A Chaotic Image Encryption Algorithm for Smart Phones

Xuan Chen¹,²
¹Information Institute, Shandong University of Political Science and Law, Jinan, China
²University of Shandong Province Key Laboratory of forensic evidence, Jinan, China
Email: chenxjn@163.com

Abstract. At present, the conventional image encryption algorithm cannot be well applied to intelligent terminals of mobile internet due to their large computational cost and high storage requirements. Using the characteristics of chaotic sequence in the paper, the optimized encryption algorithm based on logistic and Tent mapping was used to achieve the optimization of the fast image encryption algorithm. The experimental results show that, this algorithm ensures the encryption effect while reducing the amount of computation required for image scrambling processing, and is suitable for encrypting image of mobile platforms.

1. Introduction
With the rapid development of mobile Internet technology, smartphone has become an indispensable tool for people's daily life and work. Because of the small memory and limited computing power of the mobile phone, it is a new challenge to study the encryption algorithm which is suitable for smartphone, good encryption effect, time complexity and small space complexity. Chaotic systems have the characteristics of initial value sensitivity, ergodicity, parameter controllability and pseudo-randomness, which provide new ideas for the research of digital image encryption algorithms. Therefore, chaotic image encryption algorithms have become a research hotspot [1].

The existing chaotic image encryption algorithms are generally based on scrambling and surrogate models[2,3]. LI and others[4] proposed an image encryption algorithm based on Hash function and multi chaotic system. The algorithm is relatively secure, but the encryption efficiency is not high. In order to maximize the encryption speed on the premise of guaranteeing security, the image is divided into a fixed-size pixel blocks, and the pixel block is the smallest operation unit. Recently, some image encryption algorithms based on pixel block have been proposed, such as the algorithm that is designed in [5,6].

Based on the research of image encryption technology, this paper proposes an innovative image encryption algorithm for mobile platform, which is mainly used to reduce the amount of computation by using image segmentation and multiple use of chaotic sequences.

2. Chaotic System
Chaos has the characteristics of sensitivity to the initial value and system parameters, the statistical characteristics of white noise and the ergodicity of the sequence, which conforms to the diffusion, scrambling and random characteristics of the cryptographic requirements. Because of these advantages of chaos, the image encryption algorithm in this paper uses both Logistic mapping and Tent mapping to scramble pixel positions and change pixel values.
2.1. Logistic Mapping

Logistic mapping is the insect amount model[7]. It is a simple but important nonlinear iterative equation and is a chaotic map that is widely studied at present. The mathematical expression of the one-dimensional Logistic map is as follows:

\[ x_{n+1} = \mu x_n (1 - x_n) \quad x \in [0,1], \mu \in [0,4] \]  

In the equation, \( \mu \) is called the control parameter. When \( 3.5699456 < \mu \leq 4 \), the Logistic system is in a chaotic state. Logistic mapping is used in many encryption algorithms because of the simple principle of chaotic mapping generated by Logistic mapping, good initial value sensitivity, long-term unpredictability and non-convergence.

2.2. Tent Mapping

Tent mapping refers to a piecewise linear mapping in mathematics, named for its function image resembling a tent. The recursive relation of tent mapping is as follows:

\[ x_{n+1} = \begin{cases} \mu x_n, & 0 \leq x_n < 0.5 \\ \mu (1 - x_n), & 0.5 \leq x_n \leq 1 \end{cases} \]  

When \( 0 < \mu \leq 2 \), the system enters a chaotic state.

3. Image encryption algorithm for mobile platform

3.1. Encryption Algorithm Design Idea

In this paper, by reducing the amount of computation by using image segmentation and multiple use of chaotic sequences, an image encryption algorithm suitable for smart phones is designed as follows:

Step1: Divide the image matrix to be encrypted into several matrix blocks.
Step2: Grayscale transformation and image scrambling for each block.
Step3: Disperse the pixels in each block into other blocks.
Step4: Combine blocks to the final encryption image.

3.2. Image Segmentation

For the original image \( I \), the pixel matrix size is \( M \times N \). Expand (fill with 0) pixel matrix to size \( M' \times N' \), so that \( M' \), \( N' \) are perfect square. Set \( \sqrt{M'} = m, \sqrt{N'} = n \). The pixel matrix of \( M' \times N' \) is divided into \( m \times n \) blocks of size \( m \times n \) in the order from top to bottom and from left to right. For convenience, we use two dimensional array \( A[i][j] \) to represent the \( (i,j) \)th block. Use the two-dimensional array \( P_{A[i][j]}[x][y] \) to represent the pixel values in the block \( A[i][j] \) ( where \( 0 \leq i \leq m-1, 0 \leq j \leq n-1, 0 \leq x \leq m-1, 0 \leq y \leq n-1 \) ). In order to increase the security, the Logistic chaotic system and the Tent chaotic system are first iterated for key1 and key2 times respectively. Iterations key1 and key2 are used as encryption keys. The initial value of Logistic is used as the key Key3 and the parameter \( \mu \) is used as the key key4. The initial value of the Tent system is the key key5.

3.3. Pixel Processing In Block

Using Logistic mapping to generate \( m \times n \) mutually unequal sequences from 0 to \( m \times n - 1 \), set as vector \( T[m \times n] \), and use vector \( T \) to process each block as follows, with block \( A[i][j] \) as an example.

3.3.1. Construction permutation array. A random number temp is generated by using Tent mapping. Add temp to each \( T[r] \) (\( r = 0,1,2, \ldots, m \times n - 1 \)), and then take the remainder of \( m \times n \) to obtain a new sequence \( T'[m \times n] \). The equation is expressed as follows:

\[ T'[r] = (\text{temp} + T[r]) \mod (m \times n) \]  

3.3.2. Replace the pixel values in blocks. For \( P_{A[i][j]}[x][y] \), first calculate the displacement position, set:
\[ k_1 = T[\times n + y]/(m \times n), \quad k_2 = T[\times n + y]%(m \times n) \]  

(4)

then:

\[ P'_{A[i][j][k1][k2]} = P_{A[i][j][x][y]} \]  

(5)

Where \(0 \leq i \leq m - 1, 0 \leq j \leq n - 1, 0 \leq x \leq m - 1, 0 \leq y \leq n - 1\).

3.3.3. XOR and encryption operation. For each pixel value \(P'_{A[i][j][x][y]}\) in the block, a pseudo-random number \(e\) is generated by the Tent chaotic mapping, and then the pixel value \(P'_{A[i][j][x][y]}\) is XORed with \(e\). which is:

\[ P'_{A[i][j][x][y]} = P'_{A[i][j][x][y]} \wedge e \]  

(6)

3.3.4. According to 3.3.1 to 3.3.3, each block is processed in turn.

3.4. Processing between blocks

The pixel values in each block are dispersed into other blocks. The \((i, j)\)th block pixel value \(P'_{A[i][j][x][y]}\) is deposed to the \((i, j)\) position in the \((i, j)\)th block. That is:

\[ P''_{A[i][j][i][j]} = P'_{A[i][j][x][y]} \]  

(7)

3.5. Forming encrypted image

The blocks processed after the step D are arranged in order of column priority from small to large, then the encrypted image \(I'\) is obtained.

4. Decryption algorithm description

Using the values of the keys key1, key2, key3, key4, and key5 during encryption, decrypting according to the reverse process of encryption, and then removing the added redundant data (if present), the decrypted image can be obtained.

The decryption process is shown as follows:

Step1: Divide the image matrix to be decrypted into several matrix blocks.

Step2: Swap pixels between blocks.

Step3: Decrypt in the block using grayscale transformation and image scrambling.

Step4: Combine blocks to the final decryption image.

5. Analysis of Algorithms

The algorithm mainly reduces the computational complexity and shortens the running time of the encryption by reusing and blocking the chaotic sequence. By analyzing the time complexity compared with [8] and [9] to prove that the algorithm shortens the encryption time, [8] and [9] are similar in structure and operation to the algorithm, and have certain comparability. The image matrix size assumed to be \(M \times N\) in the following analysis.

The algorithm first performs block processing on the image, but in the actual programming implementation, this step does not need a specific implementation. Secondly, the algorithm generates a random sequence with a size of \(m \times n\) and a time frequency of \(T(m \times n)\), Then the sequence is used to scramble the block, the time frequency is \(T(m \times n)\), and the number of blocks is \(m \times n\); Finally, the XOR operation of the image can be performed together with the scrambling operation, that is, the two steps are combined in the implementation. Before using the chaotic system, there is an iterative process with a time frequency of \(T(\text{key1} + \text{key2})\). Therefore, the time frequency of the entire encryption process is \(T(m \times n + M \times N + \text{key1} + \text{key2})\), in general, \(\text{key1} + \text{key2}\) is much smaller than \(M \times N\).

[8] first uses Arnold to scramble the image with a time frequency of \(T(M \times N)\); the time frequency of generating the chaotic sequence is also \(T(M \times N)\); then the chaotic sequence is used to encrypt the image matrix, and its time frequency is \(T(M \times N)\), so the total time frequency of the whole process is \(T(3 \times M \times N)\). In [9], the time frequency of encrypting the image matrix by scrambling operation is \(T(4 \times M \times N)\), and the gray-scale encryption process of the second step also scans the matrix once, and
the time frequency is \( T(M \times N) \), so the time frequency of the entire encryption process is \( T(5 \times M \times N) \). Moreover, the two documents also have an iterative process in the use of turbid system materials.

From the above theoretical analysis, the encryption processing time of this algorithm is significantly smaller than that of [8] and [9]. Later, the actual running time is used to prove that the algorithm is less than the encryption time of [9].

6. Analysis of experimental results

The experimental test platform is a dual-core CPU with a model number of ARM Cortex-A9, a RAM capacity of 1GB, and an Android 4.0 operating system.

The effect of encrypting Lena \((256 \times 256)\) using the algorithm of this paper is shown in figure 1. Figure 1(a) is an image before encryption, figure 1(b) is an encrypted image, and figure 1(c) is a decrypted image. As can be seen from the figure, the algorithm achieves a better encryption effect, and the original image cannot be seen through the encrypted image.

![Image 1](image1.png)

**Figure 1.** Image encryption and decryption comparison chart.

6.1. Key sensitivity analysis

The sensitivity of the ciphertext to the key is tested by making minor changes to the key. For simplicity, only the sensitivity to the key \( k3 \) is tested here.

Use peppers \((256 \times 256)\) image as a case, set \( k3 = 0.9000000 \). Figure 2(a) is an image before encryption, and figure 2(b) is an encrypted image using key \( k3 \), and figure 2(c) is a decrypted image. As can be seen from the figure, the algorithm achieves a better encryption effect, and the original image cannot be seen through the encrypted image.

![Image 2](image2.png)

**Figure 2.** Key sensitivity test chart

It can be seen from this sensitivity test that as long as there is a small change in the key, the result of the decryption is very different, which indicates that the encryption algorithm in this paper is highly sensitive to the key.

6.2. Histogram analysis

The histogram contrast before and after the Lena image encryption is shown in figure 3. Figure 3(a) is the original image histogram, and figure 3(b) is the encrypted image histogram. The histogram is a graph of the ordered distribution of the pixels in the image, reflecting the distribution of pixels after the image processing. From figure 3, it is known that the histogram distribution after encryption is very uniform, which covers the distribution regularities of pixel values before image encryption, and can effectively resist statistical analysis and it is difficult to decipher the ciphertext image from the histogram of the encrypted image.
6.3. Time test for encryption and decryption
The algorithm is compared with the algorithm in [9] to verify that the algorithm is suitable for mobile
devices, and the encryption and decryption time is shorter than the algorithm provided in [9].
Under the same operating environment, 10 experiments were performed on Lena (128*128, 256*256,
512*512) images respectively, and the average running time was taken. The experimental results are
listed in table 1. It can be seen from the table that the encryption and decryption time of the algorithm
is within 300ms, which is obviously better than the algorithm provided by [9].

| Image size | Time of algorithm in this paper | Time of algorithm in[9] |
|------------|--------------------------------|------------------------|
|            | encryption | decryption | encryption | decryption |
| 128*128    | 72         | 68         | 217        | 224        |
| 256*256    | 261        | 268        | 1401       | 1398       |
| 512*512    | 298        | 288        | 1683       | 1499       |

6.4. Correlation analysis
The original image is highly correlated in the horizontal, vertical and diagonal directions of adjacent
pixels. The purpose of the encryption algorithm is to destroy this correlation and encrypt the original
image into a noise-like image with little or no correlation. In this way, it is difficult for code breaker to
decipher the ciphertext image by correlation between pixels.
The correlation values between two adjacent pixels can be calculated by equation (8):

\[
R(x,y) = \frac{\text{Cov}(x,y)}{\sqrt{\text{V}(x)}\sqrt{\text{V}(y)}}
\]  

In the equation, \(x, y\) are the gray values of the two adjacent pixels in the image, \(\text{V}(x)\) is the
variance of \(x\). \(\text{Cov}(x, y)\) is the covariance of \(x\) and \(y\). A good encrypted image should not be visually
recognized a trace of the original image by the naked eye, and the correlation value with the original
image approaches zero. 1000 pairs of adjacent pixel values are selected randomly in horizontal and
vertical directions, and the correlation coefficients are calculated by the above equation, as listed in
table 2. By comparing the correlation coefficient before and after encryption, it can be found that the
algorithm effectively reduces the correlation between adjacent pixel values.
7. Conclusion
In this paper, an image encryption algorithm for smart phones is designed through the research of the common image encryption technology. In order to improve the encryption speed, the image is first divided into blocks, each block is processed, and then the pixel values in each block are dispersed into other blocks, and then the encrypted image is generated. The experimental analysis of the effect of the encrypted image shows that the encrypted image approximated the noisy image, the histogram distribution was smoother and balanced, and the adjacent pixels were not correlated, and also had good randomness. Therefore, the algorithm can be effectively applied to the mobile intelligent platform and obtains better encryption results. Compared with the image encryption algorithm in the computer, the algorithm still has a certain gap in the encryption effect. In the future research, the encryption effect will be enhanced to protect the privacy of the image in the personal mobile phone.

8. Acknowledgments
The recent work is supported by Scientific Research Project of Shandong University of Political Science and Law (2017Z20B) and Project of Shandong Province Higher Educational Science and Technology Program (J18KA383).

9. References
[1] WANG X Y, LIU L T and ZHANG Y Q 2015 Opt. Las. Eng. 66 10.
[2] REN X K and MA Ch 2015 Microelectron. Comp. 32 96.
[3] WANG L Y, SONG H J and LIU P 2016 Opt. Las. Eng. 77 118.
[4] LI F Y and XU J F 2010 Comp. Eng. Des. 31 141.
[5] LIU H J, ABDURAHMAN K and NIU Y J 2014 AEU-Int. J. Electron. Commun. 68 676.
[6] MOHAMED F K 2014 Eng. Sci. Technol., Int. J. 17 85.
[7] May R M 1976 Nature. 261 459.
[8] Guan Z H, Huang F and Guan W 2005 Phys. Lett. A 346 153.
[9] Gao Tie-gang and Chen Zeng-qiang 2008 Phys. Lett. A 372 394.

| Table 2. Correlation value comparison between original image and encrypted image. |
|----------------------------------------|----------------------------------------|----------------------------------------|
|                                      | Vertical direction                     | Horizontal direction                   |
|                                      | Red component                          | Green component                        | Blue component             |
|                                      | 0.9356                                 | 0.9722                                 | 0.9855                     |
| the original image                   | 0.9655                                 | 0.9833                                 | 0.9730                     |
|                                      | 0.0545                                 | 0.0531                                 | 0.0799                     |
| the encrypted image                 | 0.0998                                 | 0.0543                                 | 0.0522                     |