $Y$ and $J$ with the same size as the original image, then we use the matrices $Y$ and $J$ to diffuse the colour image. This is mathematically represented as follows:

$$
F(i, j, k) = (\{I(i, j, k) + Y(i, j, k)\mod 256\} \oplus J(i, j, k)).
$$

Here $i, j = 1, 2, \ldots, 256$ and $k = 1, 2, 3$. The symbol $\oplus$ represents the exclusive OR operation bit-by-bit.

Step 7: The matrix $F$ is converted into its RGB components. Afterwards, each colours matrix $(R, G$ or $B)$ is scrambled by scrambling matrices $[A_i]i = 1, 2, \ldots, 6$. This is mathematically represented as follows:

$$
\begin{align*}
    R' &= A_x \times R \times A_z, \\
    G' &= A_x \times G \times A_y, \\
    B' &= A_x \times B \times A_y,
\end{align*}
$$

To enhance the security, we can perform more rounds this step, i.e. multiple scramble. In this paper, we take five rounds.

Step 8: The normalisation is processed to the matrices $R’, G’,$ and $B'$, and we can obtain $R’, G’,$ and $B’$ respectively.

Step 9: Combine $R'(x, y)$ and $G'(x, y)$ to obtain the complex amplitude image

$$
C(x, y) = R'(x, y)\exp(2\pi i G'(x, y)).
$$

Step 10: $C(x, y)$ is first multiplied by the first random phase mask $M_x(x, y)$, then transformed by fast Fourier transform (FFT), the amplitude part $B_x(x, y)$ and phase part $K_x(x, y)$ of the result after FFT can be represented as, respectively,

$$
\begin{align*}
    B_x(x, y) &= \text{PT}[\mathcal{F}[C(x, y)M_x(x, y)]], \\
    K_x(x, y) &= \text{AT}[\mathcal{F}[C(x, y)M_x(x, y)]],
\end{align*}
$$

among them, $\text{PT}\{\}$ denotes the extracting amplitude part operator, $\text{AT}\{\}$ denotes the extracting phase part operator, and $\mathcal{F}\{\}$ denotes the FFT.

Step 11: Except replace FFT with IFFT in step 10, repeat steps 9 and 10 for $B_x(x, y), M_x(x, y)$, and $B_y(x, y)$, we can obtain $B_z(x, y)$ and $K_z(x, y)$. Then $K_y, K_z, B_x$ are converted into a colour image $E$. 

![Bifurcation graph of system (2) when $\lambda \in [0.49, 0.686]$](image1.png)

Fig. 1 Bifurcation graph of system (2) when $\lambda \in [0.49, 0.686]$
In the encryption process, E is saved for encrypted image. Moreover, the decryption is the reverse process of the encryption.

4 Numerical simulation and discussion

Some numerical simulations are performed to verify the proposed encryption algorithm for one image. In the numerical simulations, colour image of Lena having a size of $256 \times 256$ pixels is shown in Fig. 3a and serves as input original image. For convenience, the encryption key $(\lambda, x_0, y_0, L)$ is fixed at $(0.65, 1, 0.01, 20,000)$, and it also is the decryption key. Fig. 3b shows the encrypted output for the original colour image. It is observed that the encryption images are completely unintelligible, and does not reveal any information about the original colour image. Fig. 3c shows the decrypted image with correct keys.

4.1 Key space and sensitivity analysis

It is well known that high sensitive to initial conditions are inherent to any chaotic system. A good image encryption algorithm should be sensitive to the cipher keys, and the key space should be large enough to make any brute-force attacks infeasible.

Fig. 4 shows the decrypted image with a tiny change of the keys. Fig. 4a shows the decrypted image with incorrect independent parameter $x_0 = 1 + 10^{-14}$, while the other keys are all correct. Fig. 4b shows the decrypted image with incorrect independent parameter $y_0 = 0.01 + 10^{-14}$, while the other keys are all correct. This is observed that even the tiny change of $10^{-14}$, the decrypted image is absolutely different from the plain image. Fig. 4c shows the decrypted image with an incorrect independent parameter $\lambda = 0.65 + 10^{-16}$, while the other keys are all correct. So the key spaces for $x_0$, $y_0$, and $\lambda$ are $S_{x_0} = S_{y_0} \approx 10^{14}$, $S_{\lambda} \approx 10^{16}$.

The iteration times $L$ are 16-bit integers, so $S_L = 2^{16}$. The total key space $S = S_{x_0}S_{y_0}S_{\lambda}S_L \approx 6.5536 \times 10^{48}$. While advanced encryption standard (AES) is an acknowledged secure encryption algorithm and the key space of the 128-bit AES algorithm is about $2^{256}$. So it can be seen that the proposed colour image encryption algorithm is good at resisting brute-force attack. The key space of
the proposed algorithm and the algorithms in [4, 7, 12] are compiled in Table 1, which shows that the proposed algorithm has larger key space than them.

The mean square error (MSE) between decrypted image and original image is an important factor to evaluate the key sensitivity of an image encryption algorithm

\[
MSE = \frac{1}{M \times N \times 3} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} (I(i, j, k) - H(i, j, k))^2,
\]

where \(I(i, j, k)\) and \(H(i, j, k)\) mean the pixel values at point \((i, j, k)\) of original colour image and decrypted image, respectively. \(M \times N\) denotes the colour image size. The MSE curves for \(x_0\) and \(L\) are computed and shown in Figs. 5a and b. The MSE is very large with a little deviation to the correct keys and the MSE is very small only when the main keys are correct. Thus the decrypted image can be recognised if and only if the keys are correct.

### 4.2 Statistical analysis

By Marion [25] we know the statistical attack is launched by exploiting the predictable relationship between data segments of the original and the encrypted image. Hence to demonstrating the proposed colour image encryption having strongly resisted statistical attacks, we test on the histograms of the enciphered images and on the correlations of adjacent pixels in the ciphered image.

Histograms of RGB colours for the original and the encrypted images are indicated in Figs. 6a–f and 7a–f, respectively. It is observed that the histogram of the encrypted image and histogram of the original image significantly different, it illustrates non-existent correlation between the two images.

It is well known that adjacent image pixels are highly correlated either in horizontal, vertical or diagonal directions. To test the correlations of adjacent pixels in original and encrypted images, randomly select 4000 pairs of two adjacent pixels (in horizontal, vertical, and diagonal directions) from an image. Then, the correlation coefficient can be calculated by using the following formulas [16]:

\[
r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{\text{D}(x) \text{D}(y)}}
\]

\[
E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

\[
D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2
\]

\[
\text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))
\]

where \(x\) and \(y\) denote two adjacent pixels and \(N\) is the total number of duplets \((x, y)\) obtained from the image.

The results of correlation analysis are shown in Table 2. The result indicates that the correlation of two adjacent pixels of the original image is close to 1 in each direction of each component, while that of the encrypted image is close to 0 in each direction of each component, so the encryption effect is rather good.

### 4.3 Robustness analysis

In this section, we focus on analysing performance of the proposed algorithm in noisy and lossy. In the first test, we check the
tolerance against the loss of encrypted data. We occlude 1.5625, 25 and 50% of the encrypted image pixels. Figs. 8a–c indicate the occluded encrypted images of Fig. 3b, whereas Figs. 8d–f illustrate the decrypted output for these images. As we can see, in the case of the small percentage pixels of Fig. 3b are occluded, in addition to having some noise, the decryption image and original image pretty much the same, while in the case of large percentage pixels of Fig. 3b are occluded, although only some information about the original image can be retrieved from the decrypted output, the basic outline of the original image can be retrieved.

In the second test, we employ three noises in the simulation, which are random noise, speckle noise and Gaussian noise, respectively. The noise effected encrypted images are then decrypted, output for which is shown in Fig. 9. Fig. 9a shows the recovered image when a random noise is added to the encrypted image of Fig. 3b. Fig. 9b shows the recovered image when speckle noise of variation 0.01 is added to the encrypted image of Fig. 3b. Fig. 9c shows the recovered image when a Gaussian noise with a zero mean and standard deviation of 0.01 is added to the encrypted image of Fig. 3b. As shown in Fig. 9, the recovered image in high noise polluted, but some information about the original image can be retrieved from the decrypted output.

By above tests, it is shown that the presented algorithm can decrypt the approximate image to the original image even if the encrypted image has been damaged. Therefore, the proposed algorithm has good robustness.

4.4 Differential analysis

We have also measured the number of pixels change rate (NPCR) to see the influence of changing a single pixel in the original image on the encrypted image by the proposed algorithm. The NPCR$_{R,G,B}$ measure the number of pixels in difference of a colour component between two images. We take two encrypted images, $C_{R,G,B}$ and $C'_{R,G,B}$, whose corresponding original images have only one-pixel difference. We also define a 2D array $D$, having the same size as the image $C_{R,G,B}(i,j)$ or $C'_{R,G,B}(i,j)$. The $D_{R,G,B}(i,j)$ is determined from $C_{R,G,B}(i,j)$ and $C'_{R,G,B}(i,j)$. If $C_{R,G,B}(i,j) = C'_{R,G,B}(i,j)$ then $D_{R,G,B}(i,j) = 0$ otherwise $D_{R,G,B}(i,j) = 1$. The NPCR$_{R,G,B}$ is evaluated through the following formula [3]:

$$\text{NPCR} = \frac{\sum_{i,j} D_{R,G,B}(i,j) M \times N}{N \times M} \times 100\%,$$

where $N \times M$ is the colour image size.
encoding and pixel scrambling techniques. We use the Logistic and is sensitive to the parameters of Logistic mapping during the result is scrambled and encoded by Logistic mapping and the proposed method is robust against the loss of data.

To test our proposed scheme, each colour component is encrypted first. Then, one similar pixel in each component is randomly selected and toggled. The modified component is encrypted again by using the same keys so as to generate a new cipher-component. Moreover, the expected values are presented in Table 3. From Table 3, it can be found that the percentage of pixels changed in the cipher image is over 99.9985% even with a one-bit difference in the plain-image. Thus, the proposed encryption scheme is able to resist the differential attack.

5 Conclusion

By combining Logistic mapping with double random-phase encoding, a novel colour image encryption algorithm is proposed. To encrypting original colour image involves the XOR, fully phase encoding and pixel scrambling techniques. We use the Logistic mapping to diffuse the colour image, then the RGB components of the result is scrambled and encoded by Logistic mapping and double random phase encoding. The performance is analysed with numerical simulations, it is shown that the key space of the new algorithm is large enough to make brute-force attacks infeasible, and is sensitive to the parameters of Logistic mapping during decryption. Robustness analysis in lossy and noisy demonstrates the proposed method is robust against the loss of data.

6 Acknowledgment

This work was supported by the National Natural Science Foundation of China (grant nos. 11071152, 11601188, 61403164), the Natural Science Foundation of Guangdong Province (grant no. 2015A030313443).

7 References