Symmetric key image encryption using chaotic Rossler system

Mrinal K. Mandal1*, Madhumita Kar2, Sandesh K. Singh3 and Vivek K. Barnwal3

1 Department of Physics, National Institute of Technology, Durgapur, 713209, India
2 Department of Physics, NSHM Knowledge Campus, Durgapur, 713212, India
3 Department of Computer Science, BCET, Durgapur, 713212, India

ABSTRACT

In this article, an algorithm for encryption of digital image based on chaos theory is proposed. This encryption algorithm includes two main operations of image element shuffling and pixel replacement. The analysis of cryptographic strength (resistance attack) has been performed to confirm the fact. The results of several experimental tests, such as key space analysis, key sensitivity analysis, and statistical analysis, show that the proposed algorithm for image cryptosystems provides an efficient and secure way for image encryption and the hacker cannot decrypt an encrypted image without original key. A comparison in terms of correlation between the original and encrypted images, number of pixels change rate, unified average changing intensity, and mean absolute error is performed, which proves that the plain image is very different from the encrypted one. We have also performed fixed point analysis to know how many pixels remain fixed after pixel shuffling operation. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

image encryption; chaos; hack; pixel shuffling; NPCR; UACI; MAE

Correspondence

Mrinal K. Mandal, Department of Physics, National Institute of Technology, Durgapur-713209, India.
E-mail: mrinalkanti.mandal@phy.nitdgp.ac.in

1. INTRODUCTION

With the advancement of science and technology, everyone wants to keep their data secure and safe from others. Everywhere, whether it is a private sector, public sector, defense services, hospital, or research, people are investing lots of money for good encryption technique because most of the information is stored in computers or transmitted across network. If these confidential images came into wrong hands, then some illegal activities might occur. Securing digital image data leads to the development of image encryption techniques. After 1990s, chaotic encryption systems have obtained tremendous importance for their capability to improve the degree of security because of some important properties such as the sensitive dependence on initial conditions and system parameters. For cryptography purpose, diffusion and mixing are useful properties of chaotic system. Many security models and cryptosystems based on number theory and chaotic map have already been proposed [1–10]. But it is found that some of them are either inefficient or weak in view of computational complexity and strength of security [11–15].

Cryptography is now routinely used to protect data, which must be communicated through public network or saved over long periods in computers to protect electronic fund transfers and classified communications. Classical cryptographic techniques are based on number theoretic or algebraic concepts. Chaos is another paradigm, which seems promising. The chaotic behavior is a subtle behavior of a nonlinear system, which apparently looks random. However, this randomness has no stochastic origin. It is purely resulting from the defining deterministic processes. Therefore, research community has realized that chaos-based communication and chaotic cryptography will be the only way out for secure communication and data storage. In this context, the chaos-based cryptographic algorithms have suggested some new and efficient ways to develop secure image encryption techniques. Good encryption schemes used chaotic systems for encryption key generation, and the key is then used for generation of chaotic sequences, which are used for pixel shuffling, pixel value confusion, and so on of the plain image to obtain the required encrypted image.

In the present work, three-dimensional chaotic Rossler system [16,17] is used for image encryption in which the major part is pixel shuffling after pixel replacement to overcome some common attacks and provide some good
results. The Rossler system is chosen to obtain three-dimensional large sequences of data. These data are restricted in the range 0 to 255 to match with the criteria for 8-bit gray scale image. The pixel value of the image is replaced by bit-XOR operation between the original image and chaotic sequence (x components data of the chaotic system). Then final encrypted image is obtained after pixel shuffling, which is performed first in row-wise then column-wise so that the whole image elements get shuffled. This shuffling process is performed on the basis of the y and z components of chaotic sequences of the Rossler system. The Rossler system is described in Equation 1. The chaotic behavior of the system for parameter values \(a=b=0.2\) and \(c=5.7\) is shown in Figure 1.

\[
\begin{align*}
    x &= -y - z \\
    \dot{y} &= x + ay \\
    \dot{z} &= b + z(x - c)
\end{align*}
\]  

(1)

The rest of the paper is organized as follows: In Section 2, the idea of the image encryption and decryption algorithms has been presented step by step. Section 3 presented the security analysis of the proposed encryption scheme such as key space analysis, key sensitivity analysis, statistical analysis, and differential analysis. Section 4 included the intensity analysis. The conclusion is drawn in Section 5.

2. ENCRYPTION AND DECRYPTION ALGORITHM

The encryption process involved three major steps. First, one is the generation of chaotic sequence using Rossler system. Next, pixel replacement and pixel shuffling is based on the generated chaotic sequences. Let \(I\) be an image of size \(M \times N\). The pixel of \(I\) is denoted by \(I(i,j)\), where \(i\) and \(j\) are in the range of \(1 \leq i \leq M\) and \(1 \leq j \leq N\). Now, \(I(i,j)\) denotes the 8-bits gray value at the pixel position \((i,j)\) of the image \(I\). During the generation of chaotic sequences, the initial conditions \((x_0, y_0, z_0)\) for the Rossler system are extracted from the secret key of 128 bits (16 characters) taken in ASCII form denoted as \(K = K_1K_2K_3 \ldots K_{16}\) (\(K_i\) denotes the 8-bit key character in the \(i\)-th key position).

Step 1. Generate the value of the initial conditions as mentioned in the succeeding text.

\[
\begin{align*}
    x_0 &= \text{mod}\left( K_{16} \times \sum_{i=1}^{5} \text{mod}\left( K_i \times 10^5, 1 \right), 1 \right), \\
    y_0 &= \text{mod}\left( K_{16} \times \sum_{i=1}^{5} \text{mod}\left( K_{i+5} \times 10^{10}, 1 \right), 1 \right), \\
    z_0 &= \text{mod}\left( K_{16} \times \sum_{i=1}^{5} \text{mod}\left( K_{i+10} \times 10^{15}, 1 \right), 1 \right).
\end{align*}
\]

Step 2. Generate the chaotic sequences \(x = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}\), \(y = \{y_1, y_2, y_3, \ldots, y_n\}\), and \(z = \{z_1, z_2, \ldots, z_n\}\) using Rossler system with initial conditions \((x_0, y_0, z_0)\) and \(a=b=0.2, \ c=5.7\), where \(k = M \times N\).

Step 3. Convert the chaotic sequences \(x, y, z\) into two-dimensional array \((x(i), y(i), z(i))\) respectively.

Step 4. Restrict the value of the chaotic sequences between 0 and 255 (8-bit gray scale) by the mod operation like \(x(i,j) = \text{mod}(x(i, j) \times 10000), 256)\).

Step 5. Execute pixel replacement of the original image by bit-wise XOR operation between the original image and the chaotic sequence \(x(i,j)\) as \(EI(i,j) = I(i,j) \oplus x(i,j)\). The symbol \(\oplus\) indicates bit-wise XOR operation.

Step 6. Generate a new chaotic sequence \(yz(i,j) = y(i,j) \oplus z(i,j)\). Now, pixel shuffling operation is performed with the help of sequence \(yz\) as mentioned in the succeeding text.

To understand pixel shuffling operation, we consider the chaotic sequence \(yz\), represented by matrix \(A\), and image \(EI\), represented by matrix \(D\). For simplicity, consider that the dimension of \(A\) and \(D\) is \(5 \times 5\). Now, sort the elements of the matrix \(A\) in ascending order row-wise so that it will look like as \(B\). Note down the positions of elements after sorting using unsorted table. [For example, after sorting (ascending order) the first row of \(A\), we obtain “1” and “8” at the first and second positions in the first row in matrix \(B\). But these two elements are present in the third and fourth positions in the matrix \(A\), so that the position matrix \(C\) is constructed by the position numbers (1 to 5 for a matrix of order 5 x 5) of the elements like 3 and 4 as indicated in matrix \(C\) for the elements “1” and “8”, respectively. Next, the elements of image matrix \(D\) will be shifted according to matrix \(C\); that is, the first and second positions of matrix \(D\) will be occupied by the elements of positions third and fourth of matrix \(D\).
respectively, as shown in matrix E. Now, shuffle all the elements of D (considered as image) row-wise according to C and after shuffle it look like E. Next, to execute column-wise shuffling, arrange the elements of A column-wise in ascending order and generate position table and accordingly shuffle the elements of E column-wise to obtain the final encrypted image matrix like F. This pixel shuffling mechanism is described in Figure 2. The overall encryption process is presented graphically in Figure 3.

The decryption is the inverse process of encryption. During decryption, continue the first four steps as mentioned in encryption process. Next, generate the chaotic sequence \(yz\) and perform pixel shuffling operation in reverse way as in encryption. After that, execute bit-XOR operation as mentioned in step 5 during encryption process to obtain back the original image.

3. SECURITY ANALYSIS

The encryption algorithm presented in this paper can take any 16 characters as one-time password, and it is very secure. In this section, we discuss the security analysis of the present algorithm such as key space analysis, sensitivity analysis, statistical analysis, fixed point analysis, and differential analysis to prove the security level of the proposed algorithm against the most common attacks.

3.1. Key sensitivity analysis

The real numbers \(x_0, y_0,\) and \(z_0\) represent the initial value of chaotic sequence for the encryption algorithm. The change in key can change \(x_0, y_0,\) and \(z_0,\) and consequently, it changes the chaotic sequence; as a result, the whole image is changed. Thus, the encryption algorithm has the ability to resist brute-force attack. An ideal image cipher should be extremely sensitive with respect to the secret key used in the algorithm. Flipping of a single bit in the secret key would produce a widely different cipher image. This guarantees the security of a cryptosystem against brute-force attacks. Sensitivity with respect to a small change in the secret key for several images has been tested. In the proposed cryptosystem, the cipher image cannot be decrypted correctly, although the slightly different key is depicted from Figure 4. At the time of encryption, we have used the key as as*&^%$1234VGHax, and during decryption, we have used same key and slightly different key as as*&^%$1234VGHay. From the result as shown in Figure 4, it is clear that decryption of the encrypted image is possible only when we use the same key. This analysis proves that the algorithm of the cryptosystem is sensitive.
to the key and it guarantees the security against known plain-text attacks also.

**3.2. Statistical analysis**

An idle cryptosystem should be resistive against any statistical attack. To show the robustness of present algorithm, we perform statistical analysis by calculating histogram and correlation coefficient between plain image and encrypted image, and the results are incorporated in this section to prove the validity of the proposed algorithm against any statistical attack.

**3.2.1. Histogram analysis**

Histogram analysis gives us the statistical properties of the ciphered image. Histogram may reflect how pixels in

![Figure 4](image-url)  
Figure 4. Results of key sensitivity analysis: (a) plain image, (b) encrypted image using key as *^&%$1234VGHax, (c) decrypted image using wrong key as *^&%$1234VGHay, and (d) decrypted image using same key as used by encryption.

![Figure 5](image-url)  
Figure 5. Histogram of (a) plain image and (b) encrypted image of Lena.
an image are distributed. Figure 5(a) and (b) shows the histogram of plain image and encrypted image, respectively. The histogram of the encrypted image is similar to that of random image, so the performance of the algorithm is good. The histogram of plain image and encrypted image is totally different, so a hacker cannot extract the pixel statistical nature of the plain image from the cipher image, and the algorithm can resist a chosen plain text or known plain text attacks.

3.2.2. Correlation coefficient analysis

We have tested the correlation between two adjacent pixels of the plain image and the encrypted image. In plain image, pixels are highly correlated with its adjacent pixel whether it is in horizontal, vertical, or diagonal, but for encrypted image, this correlation is very low. First, we have randomly selected 1000 pairs of adjacent pixels of an image, and then using the following formula, we have calculated the correlation coefficient,

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

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

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

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

where \(x\) and \(y\) are the values of two adjacent pixels of the plane images or ciphered images and \(\gamma\) is correlation coefficient. The correlation distribution of two adjacent pixels of the plane image and its encrypted image is shown in (Figure 6) the scatter diagrams, which looks like a cluster of dots in the \(x\)–\(y\) plane. The correlation coefficient of the plain Lena image is 0.973760, and in its encrypted image, the correlation coefficient is -0.008468 along the vertical direction. Similar results for horizontal direction and diagonal direction are obtained and presented in Table I.

3.3. Fixed point analysis

Image shuffling procedure may leave some pixel points fixed after shuffling. If a pixel is not changed after shuffling, we call it a fixed point. Fixed point analysis says the maximum percent of fixed pixel present in the image. The less is the percent of fixed point, whereas the better is the shuffling of pixel operation. According to present algorithm, the percentage of fixed points of Lena image is 40%, which is much less compared with other paper [18]. The calculated values of the percentage of fixed pixel for different images are shown in Table II.

3.4. Entropy analysis

Robustness of the algorithm is an important factor. To analyze the robustness of encryption algorithm, entropy information theory has been introduced here. The entropy is defined as \(H(m) = -\sum p_i \log_2(p_i)\), where \(p_i\) denotes the probability of pixel value \(i\). Theoretically, a true random

<table>
<thead>
<tr>
<th>Images</th>
<th>Maximum percent of fixed points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena.jpg</td>
<td>40.2832</td>
</tr>
<tr>
<td>Reaper.jpg</td>
<td>45.6238</td>
</tr>
</tbody>
</table>

Figure 6. Scattered diagram of (a) plane image Lena and (b) its encrypted image.
Table III. Entropy comparison.

<table>
<thead>
<tr>
<th>Images</th>
<th>Plain image</th>
<th>Encrypted image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena.jpg</td>
<td>7.415264</td>
<td>7.997045</td>
</tr>
<tr>
<td>Reaper.jpg</td>
<td>6.362127</td>
<td>7.996950</td>
</tr>
</tbody>
</table>

Table IV. Results of number of pixels change rate, unified average changing intensity, and mean absolute error.

<table>
<thead>
<tr>
<th>Image</th>
<th>NPCR</th>
<th>UACI</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena.jpg</td>
<td>99.60020</td>
<td>28.451221</td>
<td>72.5506</td>
</tr>
<tr>
<td>Reaper.jpg</td>
<td>99.642944</td>
<td>40.376509</td>
<td>102.9601</td>
</tr>
</tbody>
</table>

NPCR, number of pixels change rate; UACI, unified average changing intensity; MAE, mean absolute error.

Figure 7. Results of illegal tampering of the encrypted images and corresponding decrypted images.
system should generate $2^8$ symbols with equal probability. Therefore, according to the definition of entropy, the system must have the value of $H(m) = 8$. If the entropy is less than 8, there will be certain degree of predictability, and it is easier to hack the encrypted image security. Hence, if the entropy of encrypted image approaches 8, it will resist the entropy attack. Table III shows the entropy of the plain images and their corresponding encrypted images. From these data, it is clear that the entropy of the encrypted image is slightly less than 8, which proves the ability against the entropy attack.

### 3.5. Differential attack analysis

Differential attack is a general name of attacks applicable primarily to block ciphers working on binary sequences. The discovery of differential cryptanalysis is usually attributed to Eli Biham and Adi Shamir [19]. They have studied this type of attacks to various ciphers, including a theoretical weakness of the Data Encryption Standard. Since then, the differential attack has become a common attack that has to be considered during the cipher design. In image encryption, the cipher resistance to differential attacks is commonly analyzed via the number of pixels change rate (NPCR) and unified average changing intensity (UACI) tests [20]. NPCR represents the percentage of different pixel numbers between the plain image and the encrypted image, and UACI represents the average intensity differences between the plain image and the encrypted image or between two encrypted images. Consider two encrypted images $C_1$ and $C_2$, whose corresponding plain images have only one pixel difference. Let the gray scale values of the pixels at position $(i,j)$ are $C_1(i,j)$ and $C_2(i,j)$ of the two ciphered images $C_1$ and $C_2$, respectively. Define a bipolar array $D$ with the same size as $C_1$ and $C_2$. Then $D(i,j)$ is determined by the conditions: if $C_1(i,j)=C_2(i,j)$, then $D(i,j)=0$; otherwise, $D(i,j)=1$. NPCR and UACI are defined by the equations given in the succeeding text.

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%,$$

$$\text{UACI} = \frac{1}{M \times N} \left[ \sum_{i,j} \left| \frac{C_1(i,j) - C_2(i,j)}{255} \right| \right] \times 100\%.$$

The mean absolute error (MAE) is defined by the following equation as

$$\text{MAE} = \frac{1}{M \times N} \sum_{i,j} |C_1(i,j) - C_2(i,j)|.$$

The NPCR, UACI, and MAE results are given in Table IV.

### 4. INTENSITY ANALYSIS

An image encryption/decryption algorithm can resist illegal tampering of the intensity of the encrypted image to a certain extent. If an attacker modified the encrypted image intensity, then the authentic receiver will receive encrypted image with some distortion. Figure 7(a)-(c) shows some modified encrypted images, and their corresponding decrypted images are shown in Figures 7(d)-(f), respectively. From these results, it is clear that if an attacker modified the encrypted image, the authentic receiver can decrypt the image with some noise. Therefore, the proposed algorithm can resist illegal tampering to some extent.

### 5. CONCLUSION

A chaotic image encryption algorithm has been demonstrated using 3D Rossler system. Different types of security analysis and numerical experiments demonstrate the effectiveness of the proposed algorithm. Because the algorithm mentioned a symmetric encryption, there will be one-time key, which will keep data safe and secure as experimentally shown previously. Various security analyses such as statistical analysis, fixed point analysis, and differential analysis show some good results and hence prove that the proposed algorithm can resist any known attack and give good performance overall.

### REFERENCES


