A novel approach of image encryption using pixel shuffling and 3D chaotic map

Arwa Benlashram, Maryam Al-Ghamdi, Rawan AlTalhi and Pr. Kaouther Laabidi

1. Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21577, Saudi Arabia.
2. ENIT, Manar University, Tunis, Tunisia.
E-mail: arwa.alashram@yahoo.com

Abstract. – In recent years, with rapid development in information and communication technologies, sharing digital images in social media have facilitated. However, since images privacy matters in our society, keeping images protected became a big challenge we have to deal with. In this paper, we propose a new image encryption method using Image Pixel Shuffling and 3D Chaotic map. First, the plain image is encoded using pixel shuffling, then the result is XORed with a key, and finally, the 3D chaotic map is performed on it. The security will be increased by using three steps to encrypt the image. Our results show that the proposed approach has a good performance and it is better than those of some notable image encryption algorithms. We got entropy value 7.9901 of cipher image, NPCR of 99.6628%, and UACI of 33.6781%. MATLAB was used to implement the proposed image encryption approach.

1. Introduction
Nowadays, information security becomes a necessity for everyone. People want to keep their information and resources away from unwanted access, especially their private images, this rises the need of image encryption in our life. We can define image encryption as the process of converting the plain image into another version of image called cipher image by using some algorithms or techniques. It is also called image scrambling [1]. Image encryption is used for image content protection, image authentication and data hiding [1].

There are a lot of traditional encryption algorithms such as Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), Advanced Encryption Standard (AES), and the algorithm developed by Rivest, Shamir, and Adleman (RSA). However, these algorithms are efficient for text encryption, but they are not that much efficient for image encryption due to the size of images, which is usually big comparing to the text. they will be very expensive and time consuming [2].

Many researchers and developers proposed and designed many techniques to achieve better image encryption methods. For example, Zhenjun et al. in [1], proposed an efficient image encryption algorithm that encrypts images by using block shuffling. It divides the input image into blocks, and then shuffles these blocks to make initial version of encrypted image. Then, chaotic map is used to generate a secret matrix. Finally, the last encrypted image version is produced by XORing initial version with secret matrix. Panduranga et al. in [3], used input image of size $n \times n$ and initial block size $4 \times 4$. The input image is divided into many blocks with $4 \times 4$ size, then pixel values are shuffled in these blocks by using chaotic map to get the partial encrypted image. Kapil et al. in [4], presented a new technique for
encrypting images by using pixel shuffling combined with changing pixel values by using 128-bit secret key generated by henon chaotic map. Lingfeng et al. in [5], designed a technique that consists of three steps to encrypt images. First step is image shuffling, in this step the image’s pixel values are changed. In the second step image’s blocks are divided into sub-images, each sub-image consists of four blocks. Then every sub-image is encrypted by using certain function. Xuanping et al. in [6], proposed a chaotic image encryption method which deepens on total shuffling and bidirectional diffusion. The algorithm constructs the permutation by combining several small permutations, these permutations are generated by a chaotic system. Hikmat et al. in [7] proposed an efficient image encryption algorithm using a set of chaotic maps. Their proposed algorithm consists of three phases, first phase is confusion, then shuffling phase, and finally diffusion phase. A simple encryption non-liner technique based on 3D chaos has been proposed by Billal et al. in [8]. F. K. Tabash et al. in [2] proposed a new image encryption technique based on multi-pseudo random block permutation and 3 chaotic logistic maps.

In order to increase the level of security for our image encryption algorithm, we perform three phases. First, we shuffle image’s pixels to achieve initial encryption. Then, we perform XOR operation, the good thing in XOR encryption that it is impossible to reverse the operation without knowing one of the two values. The final phase is 3D chaotic map.

The reminder of this paper is organized as follows. Section 2 shows the background. Section 3 discusses the main idea of the proposed approach. Section 4 discusses the results and the analysis of our results. Conclusion and references are given in Section 5 and 6 respectively.

2. Background
There are two types of image encryption processes, position permutation and value transformation [8]. Position permutation is a mechanism of permuting pixels positions without changing the value of each pixel. In contrast, the value transformation type changes the value of the pixel without permuting pixel position. Pixel shuffling that we used in our proposed approach is one of the position permutation mechanism examples. While XOR operation is an example of value transformation mechanism.

There are many performance evaluation metrics that are usually used to evaluate the efficiency of image encryption approaches. In this paper we considered key sensitivity, histogram, entropy, NPCR and UACI, correlation coefficient metrics. Key sensitivity means that a small change in the key, should produce a widely different cipher image [9]. Histogram is a statistical analysis that shows the content and distribution of image’s pixels [10]. Ideal image encryption algorithms generate a cipher image with a histogram of uniform pixel values distribution [8]. Information entropy is used to measure the uncertainty related with random variables. NPCR and UACI are calculated to measure the difference between two encrypted images. One image is encrypted with original key and the other one with a single bit change in the key. Finally, correlation coefficient shows how strongly the pixels are related to each other.

3. Proposed approach
Our proposed image encryption approach composed of three main steps as shown in Figure 1(a). First is pixel shuffling, second is XOR operation, and finally 3D Chaotic Map is performed. Our proposed image decryption algorithm is an inverse process of the encryption process. First step is 3D Chaotic Map, then perform XOR operation. Final step is pixel shuffling, by using random key that is created in the encryption process. In the following sub-sections, we will discuss each step in detail.
3.1. Pixel Shuffling

Figure 1(b) shows the flowchart for Pixel Shuffling Steps, and the details of the steps are discussed below.

- **Step 1**: Read the Original Image (OI)
  
  \[ OI = \{OI, x, y\}, \text{where } 1 \leq x \leq W \text{ and } 1 \leq y \leq H, \text{ W and H respectively, are width and height of the original image pixels.} \]

- **Step 2**: Convert the original image to grayscale image.
  
  In this step, we convert the pixel value from a three-dimensional value (R, G, B) to one-dimensional value to reduce complexity. Many operations don’t work well with RGB images (3D pixels) but it works much better with grayscale images (1D pixel) (e.g. edge detection).

- **Step 3**: Generate a row vector from image pixel.
  
  Calculate the total number of the original image pixels. Convert the image matrix of size \( MN \) to a row vector with length \( NM \).

- **Step 4**: Random permutation of row vector.
  
  We obtain a new row vector by randomly replacing elements’ positions of the row vector.

- **Step 5**: Convert a row vector to matrix.
  
  The row vector is converted into matrix of size \( MN \) to produce the cipher image.

3.2. XOR Operation

After Pixel Shuffling process, the problem that pixels values of the image have not changed, which may cause histogram attack. To make our approach robust against this kind of attack, we need to change the image pixels values by using XOR operation. The feature of XOR operator that the image can’t be reversed without knowing the key.

- **Step 1**: Key generation.
  
  From Equation (1), we generate a matrix \( K \) from three selected images (M1, M2, and M3) that have the same size of the original image (OI) and we use it as a key.

  \[
  K(i,j) = \frac{(M1(i,j) \times M2(i,j)) \times M3(i,j)}{2} \tag{1}
  \]

  Where \( 1 \leq i \leq W \) and \( 1 \leq j \leq H, \text{ W and H respectively, are width and height of the original image pixels.} \)
• **Step 2:** XOR operation.
  
  We XOR the matrix (K) obtained from previous step and the image obtained from Pixel Shuffling process.

### 3.3. 3D Chaotic Map

Figure 1(c) shows the flowchart for 3D Chaotic Map steps, and the details of the steps are discussed below.

• **Step 1:** 3D Chaos generation.
  
  The simplest chaos generation process is logistic map which is given by Equation (2).

\[
X_{n+1} = \mu X_n(1 - X_n) \tag{2}
\]

Where, \(0 < X_n < 1\) and \(\mu = 4\), are the conditions that make the equation chaotic. Pawan et al. in [11] presented the 3D version that are given within following equations:

\[
X_{n+1} = \delta X_n(1 - X_n) + \beta Y_n^2 X_n + \alpha Z_n^3 \tag{3}
\]

\[
Y_{n+1} = \delta Y_n(1 - Y_n) + \beta Z_n^2 Y_n + \alpha X_n^3 \tag{4}
\]

\[
Z_{n+1} = \delta Z_n(1 - Z_n) + \beta X_n^2 Z_n + \alpha Y_n^2 \tag{5}
\]

The above equations show the chaotic behaviour for \(3.53 < \delta < 3.81\), \(0 < \beta < 0.022\), \(0 < \alpha < 0.015\) and the initial values of \(X, Y\) and \(Z\) can be any value between 0 and 1.

The Chaos sequences generated by using the Equations (3), (4), (5) and initial values of \(X(1) = 0.5250\), \(Y(1) = 0.8300\), \(Z(1) = 0.5700\), \(\delta = 3.7700\), \(\alpha = 0.0121\), and \(\beta = 0.0139\).

• **Step 2:** Chaos histogram equalization.
  
  As shown in Figure 2(a), (c) and (e) that histogram of \(X, Y\) and \(Z\) have non-uniform distribution. To increase the level of security, it should be equalized histogram. We equalize the histogram for a grey image by using the following formulas:

\[
X = \text{integer } (X \times \text{Key1}) \mod N \tag{6}
\]

\[
Y = \text{integer } (X \times \text{Key2}) \mod M \tag{7}
\]

\[
Z = \text{integer } (X \times \text{Key3}) \mod N \tag{8}
\]

Where, \text{Key1}, \text{Key2} and \text{Key3} are big numbers generally bigger than 20000. To simplify that, we need to consider, \text{Key1}, \text{Key2} and \text{Key3} as equal values.

Figure 2(b), (d) and (f) show that histogram equalized by equalling value of \text{Key1} = \text{Key2} = \text{Key3} = 20000, \(M = 256\) and \(N = 256\).
• **Step 3:** Row rotation.

Hossain et al. in [8] introduced a new approach for row rotation, that would help us for image pixel permutation, based on the value of chaos X from Equation (6). To enhance the security, we must rotate the rows left when the chaos value is even and rotate right when the chaos value is odd.

• **Step 4:** Column rotation.

There is a similarity between column rotation and row rotation. A column rotation is based on the value of chaos Y from Equation (7). To enhance security, we must rotate the columns up when the chaos is an even value and rotate down when the chaos is an odd value [8].

• **Step 5:** XOR operation.

XOR operation is the final phase of the encryption process. we XOR the Chaos and row-column shifted image [8].

4. Experimental results

We performed various experiments to validate the effectiveness of our proposed approach. In the following sub-sections, we discuss encryption validation, key sensitivity, histogram, entropy, NPCR and UACI, correlation coefficient analyses of our proposed approach. Finally, we compare between our proposed approach and other approaches. For our experiments, we used standard test images: Lena, Peppers and Mandrill with size 256 x 256.

All the work is done by MATLAB R2018b software, using MacBook Pro with CPU 2.5 GHz Intel Core i5 and 16 GB memory.

4.1. Encryption Validation

In order to validate the efficiency of our proposed approach, we apply it on a set of images. Figure 3 shows the stages of our encryption process. As we can see in Figure 3, the results show that the first two stages, pixel shuffling and XOR operation give us initial encryption. Figure 3(a) is the original Lena image, Figure 3(b) image after pixel shuffling, Figure 3(c) image after performing XOR operation on it, Figure 3(d) shows the final encryption image after applying 3D Chaotic map algorithm. Figure 3(e) shows the decrypted image. Figure 3(f), (g), (h), (i), and (j) illustrate the same process applied on Peppers image. Figure 3(k), (l), (m), (n), and (o) illustrate the same process applied on Mandrill image.
4.2. Key Sensitivity Analysis
A good chosen algorithm should be sensitive to any change in the key even if the change is very small. Suppose we use KeyA to encrypt an image and then we make a change in one bit of KeyA to get KeyB. The image encrypted using KeyA is completely different from the image encrypted using KeyB. To prove that we used KeyA to encrypt Lena image, Figure 4(a) shows the image encrypted using KeyA. We tried to decrypt the image using both KeyA and KeyB. In case of KeyA, the image was successfully decrypted to the plain image again as shown in Figure 4(b), however, with KeyB the plain image couldn’t be decrypted successfully as shown in Figure 4(c).

4.3. Histogram Analysis
Histogram is a statistical analysis that shows the pixel distribution of the image. It is widely used in image processing field and it is one of the simplest ways to validate the quality of the image encryption algorithm [12]. Ideal image encryption algorithms generate a cipher image with a histogram of uniform pixel values distribution [8].

In order to prove that our proposed approach is intricate to hack, we performed statistical analysis and the results are shown below. Figure 5(a) and (f) show the original image Lena and its histogram, respectively. Figure 5(b) and (g) present the image after implementing pixel shuffling and its histogram. Figure 5(c) and (h) show the image after doing XOR operation with our key K and its histogram. Figure 5(d) and (i) show the final encrypted image, after implementing 3D Chaotic algorithm, and its histogram. As we can see from our encrypted image histogram, the pixel values are uniformly distributed, and it is totally different from the histogram of the plain image. In addition, we performed the statistical analysis for Peppers and Mandrill images. Figure 6(a), (b), (e) and (f) show the histogram for original and cipher Peppers images. Figure 6(c), (d), (g), and (h) present the histogram for original and cipher Mandrill image.
Figure 5. Histograms of Original and Encrypted Images of Lena.

Figure 6. Histograms of Original and Encrypted Images of Peppers and Mandrill.

4.4. Entropy Analysis

Histogram is a visual analysis of the images, to do the theoretical analysis, we need to calculate the entropy value. Information entropy, also called Shannon's entropy, is used to measure the uncertainty related with random variables.

We used Equation (9) to calculate the entropy value $H(m)$ [8].

$$H(m) = -\sum_{i=0}^{N-1} P(m_i) \log_2[P(m_i)]$$  \hspace{1cm} (9)

Where $N$ refers to the total number of pixels values, $m_i$ indicates to symbol source and $P(m_i)$ refers to the probability of the symbol. Table 1 shows the entropy values for plain and encrypted images Lena, Peppers, and Mandrill. The optimal entropy value is 8.0, our entropy of encrypted image Lena and Peppers is 7.9901 while Mandrill is 7.9903 which are very close to the optimal value. So, we could say that our proposed encryption approach is robust against the entropy attacks.

|          | Lena                      | Peppers                   | Mandrill                  |
|----------|---------------------------|---------------------------|---------------------------|
| Plain Image | 7.4467                   | 7.5553                   | 7.2636                   |
| Cipher Image | 7.9901                   | 7.9901                   | 7.9903                   |

4.5. NPCR and UACI

We have two important measures in image encryption which are NPCR (Number of Pixel Change Rate) and UACI (Unified Average Changing Intensity). They are used to test the strength of image encryption algorithm which is robust against differential attacks. NPCR is used to test if the algorithm is sensitive to the change in plain image or not. Equations (10), (11) and (12) are used to calculate the values of NPCR and UACI [13].
\[
NPCR = \sum_{ij} \frac{D(i,j)}{MN} \times 100\%
\]

(10)

Where,

\[
D(i,j) = \begin{cases} 
0, & \text{if } C_1(i,j) = C_2(i,j) \\
1, & \text{if } C_1(i,j) \neq C_2(i,j)
\end{cases}
\]

(11)

UACI is used to test the number of averaged changed intensity between two different decrypted images of \(MN\).

\[
UACI = \sum_{ij} \frac{|C_1(i,j) - C_2(i,j)|}{MN \cdot 255} \times 100\%
\]

(12)

Where \(C_1\) and \(C_2\) are different cipher images generated by different keys. The results of NPCR and UACI are shown in Table 2 for three different plain images. The worst value of NPCR is 99.6505 and the worst value of UACI is 33.6096 which are satisfactory for image encryption.

| Parameter | Lena | Peppers | Mandrill |
|-----------|------|---------|----------|
| NPCR      | 99.6506 | 99.6628 | 99.6582  |
| UACI      | 33.6096 | 33.6737 | 33.6781  |

4.6. Correlation Coefficient Analysis

Correlation measures the degree of similarity between two adjacent pixels of an image [14]. The following Equations (13), (14) and (15) are used to calculate the correlation coefficients [8]. Where, \(R\) is the number of randomized pairs, and \(a, b\) are the values of the randomized image pair.

\[
y = \frac{\text{Cov}(a, b)}{\sqrt{D(a) \cdot D(b)}}
\]

(13)

\[
D(a) = \frac{1}{R} \sum_{i=0}^{R} (a - \bar{a})^2
\]

(14)

\[
\text{Cov}(a, b) = \frac{1}{R} \sum_{i=1}^{R} (a - \bar{a})(b - \bar{b})
\]

(15)

Table 3 shows the correlation coefficients for the images Lena, Peppers, and Mandrill, both plain and cipher images. As we can see in the results, the correlation coefficients of cipher image using our proposed algorithms are close to zero. This indicates that the proposed approach is resistance against statistical attack. Figure 7(a) shows the horizontal coefficient for the plain image Lena. Figure 7(b) presents the horizontal coefficient for the encrypted Lena image. Figure 7(c) and (d) show vertical correlations for plain image Lena and encrypted Lena image, respectively. It is clear that there is correlation between each two adjacent pixels in the cipher images.

| Direction | Lena | Peppers | Mandrill |
|-----------|------|---------|----------|
|           | Plain Image | Cipher Image | Plain Image | Cipher Image | Plain Image | Cipher Image |
| Horizontal | 0.9700 | -0.0127 | 0.9315 | 0.0021 | 0.8761 | -0.0049 |
| Vertical   | 0.9374 | -0.0242 | 0.8225 | 0.0122 | 0.9124 | 0.0102 |
4.7. Time Complexity Analysis
To evaluate the efficiency of our proposed approach, the encryption time has been computed using 20 different test images. The average processing time of encryption process is 0.01316 seconds.

4.8. Comparisons
In this section, we have compared the algorithms of the comparative chaos principle according to the following aspects: NPCR and UACI, entropy and correlation. In Table 4 we compared our algorithm with other algorithms based on NPCR and UACI measures, we conclude that our algorithm achieved better results than the rest. Table 5 shows the comparison of entropy among different algorithms with ours. We conclude that entropy value of our proposed approach is better than the majority of the other references. Also, it has a value near to the maximum value 8. In Table 6, we compared our algorithm to the other references based on correlation factor. It is clearly show that we got the best results compared to the other references.

**Table 4.** NPCR and UACI Comparison with Other Algorithms.

| Reference number | NPCR  | UACI     |
|------------------|-------|----------|
| [15]             | 90.21 | 31.00    |
| [16]             | 99.6115 | 33.6319 |
| [17]             | 99.6002 | 28.4512 |
| [18]             | 98.99  | 32.62    |
| [14]             | 99.6292 | 28.505  |
| [8]              | 99.6048 | 33.5044 |
| [19]             | 99.5376 | 32.5738 |
| **Ours**         | **99.6628** | **33.6781** |

**Table 5.** Entropy Comparison with Other Algorithms.

| Reference number | Plain Image | Cipher Image |
|------------------|-------------|--------------|
| [9]              | 7.4451      | 7.7333       |
| [1]              | 7.4455      | 7.9992       |
| [8]              | 7.5553      | 7.9890       |
| [12]             | 6.95        | 7.7719       |
| [20]             | 7.4455      | 7.9891       |
| [19]             | 7.4340      | 7.9874       |
| [15]             | 7.440       | 7.5220       |
| **Ours**         | **7.2636**  | **7.9901**   |

**Table 6.** Correlation Comparison with Other Algorithms.

| Reference number | Plain Image | Cipher Image |
|------------------|-------------|--------------|
|                  | Horizontal  | Vertical     |
|                  | Horizontal  | Vertical     |
| [8]              | 0.9700      | 0.9374       | −0.0043     | 0.0014        |
| [16]             | 0.9427      | 0.9873       | −0.0046     | 0.9360        |
| [20]             | 0.9535      | 0.9616       | 0.0095      | 0.0106        |
| [19]             | 0.9132      | 0.9768       | 0.0021      | 0.0004        |
| **Ours**         | **0.9700**  | **0.9374**   | −0.0127     | −0.0242       |
5. Conclusion
This paper proposed a new image encryption approach works in three steps. First, we perform pixel shuffling in the plain image, then perform XOR operation between the result from first step and a key generated by some mathematical operations. Finally, we use 3D chaotic map to get the cipher image. Our approach is so secure since we use secret keys that provide an enough large key space. To validate our proposed approach, many experiments have been performed including key sensitivity, entropy, NPCR, UACI, correlation coefficient analysis and the results have shown that our approach has better performances than many other approaches.

6. References
[1] Tang, Z, Zhang, X and Lan, W 2014. Efficient image encryption with block shuffling and chaotic map. Springer Science, 5430-5448.
[2] Tabash, F K, Rafiq, M and Izharrudin, M 2013. Image encryption algorithm based on chaotic map. International Journal of Computer Applications.
[3] H T, P S K, N and Kiran 2013. partial image encryption using block wise shuffling and chaotic map. International Conference on Optical Imaging Sensor and Security.
[4] Mishra, K, Saharan, R and Rathor, B 2017. A new cryptographic method for image encryption. Journal of Intelligent & Fuzzy Systems, 2885–2892.
[5] Liu, L., Hao, S., Lin, J., Wang, Z., Hu, X. and Miao, S. (2018). Image block encryption algorithm based on chaotic maps. The Institution of Engineering and Technology, 22-30.
[6] Zhang, X and Zhao, Z 2013. Chaos-based image encryption with total shuffling and bidirectional diffusion. Springer Science, 319–330.
[7] Abdullah, H N and Abdullah, H A 2017. Image encryption using hybrid chaotic map. International Conference on Current Research in Computer Science and Information Technology (ICCIT), 121-125.
[8] Hossain, M, Rahman, M, Rahman, A and Islam, S 2014. A new approach of image encryption using 3D chaotic map to enhance security of multimedia component. 3rd International Conference on Informatics, Electronics & Vision.
[9] Pareek, N K 2012. Design and analysis of a novel digital image encryption scheme. International Journal of Network Security & Its Applications (IJNSA), 95-108.
[10] Alsaedi, M 2017. Colored image encryption and decryption using chaotic lorenz system and DCT2. Cornell University Library, 1-22.
[11] Khade, P N and Narnaware, M 2012. 3D Chaotic functions for image encryption. IJCSI International Journal of Computer Science Issues, 323-328.
[12] Philip, M 2011. An enhanced chaotic image encryption. International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), 77-83.
[13] Wu, Y, Noonan, J P and Agaian, S 2011. NPCR and UACI randomness tests for image encryption. Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), 31-38.
[14] Abdurruhdha, H H and Nasir, Q 2011. Low complexity high security image encryption based on nested PWLCM chaotic map. 6th International Conference on Internet Technology and Secured Transactions, pp. 220-225.
[15] Hasnat, A, Barman, D and Mandal, S N 2016. A novel image encryption algorithm using pixel shuffling and pixel intensity reversal. International Conference on Emerging Technological Trends [ICETT].
[16] Zhu, S, Zhu, C and Wang, W 2018. A new image encryption algorithm based on chaos and secure hash SHA-256. entropy.
[17] Mandal, M K, Kar, M, Singh, S K and Barnwal, V K 2013. Symmetric key image encryption using chaotic Rossler system. Wiley Online Library, 2145-2152.
[18] Hussain, I, Shah, T and Gondal, M A 2013. Image encryption algorithm based on total shuffling scheme and chaotic S-box transformation. Journal of Vibration and Control, 2133–2136.
[19] Liu, H, Wang, X and kadir, A 2012. Image encryption using DNA complementary rule and chaotic maps. Applied Soft Computing, 1457–1466.
[20] Ahmad, M and Alam, S 2009. A new algorithm of encryption and decryption of images using chaotic mapping. International Journal on Computer Science and Engineering, 46-50.