Low Time-Complexity Chaotic Image Encryption Algorithm Based on Plaintext Information

Nianhang TANG  a,b, 1, Zeming WU  c and Peng GUO  a

a School of Electronic Information and Communications, Huazhong University of Science and Technology, China
b State Key Laboratory of Digital Household Appliances, China
c Electronic Engineering College, Heilongjiang University, China

Abstract. Based on plaintext information, a dynamic chaotic color image encryption algorithm is proposed as a solution for the security problem in image data transmission. Comparatively, the existing one has the characteristics of high key acquisition complexity and low encryption speed. Firstly, normalize the RGB components of the images so as to obtain the dynamic key associated with the plaintext, which encrypts images with any shape by rearranging and combining pixels from the original image. Secondly, the parameter offset key is introduced as the initial parameter of the logistic map to generate a high-precision chaotic encryption sequence. The chaotic value sequence is stored for disordered pixels according to the image size, and enlarged and rounded to enhance the pixel value diffusion of the original image. Finally, RGB components are fully fused and uniformly distributed to form a completely encrypted image. According to the experimental results and security analysis, the proposed algorithm is effective against differential attacks, with low complexity, high efficiency, and faster encryption speed.

Keywords. Chaotic System, Dynamic Keys, Arbitrary image encryption, Plaintext correlation, Low time-complexity

1. Introduction

It is convenient for people to use cloud computing and Internet of Things technology because of its rapid development, for example, people can retrieve the needed information from many images easily and quickly. However, with a large amount of data transmission in the network, privacy information leakage and other security issues, the study of how to protect images and transmit information quickly becomes an important topic. To protect digital image transmission security, it is commonly used as a digital image encryption method, such as traditional RSA and DES encryption algorithms that are to encrypt text information [1, 2]. Nevertheless, the technology doesn’t apply to video and...
image information that contain a lot of data, nor does it guarantee its security [3]. Hence, chaos theory is extensively implemented in image encryption since it is non-periodic, sensitive to initial values, and unpredictable. So far, there have been many encryption results that combine chaos with other algorithms, such as the encryption algorithm that combines a chaotic system with DNA coding in order to make encrypted images more secure [4–6]. Nonetheless, after various studies, it is shown that the performance of encrypting images by DNA coding alone is relatively low and cannot resist selective plaintext attacks. To enhance image processing speed, the image can be compressed before it is encrypted, because compression and encryption are tightly coupled [7, 8], but image encryption algorithms, including the compression step encryption algorithm, are not suitable for schemes that require high data accuracy.

To solve some problems, an image encryption algorithm is proposed based on the above documents, which is highly secure and has a low complexity. In this way, a discrete Logistic chaotic map is designated to make the encryption algorithm more efficient and faster. Image encryption relies on both the plaintext image, as well as the offset key introducing some initial parameters to the logical chaotic map. It is possible to significantly increase the key space of the encryption algorithm without increasing the chaotic system’s size, as well as improve its performance against differential attacks. Scrambling and diffusion of images are performed with the key generated by a chaotic system, which dramatically reduces the number of encryption keys to be obtained and makes the encryption algorithm more efficient. Since the image pixels are rearranged during the encryption process, the proposed encryption algorithm can encrypt images of any shape.

2. Basic theory and encryption algorithm

2.1. Logistic Chaotic Mapping

This section introduces the logical chaotic system used in the paper [9], which is also called an unimodal image. Described by a quadratic polynomial map (recursive relationship), it is often used to demonstrate how chaotic phenomena can derive from nonlinear dynamic equations. According to Eq.(1), logical mapping can be expressed mathematically as follows:

\[ X_{n+1} = \mu X_n (1 - X_n) \]  

(1)

in this case, \( \mu \in [0, 4] \) is called the Logistic parameter. It is shown that when \( X_n \in [0, 1] \), and \( 3.5699 < \mu \leq 4 \), the logical map is in a chaotic state, this means that the sequence generated by the logistic mapping is acyclic and non-convergent. The generated sequence \( \{X_k, k = 0, 1, 2, 3, \cdots\} \) is noncyclic and non-convergent when it is outside this range. Therefore, image encryption requires a more random algorithm, which is sensitive to the initial value.

2.2. Encryption Algorithm

To achieve a better encryption effect and fully use plaintext information and a Logistic chaotic map, a color image chaotic encryption algorithm is proposed in this paper that has low complexity and high security when compared to plaintext encryption. With the
following algorithm steps, the proposed encryption algorithm is able to encrypt any size shape. The encryption process is shown in Figure 1.

Step 1: Extract the RGB component of a color image of an $M \times N (M \neq N)$ size, and convert the extracted RGB to a matrix of three 8-bit binary numbers: $I_1, I_2, I_3$.

Step 2: Normalize the generated matrix to generate three plain-text related keys: $a_1, a_2, a_3$, and these three values are multiplied by each other to form the plain-text key. The key obtained here is the dynamic key. Depending on the read images, the obtained key changes dynamically, i.e., it changes with the image.

Step 3: Convert the matrix generated by the color image into 1 row $M \times N$ column and save it as $IR, IG, IB$.

Step 4: $\mu', a_0'$ are the keys independent of the plaintext of the chaotic system. The logistic mapping is initialized with values and parameters generated by (2) and (3). According to the above conditions, a chaotic order $x$ with a length of $(M \times N \times 3 + \text{floor}(a_3 \times 10000))$ is generated, which will be generated in this step and in the above steps, $\mu', a_0', a_1, a_2, a_3, q$ are transmitted to the decryption end as symmetric keys.

\begin{align*}
\mu_1' + q_1a_1 &= \mu \\
a_0' + q_2a_2 &= a_0
\end{align*}

Step 5: For the sequence $x$ generated after step 4, a chaotic sequence has a length of $M \times N \times 3$ is intercepted from the $\text{floor}(a_3 \times 10000)$-th bit of the chaotic sequence $x$, and this chaotic sequence scrambles the pixel positions with a value order of $L(i)$.

\[
[A, L] = \text{sort}(x)
\]

Step 6: There are three sequences, $R(i), G(i)$ and $B(i)$, with length of $M \times N$ divided by the chaotic sequence which the length is $M \times N \times 3$ obtained in step 5. After taking the these three segments’ values according to Eq.(5), they are respectively used as the sequence of RGB components of encrypted color image.
\[(a(i) \times p) \mod 256 \tag{5}\]

where, the value of \(a(i)\) is \(R(i), G(i), B(i)\); Considering that the value of the encryption sequence between 0-255, the value of \(p\) should be greater than 1000. In this paper, the value of \(p\) is 10000.

Step 7: XOR the sequences \(R(i), G(i), B(i)\) generated in step 6 with the corresponding \(I_R, I_G, I_B\) to the encrypted image data \(E_1, E_2, E_3\).

Step 8: Connecting \(E_1, E_2, E_3\) produces a sequence \(E\) with length \(M\times N\times 3\).

Step 9: Use the position sequence \(L(i)\) generated in step 5 to scramble the position of the data in sequence \(E\) (if the transmission problem is considered, it can be transmitted after this step), and convert the scrambled sequence into three \(M\times N\) matrices in order.

Step 10: Will get three matrices merged to form the final encrypted image.

2.3. Decryption Scheme

An algorithm for image encryption is proposed based on symmetric cryptography, and the decryption algorithm belongs to the inverse process encryption algorithm. In this decryption process of the encrypted image, it is crucial to obtain the chaotic sequence to determine the pixel position. Here, we use \([A, B] = \text{sort}(x)\) twice to generate the sequence of restoring the original image position.

3. Simulation Experiments and Performance Analysis

After encrypting and decrypting a plurality of color images according to the steps shown in Sect. 2.2, image encryption algorithm enactment is examined in Windows 10 environment with an operating system of 2.5 GHz Intel CPU I5-4200U, 4 GB RAM, and Matlab 2020b.

3.1. Encryption and Decryption Analysis

In this section, Lena (512×512) and a portion of Splash (260×512) are used to evaluate the mentioned algorithm’s performance. In the encryption and decryption process, the keys are chosen as \(a_0' = 0.02456\) and \(\mu_0' = 3.015\), with the constant \(q_1 = q_2 = q_3 = 1\), and show these encryption and decryption results using this algorithm. In Figures 2-3, we show how it encrypts and decrypts an images of any size, obscuring the original image’s information completely, which indicates its high security.

Figure 2. The Lena (512 × 512 ) image consists of a set of the original image (a), encryption image (b), and decrypted image (c).
Figure 3. The Splash (260 x 512) image consists of a set of the original image (a), encryption image (b), and decrypted image (c).

3.2. Histogram Analysis

In Figure 4 and 5, they show the histograms of Lena images, Splash and their corresponding cryptograms, respectively. They also show that the histogram pixel values of the original image will be centered within a specific range of values. Compared with the histogram of the original image, that of the encrypted image is more uniform, which shows that the proposed encryption method has a good performance in resisting statistical attacks.

Figure 4. Original histograms of Lena (a) in the red (b), green (c), and blue (d) components; Encrypted histograms of Lena (e) in the red (f), green (g), blue (h) components.

Figure 5. Original histograms of Splash (a) in the red (b), green (c), and blue (d) components; Encrypted histograms of Splash (e) in the red (f), green (g), blue (h) components.
3.3. Correlation Analysis

The image encryption algorithm could improve the correlation by dislocating and diffusing images with chaotic logistic sequences, and good permutation is measured by the correlation coefficient. This section analyzes a before and after image of encryption by Eq. (6).

\[ r(X, Y) = \frac{\text{cov}(X, Y)}{\sigma(X)\sigma(Y)} \]  

(6)

There is a covariance among X and Y, and \(\sigma(\cdot)\), determined by \(\text{cov}(X, Y)\). Accordingly, Y is composed of horizontal, vertical, and diagonal pixels, and \(r(x)\) is composed of the horizontal, vertical and the diagonal correlation coefficients. \(r(x)\) is used to determine the correlation coefficient for two adjacent pixels. The correlation is high if \(r(x)\) approaches 1, and very low if \(r(x)\) approaches 0. The correlation of encrypted images is shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Horizontal correlation</th>
<th>Vertical correlation</th>
<th>Diagonal correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena R</td>
<td>-0.0038989</td>
<td>0.0091044</td>
<td>0.008476</td>
</tr>
<tr>
<td>Lena G</td>
<td>0.0023145</td>
<td>-0.0053152</td>
<td>-0.0056106</td>
</tr>
<tr>
<td>Lena B (512 × 512)</td>
<td>-0.008546</td>
<td>-0.0081218</td>
<td>-0.0072199</td>
</tr>
<tr>
<td>Splas R</td>
<td>0.0016928</td>
<td>-0.0071815</td>
<td>0.0037677</td>
</tr>
<tr>
<td>Splas G</td>
<td>0.0032405</td>
<td>-0.0078424</td>
<td>0.0055903</td>
</tr>
<tr>
<td>Splas B (260 × 512)</td>
<td>0.01608</td>
<td>0.0063418</td>
<td>0.003174</td>
</tr>
<tr>
<td>Literature [4] R</td>
<td>0.0016</td>
<td>0.0041</td>
<td>-0.0089</td>
</tr>
<tr>
<td>Literature [4] G</td>
<td>-0.0071</td>
<td>0.0043</td>
<td>-0.0071</td>
</tr>
<tr>
<td>Literature [4] B</td>
<td>-0.0031</td>
<td>-0.0023</td>
<td>-0.0047</td>
</tr>
<tr>
<td>Literature [8] Average correlation</td>
<td>0.0116133</td>
<td>0.0125545</td>
<td>0.0218114</td>
</tr>
</tbody>
</table>

3.4. Information Entropy

In this section, this information entropy is calculated for a genuine image pixel value before and after encryption. Information entropy increases as the illness level of the image increases, and vice versa. Chaos level, on the other hand, increase the encryption effect and make algorithms more secure. As a result, information entropy can be expressed as tails.

\[ H = -\sum_{i=0}^{L} p(i) \log_2 p(i) \]  

(7)
where L is the image’s gray level and \( p(i) \) is the probability of occurrence of gray value i. It can be seen in Table 2—information entropy of the original image of Lena image and the encrypted image.

### Table 2. Information entropy comparison of RGB components of original image and encrypted image.

<table>
<thead>
<tr>
<th>Information entropy</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena (512×512)</td>
<td>7.9993</td>
<td>7.9992</td>
<td>7.9993</td>
</tr>
<tr>
<td>Splas (260×512)</td>
<td>7.9987</td>
<td>7.9987</td>
<td>7.9986</td>
</tr>
<tr>
<td>Literature [7]</td>
<td>Average information entropy</td>
<td>7.9989</td>
<td></td>
</tr>
</tbody>
</table>

As shown in Table 2—information entropy of the original Lena image and the encrypted image, L is the gray level, and \( p(i) \) is the probability that gray value i will occur. In Table 2, it illustrates that as pixels increase, the encryption image’s information entropy increases as well. From the information comparison in Table 2, by encrypting the exact sized image, this paper’s encryption algorithm has higher information entropy than other algorithms in the literature.

### 3.5. Encryption Speed

Time complexity is a significant tool for evaluating the efficiency of encryption algorithms [10]. The total time calculation includes the time of chaotic sequence generation and the time of image dislocation and diffusion. In Table 3, encryption and decryption time, the comparison with the running time of other algorithms and itself are shown in the literature. It can be seen from the literature that with the steady improvement of the hardware processing capability, the operating cycle and processing speed of the encryption algorithm will be improved.

### Table 3. Comparison of encryption and decryption time for images of different sizes and algorithms.

<table>
<thead>
<tr>
<th></th>
<th>Encryption time</th>
<th>Decryption time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree (256 × 256)</td>
<td>0.231710</td>
<td>0.239642</td>
</tr>
<tr>
<td>Lena (512 × 512)</td>
<td>0.360589</td>
<td>0.399276</td>
</tr>
<tr>
<td>Splas (260×512)</td>
<td>0.262001</td>
<td>0.290575</td>
</tr>
<tr>
<td>Literature [8] gray image (512 × 512)</td>
<td>0.296</td>
<td>-</td>
</tr>
<tr>
<td>Literature [11] Lena</td>
<td>2.090</td>
<td>-</td>
</tr>
<tr>
<td>Literature [12] (512 × 512)</td>
<td>0.62</td>
<td>-</td>
</tr>
<tr>
<td>Literature [10] color image (512 × 512)</td>
<td>1.76</td>
<td>1.54</td>
</tr>
</tbody>
</table>

From Table 3, as the time of encryption and decryption in the image encryption system varies with the environment, hardware or software, and system configuration, which directly compares with the encryption time of different algorithms is not rigorous. Therefore, this paper analyzes all steps of the computational complexity of the proposed image
encryption algorithm, including pseudo-random sequence generation, scrambling, and diffusion steps during encryption/decryption. Table 4 compares the calculations in this paper and other literatures.

### Table 4. Computational complexity comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Computation complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>$O(3\times M \times N)$</td>
</tr>
<tr>
<td>literature [13]</td>
<td>$O(24\times M \times N)$</td>
</tr>
<tr>
<td>literature [14]</td>
<td>$O(24\times M \times N)$</td>
</tr>
<tr>
<td>literature [15]</td>
<td>$O(8\times M \times N)$</td>
</tr>
<tr>
<td>literature [16]</td>
<td>$O(4(M\log N + M + N))$</td>
</tr>
</tbody>
</table>

4. Security Analysis

In this section, different methods of verifying the security of our method of image encryption are shown in this section. Using the same encryption algorithm, the encryption results of images of different sizes are the same. As shown in Section 3, this section selects the familiar Lena graph as the image for the study of encryption algorithm security.

#### 4.1. Key Space Analysis

When the key space is more extensive, brute-force cracking can be excluded more easily, and the time required to crack the key becomes greater. Using the algorithm submitted, there are two domains of key space: one associated with the logistic chaos mapping a system and another with plain-text keys associated with the image. Keys of the logistic chaos system are $\mu', a_0'$; and the one related to the plain-text consist of $q_1, q_2, q_3, a_1, a_2, a_3$.

As the computer has a precision of $(10^{15})$, and there are 8 keys in total, the key space proposed in this paper is $(10^{15})^8 = 10^{120}$, which is greater than $2^{100}$. As a result, brute force cracking can actually be avoided by the algorithm.

#### 4.2. Key Sensitivity Analysis

The more sensitivity the key, the more secure the algorithm. This paper chooses key $\mu'$ in order to test the encryption algorithm’s key sensitivity. Since the key effect of the sensitivity test is the same for any image, in this section, select the Lena image (512 x 512). Since the computer can handle the accuracy of $10^{-15}$, the value of $\mu$ is increased by $10^{-15}$ in this paper, and the decrypted image is observed as shown in Figure 6. The images show that any slight modification of the key cannot be decrypted accurately, suggesting that the encryption algorithm has restorative key sensitivity.
4.3. Differential Attacks

In this section of the proposed algorithm, differential attack resistance is tested. NPCR that is Number of Pixel Change Range and UACI that is Uniform Average Change Intensity are used to perform experiments on multiple image, defined as follows:

$$\begin{align*}
\text{NPCR} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j) \\
\text{UACI} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i, j) - C_2(i, j)|}{255}
\end{align*}$$

Here, according to the examination manner check in publication [17]. Firstly, the initial image is encrypted to accept a cryptographic image C1. Next, specify and modify a value in the initial image to be C2. Finally, the NPCR and UACI are estimated by Eq.(8) estimates the NPCR and UACI. As shown in Table 5, this paper’s encryption algorithm has NPCR values greater than 0.9965 and UACI values in the range of (0.3335, 0.3352). From Table 5, it can be seen that the encryption algorithm provided in this paper has NPCR values greater than 0.9960 and UACI values within the range of (0.3335, 0.3352). The ideal values of UACI and NPCR for image encryption anti-attack test are 33.4635% and 99.6094%, respectively, for the image encryption anti-attack test. Hence, this paper presents an image encryption algorithm that can resist the differential attack, indicating that it is effective at preventing that.

<table>
<thead>
<tr>
<th>Image</th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena(512×512)</td>
<td>0.9961</td>
<td>0.9962</td>
<td>0.9960</td>
<td>0.3335</td>
<td>0.3343</td>
<td>0.3342</td>
</tr>
<tr>
<td>Tree(256×256)</td>
<td>0.9960</td>
<td>0.9961</td>
<td>0.9956</td>
<td>0.3339</td>
<td>0.3335</td>
<td>0.3341</td>
</tr>
<tr>
<td>Splas(260×512)</td>
<td>0.9962</td>
<td>0.9960</td>
<td>0.9962</td>
<td>0.3342</td>
<td>0.3339</td>
<td>0.3352</td>
</tr>
<tr>
<td>Literature [17] Lena (256×256)</td>
<td>0.99611</td>
<td>0.99627</td>
<td>0.99616</td>
<td>0.33400</td>
<td>0.33329</td>
<td>0.33483</td>
</tr>
<tr>
<td>Literature [18] Tree (256×256)</td>
<td>0.9962</td>
<td>0.9963</td>
<td>0.9962</td>
<td>0.3353</td>
<td>0.3343</td>
<td>0.3339</td>
</tr>
<tr>
<td>Literature [5] Baboon (256×256)</td>
<td>Average</td>
<td>NPCR</td>
<td>0.99617</td>
<td>Average</td>
<td>UACI</td>
<td>0.33474</td>
</tr>
</tbody>
</table>

5. Conclusion

Using plaintext data for image encryption with attached size, a low-time-complexity chaotic encryption algorithm is presented in this paper. In addition, the key space of
reaches $10^{120}$. Since the initial values and parameters of logistic mapping are connected with the plaintext, this algorithm has higher security than other algorithms. As a result, the encryption key is generated dynamically, and it is possible to create nonlocal keys for each image, enhancing the algorithm’s resistance to differential attacks significantly. Through simulation experiments and investigations, we found that this algorithm not only ensures the security of the encryption algorithm, but also ensures the efficiency of encryption. In the future, through the modification and optimization of the algorithm and the continuous improvement of the hardware processing qualification, color image encryption will be used to transmit real-time images with high security requirements in this paper.

References

[1] Zhang Li, Wu Wenling, Zhang Lei, Zheng Yafei, Key recovery attack based on exchange equivalence reduction round AES-128, Computer research and development, 58 (10): 2213-2221(2021)
[7] Es A, Rw B, Aks B, Securing color image transmission using compression-encryption model with dynamic key generator and efficient symmetric key distribution, Digital Communications and Networks, 2020
[8] Zhang Miao, Tong Xiaojun, Wang Zhu, Chen Penghui, Joint Lossless Image Compression and Encryption Scheme Based on CALIC and Hyperchaotic System, Entropy, 23(8), 2021
[17] Wu Zeming, Pan Ping, Sun Chunyang, Zhao Bing. Plain-text-Related Dynamic Key Chaotic Image Encryption Algorithm, Entropy, 23(9), 2021