Image Encryption Based on 2D Baker Map and 1D Logistic Map

Priyanka Gupta
Shaheed Sukhdev College of Business studies
University of Delhi, Delhi, India

Aditya Gupta
Department of Computer Science
University of Delhi, Delhi, India

DOI: 10.23956/ijarcsse/V7I6/0281

Abstract: In this paper, a new image encryption is proposed based on the combination of the pixel shuffling and changing the gray values of image pixels. Firstly, the Baker map is used to shuffle the positions of the image pixels in the spatial-domain. Then the shuffled image is encrypted by logistic map. The performance of the proposed algorithm was measured through a series of tests. Encryption and decryption of the images can be successfully done by the proposed algorithm. The experimental results show that the encrypted image has good resistance against statistical attacks.

Keywords: Chaotic map, Baker map, 1D Logistic map, Shuffling, Correlation, Entropy

I. INTRODUCTION

The bulk data capacity and high redundancy are the inherent features of the images. Due to these intrinsic features of the images, the traditional encryption algorithm such as AES, DES, RSA for text encryption are not appropriate for the multimedia data due to their slow processing speed. Recently, interest in the chaos based image encryption has been increased in the field of information security, and various image encryption algorithms based on chaotic maps have been proposed. The chaos based image encryption techniques are secure enough to meet the demand for real-time image transmission over the communication channels. The main characteristics of chaotic maps are sensitivity to initial conditions and control parameters, pseudo-randomness and ergodicity [1, 3, 5, 6, 10-12] which make them robust against the statistical attacks.

In this paper a new algorithm for image encryption based on Baker map and 1D Logistic map is discussed. The rest of the paper is organized as follows. In section 2 Baker map and 1D Logistic map are discussed. In section 3 encryption and decryption algorithm is presented. Section 4 describes the experimental analysis. Security analyses are specified in section 5, and finally the paper is concluded in section 6.

II. CHAOTIC MAP

A. Baker map

The 2D Baker map is a chaotic map, can be described with the following formulas [2]:

\[ B(x, y) = (2x, y/2) \] when \( 0 \leq x < 1/2 \).
\[ B(x, y) = (2x - 1, y/2 + 1/) \] when \( 1/2 \leq x < 1 \).

\[ B(x, y) = \begin{cases} \frac{N}{n_i} \left( x - N_i \right) + y \mod \frac{N}{N_i} \\ \frac{N}{n_i} \left( y - y \mod \frac{N}{N_i} \right) + N_i \end{cases} \]

where \( i = 0, 1, \ldots, t \), \( N_i = 0 \), \( N_i \leq x < N_i + 1 \) and \( 0 \leq y \leq N \)

The inverse transform of baker map \( B^{-1}(x, y) \) is defined as follows:

\[ B^{-1}(x, y) = \begin{cases} \frac{N}{n_i} \left( x - x \mod \frac{N}{N_i} \right) + N_i \\ \frac{N}{n_i} \left( y - y \mod \frac{N}{N_i} \right) + x \mod \frac{N}{N_i} \end{cases} \]
The discretized Baker Map is applied on “camera” image of size 256 x 256 using the sequence \{32, 8, 64, 16, 8, 32, 64, 32\}, which comprises of 8 divisors of 256. The results are shown in Fig. 2. As Fig. 2(b) shows that after five iterations, the correlation among the adjacent pixels is removed and the image is completely different from original image.

![Camera image and shuffled image using baker map](image)

**Fig. 2** Camera image and shuffled image using baker map

### B. 1D Logistic Map

One-dimensional Logistic Map is a non-linear chaotic system proposed by R.M. May [8]. 1D Logistic Map is defined as follows:

\[ z(c+1) = \mu z(c)(1 - z(c)) \]

where \( c=0, 1, 2, \ldots \), and \( \mu \) is the system parameter, \( z(0) \) is the initial condition. The map shows good chaotic behavior for \( \mu \in [3.9, 4] \) and \( z(c+1) \) is the number between zero and one for all \( c \).

### III. PROPOSED ALGORITHM

The first step in the encryption process is shuffling of the pixels in the plain-image using Baker map. The second step encrypts the pixel values of the shuffled image using logistic map. The block diagram of the proposed encryption algorithm is shown in Fig. 3. The encryption steps of proposed algorithm are as follows:

1. **Step 1.** Consider the plain-image \( P(x, y) \) of size \( N \times N \), where \( x, y=1, 2, \ldots , N \).

2. **Step 2.** Apply Baker map using sequence \( \{n_1, \ldots , n_t\} \) to shuffle the pixels of the plain image \( P \). After iterating this step for \( R \) times we get shuffled image \( S(x, y) \).

3. **Step 3.** For encryption of every pixel value of \( S(x, y) \), convert the pixel values \( S(x, y) \) to binary number \( SB(x, y) \) and then iterate the logistic map for \( c \) times to obtain the resultant \( z(c+1) \). For every next pixel, the \( z(c+1) \) of previous pixel become starting point \( z(0) \). The resultant of 1D Logistic map is processed as follows:

\[ B_z = \text{de}2\text{bi} (\text{mod} (z(c+1) \times 10^{14}, 256)) \]

The function \( \text{de}2\text{bi}(x) \) converts the decimal number \( x \) to binary value and \( \text{mod}(x, y) \) returns the remainder after \( x \) divided by \( y \). The shuffled image \( S(x, y) \) is encrypted as:

\[ E_B(x, y) = SB(x, y) \oplus B_z \]

Where \( x, y=1, 2, \ldots , N \). The symbol \( \oplus \) represents the exclusive-OR operation bit by bit.

4. **Step 4.** For \( x, y=1, 2, \ldots , N \), perform the conversion of binary number \( E_B(x, y) \) to decimal number \( E(x, y) \) to obtained the encrypted image \( E(x, y) \).

![Block diagram of proposed scheme](image)

**Fig. 3** Block diagram of proposed scheme

The encrypted image is decrypted to plain-image using reverse steps of encryption process.

![Plain image and its histogram](image)

**Fig. 4** Plain image and its histogram
IV. EXPERIMENTAL ANALYSIS

The proposed encryption algorithm is implemented in MATLAB. We take a grayscale “camera” image of 256 × 256 in size for experimental purposes. The secret keys are chosen as R=5, \{t=8, n_1=32, n_2=8, n_3=64, n_4=16, n_5=8, n_6=32, n_7=64, n_8=32\}, z(0) = 0.3195, µ=3.9985, and c=100. The plain-image of size 256 × 256 and its histogram is shown in Fig. 4. The ciphered image and its histogram are shown in Fig. 5. From the Fig. 5, it is observed that the histogram of the encrypted image is uniformly distributed and is completely different from that of the original image.

V. SECURITY ANALYSIS

A. Key Space Analysis

Key space is defined as the total number of different keys that can be used in the used in the encryption/decryption procedure. The secret keys of the proposed algorithm is composed of two parts: shuffling key S1-Key and substitution key S2-Key. As described above, S1-Key is composed of R and \{n_{i},...,n_{t}\}. The total numbers of possible keys, K(N), depends on N and on total number of different divisors exist for N [2]. In this example, we take N=256 then K(N) is approximately equal to 10^{63}. S2-Key is composed of three initial values of Logistic map z(0), µ, c. If the precision is 10^{-10}, the key space size is \(10^{63}\cdot10^{30}\) i.e. \(10^{93}\). The S1-Key and S2-Key are independent of each other, thus the key space of the proposed cryptosystem is:

\[\text{Key}_{\text{total}}=\text{S1-Key} \times \text{S2-Key} \approx 10^{63} \times 10^{30} = 10^{93}\]

The key space size Key\(_{\text{total}}\) is large enough to resist all kinds of the brute force attacks.

B. Key Sensitivity Analysis

A good encryption system should be sensitive to the small changes in secret keys. A very little change in secret keys in decryption process results into a image which is completely different from plain-image. The experimental results demonstrate that proposed algorithm is very sensitive to secret keys. The decryption is performed using the correct secret keys, the image obtained after decryption is shown in Fig. 6. It is clearly seen that the decrypted image and its histogram are exactly same as that of the plain-image and its histogram.

Now, the decryption is performed using same keys except z(1)=0.3195000001. As it can be seen that, a minute change in z(1), the decrypted image and its histogram shown in Fig. 7 is totally different from the the plain-image and its histogram. Similarly, any changes made to secret keys R, \{n_{i},...,n_{t}\}, µ, and c, the encrypted image can not be correctly decrypted. This shows that the proposed algorithm is highly key sensitive to the secret keys.

![Fig. 6 Decrypted image and its histogram with correct keys](image)

![Fig. 7 Decrypted image and its histogram with wrong key z(1)=0.3195000001](image)
C. Analysis of correlation of two adjacent pixels

To evaluate the correlation property between two vertically adjacent pixels, two horizontally adjacent pixels, two diagonally adjacent pixels in an encrypted image following method are used. First we randomly select 4096 pairs of two adjacent pixels from the image. Second calculate the correlation coefficient of each adjacent pair by using the following formulas [4]:

\[
\bar{A} = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad \bar{B} = \frac{1}{N} \sum_{i=1}^{N} y_i
\]

\[
r = \frac{\sum_{i=1}^{N}(x_i - \bar{A})(y_i - \bar{B})}{\sqrt{\left(\sum_{i=1}^{N}(x_i - \bar{A})^2\right)\left(\sum_{i=1}^{N}(y_i - \bar{B})^2\right)}}
\]

where \(N\) is the number of adjacent pixels selected from the image to calculate the correlation in an image, \(x_i\) and \(y_i\) are the values of adjacent pixels vertically, horizontally and diagonally in the image.

Fig. 8 shows the correlation distribution of two horizontally adjacent pixels in the original image and that in the encrypted image. Fig. 9 shows the correlation distribution of two vertically adjacent pixels in the original image and that in the encrypted image. Fig. 10 shows the correlation distribution of two diagonally adjacent pixels in the original image and that in the encrypted image. It is observed that neighboring pixels in the plain-image are highly correlated either in horizontally, vertically or diagonally, while there is a little correlation between neighboring pixels in the encrypted image. The correlation coefficient values are shown in table I.

<table>
<thead>
<tr>
<th>Image</th>
<th>Orientation</th>
<th>Horizontal</th>
<th>Diagonal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td>0.9814</td>
<td>0.9503</td>
<td>0.9814</td>
<td></td>
</tr>
<tr>
<td>Encrypted image</td>
<td>0.0199</td>
<td>0.0160</td>
<td>-0.0042</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8 Correlations of two horizontally adjacent pixels in the original image and in the encrypted image.

Fig. 9 Correlations of two vertically adjacent pixels in the original image and in the encrypted image.

Fig. 10 Correlations of two diagonally adjacent pixels in the original image and in the encrypted image.
D. Information Entropy Analysis
The Information entropy $H(m)$ of a source $m$ expresses the degree of uncertainties in the system and is calculated as follows [9]:

$$H(m) = -\sum_{i=0}^{\infty} P(m_i) \log_2 [P(m_i)]$$

Where $P(m_i)$ represents the probability of occurrence of symbol $m_i$, and the entropy is expressed in bits. For good image encryption scheme, the entropy of encrypted image should be close to the ideal value $H(m)=8$. The value of information entropy for the plain-image is calculated as 6.8656. The entropy value of its corresponding encrypted image is 7.9966. This shows that the proposed algorithm is safe against entropy attack.

VI. CONCLUSION
In this paper, an image encryption algorithm based on combination of Baker map and 1D Logistic Map has been discussed. Experimental results show that the presented algorithm is not vulnerable to statistical attack, has large key space and highly sensitive to the secret keys.

REFERENCES