www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 9, No: 2, 2022 (708-720)

El-Cezerî Journal of Science and Engineering
Vol: 9 No: 2, 2022 (708-720)
DOI : 10.31202/ecjse.995766

ECJSE

Research Paper / Makale

An Image Encryption Method by Sub-image Shuffling, Bit-level Permutation and Diffusion using Chaotic Maps

Mehmet DEMİRTAŞ[a*]

[a]Department of Electrical and Electronics Engineering, Faculty of Engineering, Necmettin Erbakan University, Konya, Turkey
mdemirtas@erbakan.edu.tr

Received/Geliş: 15.09.2021 **Accepted/Kabul:** 22.11.2021

Abstract: Transmission of images securely through a channel can be ensured using cryptography techniques. Encrypting an image with the help of chaotic maps provides security and authentication. This paper presents an image encryption method that consists of three main steps: Sub-image shuffling, bit-level permutation, and diffusion. The sub-image shuffling step is performed using a logistic map to increase the robustness of the proposed method. In the bit-level permutation process, the highest four bits of each pixel are obtained. The bit-level row and column transformations are applied using the sequences generated by a chaotic tent map. Finally, in the diffusion step, the values of the image's pixels are altered. Two different chaotic sequences which are obtained from the logistic map and tent map are employed. Five different parameters are selected as the secret keys for the encryption. The performance of the suggested method is tested with several analysis methods. Key space analysis shows that the proposed method can withstand brute force attacks. It is also proven that the method is highly sensitive to secret keys. Histogram analysis illustrates the fact that the encrypted image's pixels are uniformly distributed. Also, the correlation between the neighboring pixels is reduced in the encrypted image compared to the plain image. The differential analysis demonstrates that the method is sensitive to the slightest changes in the plain image even though the same secret keys are used.

Keywords: chaotic maps, cryptography, image encryption, logistic map, tent map

Kaotik Haritalarla Gerçekleştirilen Alt-Görüntü Karıştırma, Bit-Seviyesi Permütasyon ve Difüzyon ile Bir Görüntü Şifreleme Yöntemi

Öz: Görüntülerin bir kanal üzerinden güvenli bir şekilde iletilmesi kriptoloji teknikleri ile sağlanabilir. Kaotik haritalar yardımıyla bir görüntünün şifrelenmesi, güvenliği ve kimlik doğrulamasını mümkün kılar. Bu makale, üç ana adımdan oluşan bir görüntü şifreleme yöntemi sunmaktadır: Alt görüntü karıştırma, bit seviyesinde permütasyon ve difüzyon. Alt görüntü karıştırma adımı, önerilen yöntemin sağlamlığını artırmak için bir lojistik harita kullanılarak gerçekleştirilmiştir. Bit seviyesi permütasyon işleminde, her pikselin en yüksek dört biti elde edilir. Bit düzeyinde satır ve sütun dönüşümleri, kaotik bir çadır haritası tarafından oluşturulan diziler kullanılarak uygulanmıştır. Son olarak, difüzyon adımında görüntünün piksellerinin değerleri değiştirilmiştir. Lojistik harita ve çadır haritasından elde edilen iki farklı kaotik dizi kullanılmıştır. Şifreleme için gizli anahtar olarak beş farklı parametre seçilmiştir. Önerilen yöntemin performansı çeşitli analiz yöntemleri ile test edilmiştir. Anahtar uzay analizi, önerilen yöntemin kaba kuvvet saldırılarına dayanabileceğini göstermektedir. Ayrıca yöntemin gizli anahtarlara karşı oldukça hassas olduğu da kanıtlanmıştır. Histogram analizi, şifrelenmiş görüntünün piksellerinin eşit olarak dağıldığını göstermektedir. Ayrıca şifrelenmiş görüntüde komşu pikseller arasındaki korelasyon düz görüntüye göre çok daha azdır. Diferansiyel analiz, aynı gizli anahtarlar kullanılmasına rağmen yöntemin düz görüntüdeki çok küçük değişikliklere duyarlı olduğunu göstermektedir.

Anahtar Kelimeler: Çadır haritası, görüntü şifreleme, kaotik haritalar, kriptoloji, lojistik harita

How to cite this article
Demirtaş, M., "*An Image Encryption Method by Sub-image Shuffling, Bit-level Permutation and Diffusion using Chaotic Maps*" El-Cezerî Journal of Science and Engineering, 2022, 9 (2); 708-720.

Bu makaleye atıf yapmak için
Demirtaş, M., "Kaotik Haritalarla Gerçekleştirilen Alt-Görüntü Karıştırma, Bit-Seviyesi Permütasyon ve Difüzyon ile Bir Görüntü Şifreleme Yöntemi" El-Cezerî Fen ve Mühendislik Dergisi 2022, 9(2); 708-720.
ORCID ID: [a]0000-0002-9018-3124

1. Introduction

A secure data transmission from one point to another through a medium can be made possible using encryption techniques [1]. Since traditional encryption techniques are not able to provide sufficient security for the images [2], chaotic maps are widely used in the field of image encryption [3]. There are several types of chaotic maps employed for image encryption due to their sensitivity to initial values, ergodicity, determinacy, and pseudo-randomness [4,5]. Logistic map [6-8], tent map [9,10], sine map [11,12], Henon map [13,14], Arnold's cat map [15-17], Baker's map [18], Chebyshev map [19,20] are some examples of the chaotic maps utilized in image encryption.

Image encryption scheme generally consists of two steps. Firstly, the plain image's pixels are shuffled pseudo-randomly according to a secret key in the confusion/permutation step. Subsequently, in the diffusion step, pixels' values are changed [21]. These steps aim to reduce the resemblance between the plain image and the encrypted image. In this study, the confusion step is performed using sub-image shuffling and bit-level permutation. The sub-image shuffling makes the confusion step more complicated which results in a robust scheme [22]. The bit-level permutation is the next operation of the confusion step which is generally used to change both the positions and values of the pixels [23-25]. To change the values of the pixels in the diffusion stage, two different chaotic sequences are generated with the help of a logistic map and tent map. Pixel values are changed accordingly, and each value of the updated pixel depends on the value of the previously updated pixel's value. This process can improve the scheme's resistance against statistical and differential attacks [23,26].

The performance of the proposed image encryption method can be measured by several evaluation metrics. Key space analysis evaluates the method's resistance against brute-force attacks [25]. Theoretically, the secret key space must be larger than 2^{100} [27]. On the other hand, key sensitivity analysis shows how much the encryption is sensitive to the selected secret keys. The decryption of an encrypted image with a slightly different secret key must produce a completely different image than the plain image [28]. Histogram graph is another evaluation metric that should be uniform in the encrypted image so that the scheme can prevent statistical attacks [29]. In the correlation analysis, the correlation coefficients between adjacent pixels are computed in horizontal, vertical, and diagonal directions. The encrypted image's adjacent pixels must have a low correlation which means the correlation coefficients must be very close to 0 [30]. Finally, Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI) metrics can be used to perform differential analysis. Although NPCR and UACI tests are criticized in the literature [31] such that they are not sufficient to comment on an encryption scheme's security, they are still widely used as an evaluation measure along with other metrics [32-35]. In this paper, the aforementioned evaluation metrics are used as analysis methods for the proposed encryption scheme

The organization of the paper is given as follows. In the Materials and Methods section, the encryption method is explained in detail. The Results and Discussion section includes experimental results and performance analysis of the suggested method. Finally, concluding remarks are presented in the Conclusion section.

2. Material and Methods

The proposed image encryption method consists of two steps. In the first step, the plain image is confused using two shuffling operations. Initially, the original image is divided into 4 blocks and each block is further subdivided into 4 sub-images. In total, 16 different sub-images are obtained. The blocks and the sub-images in each block are shuffled pseudo-randomly using the chaotic logistic map. Subsequently, a bit-level permutation process is implemented on the highest four bits of each pixel

using the tent map to complete the confusion process. Secondly, in the process of diffusion, the shuffled image's pixels are changed using the logistic map and tent map.

2.1. Sub-image Shuffling

Sub-image shuffling is the first process to enhance the complexity and robustness of the proposed image encryption method. The plain image is split into 4 same-size blocks namely A, B, C, and D. Those square blocks are then divided into 4 same-size sub-images which are labeled as 1,2,3, and 4. A, B, C, D blocks and the sub-images in each block are shuffled independently using the same logistic map. The logistic map which is used to shuffle the blocks and the sub-images can be written as in Equation 1.

$$x_{n+1} = \mu x_n(1 - x_n) \tag{1}$$

where μ is the control parameter which is in the interval of (0,4]. The output sequence x_n is in between 0 & 1 while x_0 represents the initial value of the logistic map. If the control parameter μ is chosen approximately 3.56995 or larger, a pseudo-random chaotic sequence like $\{x_n\}_{n=0}^{\infty}$ can be generated by iterating the logistic map.

The chaotic sequence could be employed to create 5 different arrays. One of the arrays is used to scramble A, B, C, and D blocks while the other four arrays are used to shuffle the sub-images in the blocks. Each array consists of integers 1, 2, 3, and 4 in a pseudo-random order. Equation 2 should be iterated sufficiently to obtain the shuffling arrays.

$$A = mod(floor(x_K \times 10^{10}), 4) + 1 \tag{2}$$

where A is an integer in between 1 and 4. The $floor$ operator outputs the greatest integer less than or equal to the multiplication result. x_K denotes the value of the Kth element of the chaotic sequence obtained by (1). K is incremented in each iteration of (2) in which the values in the sequence $\{x_K, x_{K+1}, x_{K+2}, x_{K+3}$ ...} are used until the shuffling arrays are obtained. For example, if the shuffling array is calculated as [2 1 4 3], then the second sub-image moves to the upper left, the first sub-image moves to the upper right, the fourth sub-image moves to the bottom left, and the third sub-image moves to the bottom right position. All blocks and sub-images are shuffled according to the shuffling arrays. It is clear that the neighboring pixels are highly correlated in the plain image. To reduce this strong correlation, sub-image shuffling and bit-level permutation are both implemented to accomplish confusion of the plain image.

2.2. Bit-level Permutation

A pixel of a grayscale image has 256 different brightness values ranging from 0 to 255; therefore, a pixel can be represented by an 8-bit binary number. Obviously, more significant bits contain more information about the grayscale image. The binary image representations of the four most significant bits for the cameraman image are shown in Figure 1. In this figure, the rightmost binary image shows the most significant bits of the pixels of the cameraman image. This binary image is the most similar to the original image. Actually, the four most significant bits shown in Figure 1 contain approximately 94% of the original image's information [24]. For that reason, the bit-level permutation is only implemented on these most significant bits to reduce the execution time of the encryption algorithm.

Bit-level permutation and pixel-level permutation are different from each other. While pixel-level permutation aims to change the positions of the pixels, the bit-level permutation process both changes the positions of the bits and so the values of the pixels.

Figure 1. Binary image representations

Given an original grayscale image of size M x N, $P_{x,y}$ represents the value of the pixel at (x, y) position. Each pixel value is converted into 8-bit binary values so that any pixel can be expressed as a vector of binary numbers as given in (3).

$$pixel_{bin} = \{P_{x,y}(1), P_{x,y}(2), \dots, P_{x,y}(8)\} \tag{3}$$

where $P_{x,y}(i)$ represents the ith bit of the pixel value at (x, y) position of the image. A matrix consisting of one of the highest four bits of the pixel values is shown in (4) where $j = 5,6,7,8$. B_8, B_7, B_6 and B_5 binary matrices are formed using Equation 4.

$$B_j = \begin{bmatrix} P_{1,1}(j) & \cdots & P_{1,N}(j) \\ \vdots & \ddots & \vdots \\ P_{M,1}(j) & \cdots & P_{M,N}(j) \end{bmatrix} \tag{4}$$

The bits in the binary matrices are permuted according to the values generated by the tent map. The tent map $T(x)$ can be defined iteratively as in Equation 5.

$$T(x) = x_{n+1} = \begin{cases} \mu x_n & \text{if } x_n < 0.5, \\ \mu(1 - x_n) & \text{otherwise,} \end{cases} \tag{5}$$

where $\mu \in [0,2]$ is a control parameter, and x_n is in between 0 and 1. If the initial value $x_0 \in [0,1]$, then $T(x) : [0,1] \to [0,1]$. A chaotic orbit $\{x_0, x_1, x_2, \dots\}_{n=0}^{\infty}$ can be generated if the control parameter is in between 1 and 2. This chaotic sequence should be sensitively dependent on the initial conditions. In Figure 2, two time series plot of an iterated tent map is shown where the initial conditions are set to be 0.2 and 0.20001 and the control parameter is 1.999. It can be observed from Figure 2 that different initial conditions produce a unique sequence of numbers even if the initial conditions are very close to each other. The produced numbers are deterministic, aperiodic, and bounded between 0 and 1 which implies that the system in (5) has chaotic dynamics given the mentioned conditions.

Given a binary image of size M x N, the bit-level permutation process is implemented using the following steps:

- The chaotic tent map given in (5) is used to produce the sequence $S_1 = \{x_{a+1}, x_{a+2}, \dots, x_{a+M}\}$ where a is defined as the number of nonzero elements of the binary image matrix.
- Similarly, another chaotic sequence $S_2 = \{x_{b+1}, x_{b+2}, \dots, x_{b+N}\}$ is generated using the tent map where b is the total number of zero elements of the given binary image matrix.

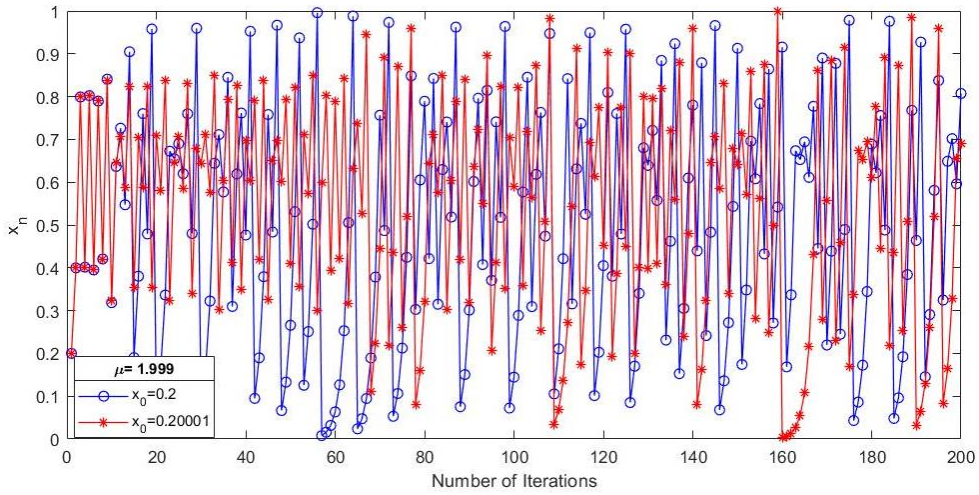

Figure 2. Time series plot of tent map for two different initial values

- The elements in S_1 and S_2 are sorted in ascending order. Next, the sorted elements' positions are found in the original sequences. Thus, two position arrays $S_r = (r_1, r_2, \dots, r_M)$ and $S_c = (c_1, c_2, \dots, c_N)$ are obtained.
- Firstly, the row transformation is performed on the binary matrix using S_r elements. For instance, r_1th row is moved to the first row, r_2th row is moved to the second row and so on. Then, the columns of the binary matrix are scrambled using the sequence S_c. Similar to the row transformation, c_1th column is moved to the first column, c_2th column is moved to the second column and so forth.
- This bit-level permutation process is performed for B_8, B_7, B_6 and B_5 binary matrices. Then, permuted and unpermuted binary matrices are combined to form the ciphered image.

When this bit-level process is applied to the binary image representations shown in Figure 1 by iterating a tent map with $\mu = 1.999$ and $x_0 = 0.2$, the scrambled binary images displayed in Figure 3 can be obtained. In this process, x_0 value can be employed as the secret key.

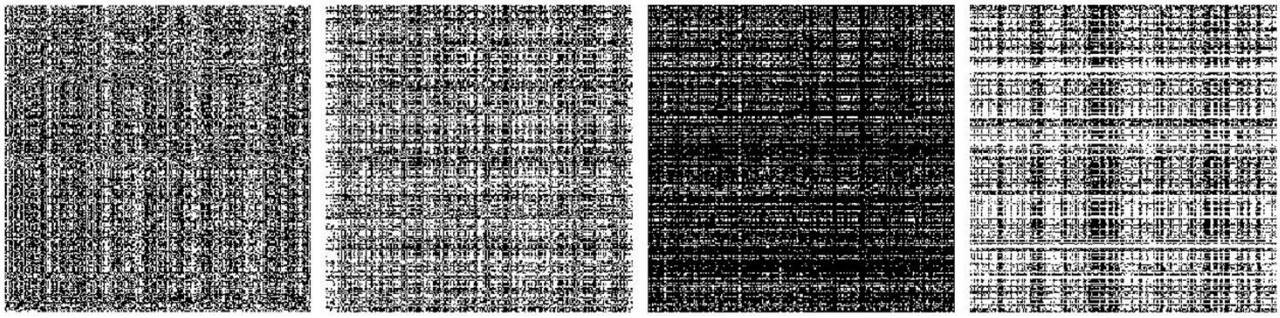

Figure 3. Scrambled binary image representations

2.3. Diffusion

When the sub-image shuffling and the bit-level permutation are completed, the diffusion process is implemented to get the encrypted image. The diffusion process aims to change the values of the pixels so that better encryption that resists differential and statistical attacks is possible. The diffusion operation is described as follows.

- Assume an image I_{MxN} is obtained after the sub-image shuffling and the bit-level permutation processes. This image matrix is converted into a one-dimensional vector $I_{1\,x\,MN}$ by scanning each element horizontally starting from the upper left element.
- Equation (1) is iterated $L_0 + MN$ times and the chaotic sequence $x_1 = \{x_{L_0+1}, x_{L_0+2}, \dots, x_{L_0+MN-1}\}$ is obtained. Similarly, Equation 5 is iterated $T_0 + MN$ times to get another chaotic sequence $x_2 = \{x_{T_0+1}, x_{T_0+2}, \dots, x_{T_0+MN-1}\}$. $L_0, T_0 \in Z$ and initial values of the chaotic functions are the secret keys for the diffusion process.
- The chaotic sequences are converted into integers between [0,255] using the following formulas.

$$L(i) = mod(floor(x_1(i) \times 10^{10}), 256) \qquad i = 1,2, \dots, MN \tag{6}$$

$$T(i) = mod(floor(x_2(i) \times 10^{10}), 256) \qquad i = 1,2, \dots, MN \tag{7}$$

- Finally, the diffusion operation is implemented as expressed in Equation 8 using the sequences given in (6) and (7). Here, $\oplus$ works as the bitwise XOR operator, and C represents the ciphered final image. All pixel values are altered sequentially, and the new value of each pixel depends on the value of the previous pixel. When the diffusion process is over, the ciphered one-dimensional vector of size 1 x MN is reshaped to form an M x N matrix.

$$\begin{aligned} C(1) &= I(1) \oplus \Big(mod\big((L(1) + T_0), 256\big) \oplus mod\big((T(1) + L_0), 256\big)\Big) \\ C(i) &= \big(I(i) \oplus C(i-1)\big) \oplus \Big(mod\big((L(i) + T_0), 256\big) \oplus mod\big((T(i) + L_0), 256\big)\Big) \end{aligned} \tag{8}$$

where $i = 2, \dots, MN$.

The overall encryption architecture suggested in this work is shown in Figure 4. The ciphered image can be decrypted by following the above steps in reverse order.

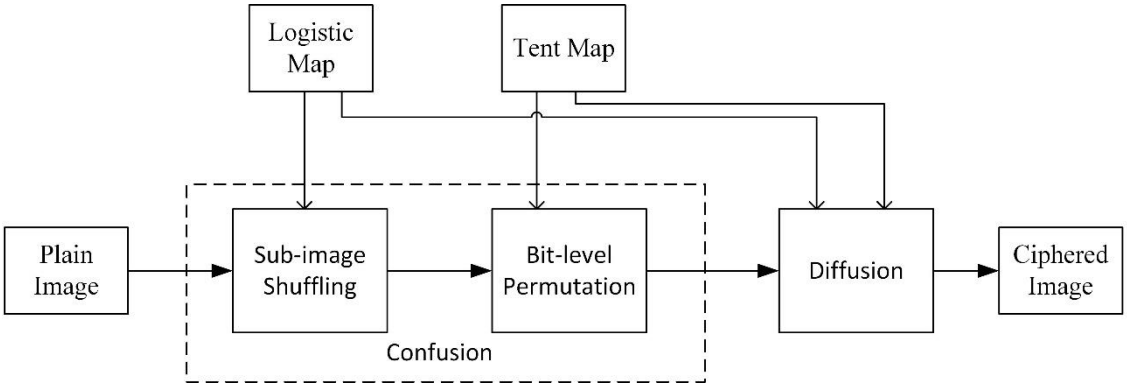

Figure 4. The encryption architecture

3. Results and Discussion

3.1. Experimental Results

The experimental analysis of the proposed encryption method is performed using MATLAB 2017b in a PC with an Intel Core i7 processor (2.80 GHz) and 16 GB RAM. Cameraman image which is a standard grayscale image with a size of 256 x 256 is used as the plain image for the experimental analysis. Figure 5a shows the original Cameraman image, Figure 5b is the image after sub-image

shuffling, Figure 5c displays the image after bit-level permutation and Figure 5d is the final ciphered image.

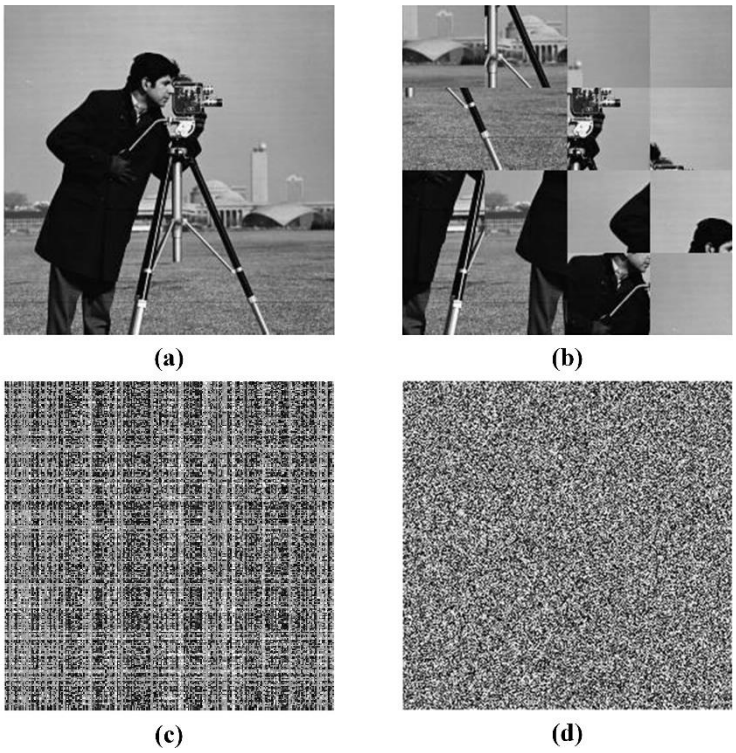

Figure 5. (a) Plain image (b) The image after sub-image shuffling (c) The image after bit-level permutation (d) Ciphered image

3.2. Key Space Analysis

The set of all possible keys constitutes the key space that must be sufficiently large to resist brute-force attacks. The key-space must be larger than 2^{100} to be secure [27]. In this work, five different variables are determined as the secret keys. In the bit-level permutation process, the initial value of the tent map (x_{01}) which has a precision of 10^{-14} is the first secret key. Similarly, the initial values of the tent map (x_{02}) and logistic map (x_{03}) used for diffusion are other keys which have a precision of 10^{-10}. Also, L_0 and T_0 values can be utilized as secret keys whose range are assumed to be in [0,255]. Therefore, they can be represented by 8 bits. The key space can be computed as in (9). The key space size is far larger than the required size which implies that the method meets that necessary condition.

$$Key\ Space = 10^{14}\ x\ (10^{10})^2\ x\ (2^8)^2 \approx 2^{129} > 2^{100} \tag{9}$$

3.3. Key Sensitivity Analysis

An efficient image encryption method should be sensitively dependent on the exact values of the secret keys. A small change in one key must cause a totally different decrypted image even if the other keys are correct. The following keys are chosen to test the key sensitivity of the proposed method.

$$Keys = [x_{01}, x_{02}, x_{03}, T_0, L_0] = [0.2, 0.2, 0.399999999, 200, 150] \tag{10}$$

Figure 6a illustrates the resulting image if the ciphered image is decrypted with correct keys as given in (10). Figures 6b, 6c, 6d, 6e, and 6f show that even a minor change in only one key results in a wrongly decrypted image. This proves that the encryption method is sensitive to secret keys.

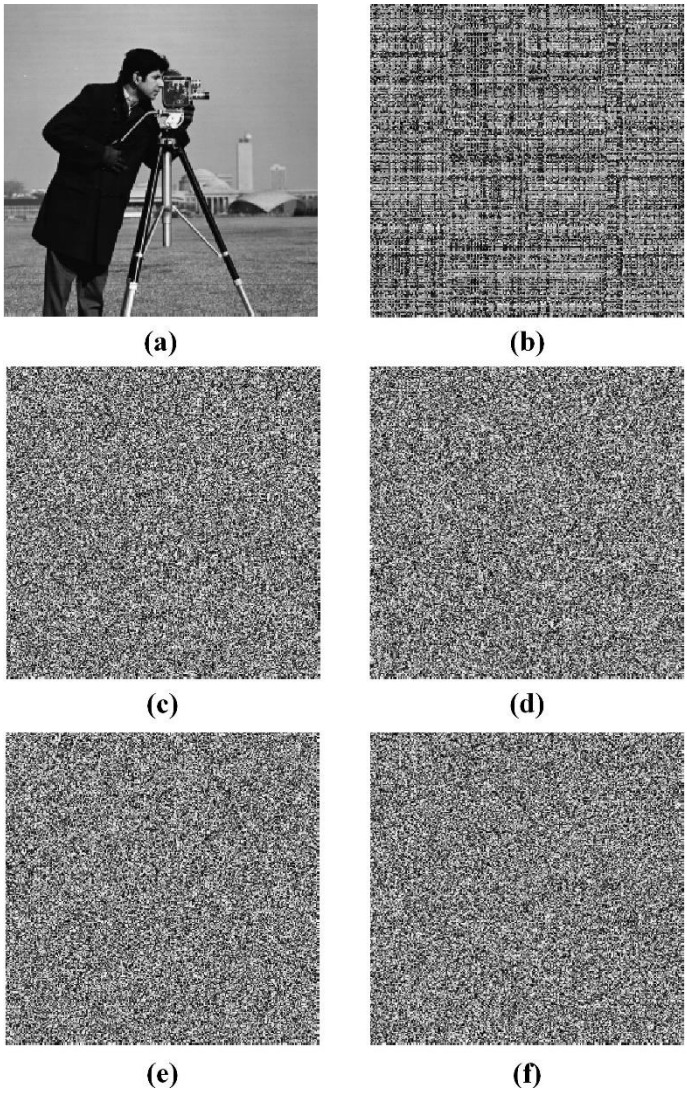

Figure 6. (a) Decryption with the correct keys (b) Decryption with the wrong key: $x_{01} = 0.2 + 10^{-14}$ (c) Decryption with the wrong key: $x_{02} = 0.2 + 10^{-10}$ (d) Decryption with the wrong key: $x_{03} = 0.3999999999 - 10^{-10}$ (e) Decryption with the wrong key: $T_0 = 201$ (f) Decryption with the wrong key: $L_0 = 149$

3.4. Histogram Analysis

The distribution of the pixel values of an image can be visualized by a histogram graph. A securely encrypted image must have a uniform distribution of pixel intensity values so that statistical attacks can be made infeasible. If the pixel values of the ciphered image are not uniformly distributed, an attacker may obtain a certain amount of information about the image using statistical analysis. Figure 7a shows the histogram of the plain Cameraman image and Figure 7b illustrates the histogram of the

encrypted image. The histogram of the encrypted image shows a uniform distribution which proves that the encryption method can resist statistical attacks.

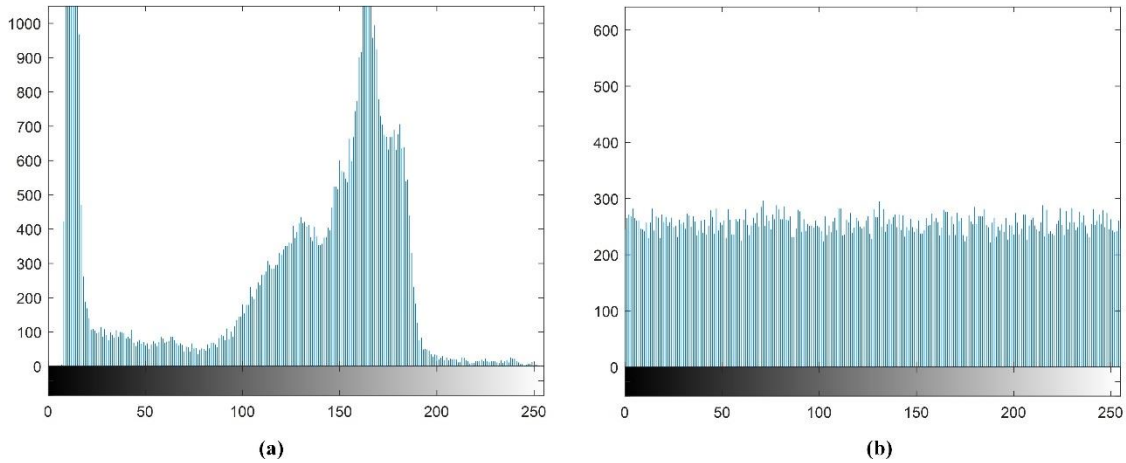

Figure 7. (a) Histogram of the plain image (b) Histogram of the encrypted image

3.5. Correlation Analysis

There is a high correlation between neighboring pixels of the plain image in the horizontal, vertical, and diagonal directions. To evaluate the correlation between the neighboring pixels, a metric which is called the correlation coefficient can be used. The correlation coefficient is expressed as in (11).

$$\rho_{xy} = \frac{Covariance(x,y)}{\sqrt{Variance(x)Variance(y)}} \tag{11}$$

The covariance and variance functions are given in the following equations.

$$Covariance(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(x))(y_i - E(y)) \tag{12}$$

$$Variance(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(x))^2 \tag{13}$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n}x_i \tag{14}$$

where x and y are the values of the neighboring pixels and n is the total number of randomly chosen pixels. The proposed method aims to reduce the correlation between the neighboring pixels.

Table 1. Correlation coefficients of different images

	Correlation coefficients		
	Horizontal	**Vertical**	**Diagonal**
Plain image (Figure 5a)	0.93163	0.96171	0.90667
The image after sub-image shuffling (Figure 5b)	0.92877	0.95400	0.89272
The image after bit-level permutation (Figure 5c)	0.23640	0.09901	0.03908
Encrypted image (Figure 5d)	-0.00421	0.00718	-0.00857

Therefore, the correlation coefficient of the plain image can be around 1 but the correlation coefficient of the encrypted image should be close to 0. Table 1 displays the calculated values of the correlation coefficients of different images given in Figure 5. Each correlation coefficient is computed by selecting randomly 10,000 pairs of neighboring pixels from the regarding the image. The correlation coefficients of the encrypted image are very close to 0 in all directions which means that the encryption method can resist a statistical attack.

Figure 8 illustrates the distribution of two neighboring pixels in all directions for the plain image and the encrypted image for 10,000 randomly chosen pixel pairs. It can be observed from Figure 8a, Figure 8b, and 8c that the neighboring pixels in the plain image are correlated. However, Figure 8d, Figure 8e, and Figure 8f show that the neighboring pixels of the encrypted image are randomly distributed all over the graph.

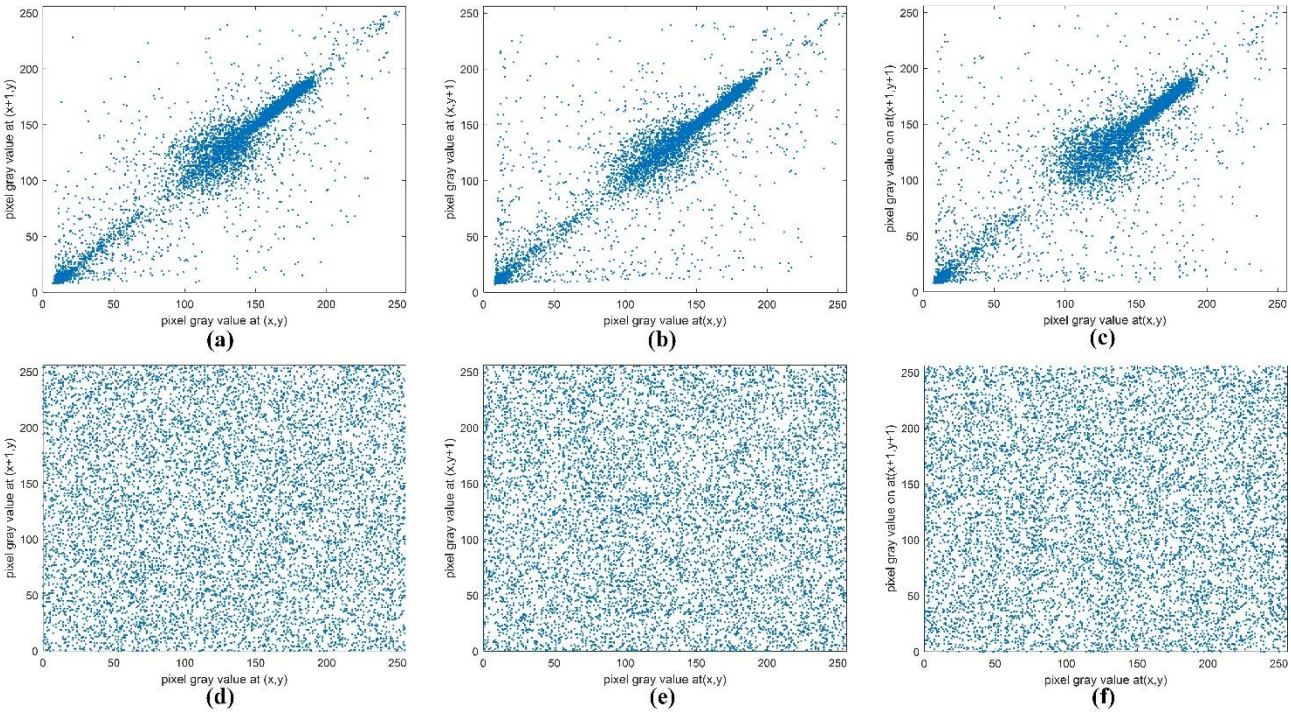

Figure 8. Correlation of neighboring pixels in the plain image: (a) Horizontal (b) Vertical (c) Diagonal. Correlation of neighboring pixels in the encrypted image: (d) Horizontal (e) Vertical (f) Diagonal.

3.6. Differential Analysis

Two evaluation measures can be used to test the encryption method's resistance against differential attacks: Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI). NPCR finds the rate of the different number of pixels between two encrypted images and UACI finds the average difference of intensities between two encrypted images. The encrypted images whose corresponding plain images have only one pixel difference must be obtained using the same secret keys. NPCR and UACI are calculated as in (15) and (16), respectively.

$$NPCR = \frac{1}{M \times N} \sum_{j=1}^{N} \sum_{i=1}^{M} D(i,j) \times 100\ \% \tag{15}$$

$$UACI = \frac{1}{M \times N \times 255} \sum_{j=1}^{N} \sum_{i=1}^{M} |E_1(i,j) - E_2(i,j)| \times 100\ \% \quad (16)$$

where $D(i,j)$ is defined as in (17).

$$D(i,j) = \begin{cases} 0 \text{ if } E_1(i,j) = E_2(i,j) \\ 1 \text{ if } E_1(i,j) \neq E_2(i,j) \end{cases} \quad (17)$$

One pixel is picked randomly from the plain image and its value is changed to create a slightly different plain image. Those two plain images are encrypted using the same secret keys so that two encrypted images E_1 and E_2 are found. This operation is repeated 50 times for different pixels. The average NPCR and UACI values are found as 98.0774% and 33.2384%, respectively. These results show that the proposed method can resist differential attacks because NPCR and UACI values are very close to their optimal values.

4. Conclusions

This paper proposes a new image encryption method that is based on sub-image shuffling, bit-level permutation, and diffusion stages. The sub-image shuffling is performed with the help of a chaotic sequence generated by a logistic map. The tent map is used to permute the highest four bits of the pixels. The diffusion phase employs both chaotic maps to alter the values of the pixels. The encryption method's key space is proven to be large enough to resist brute-force attacks. The key sensitivity is also shown. Even a tiny change in only one key results in a totally different decrypted image. Histogram graph and correlation coefficient calculations show that the pixels of the encrypted image are randomly and uniformly distributed which is a sign of good encryption. Differential analysis confirms that the method is very sensitive to a change in plain image. To sum up, the offered method is secure and efficient and can be applied in field image encryption.

Authors' Contributions

MD developed the algorithm, performed the calculations, made the computer-based simulations, carried out the experimental work, and wrote up the article.

References

[1]. Güvenoğlu, E., Resim Şifreleme Amacıyla Dinamik S Kutusu Tasarımı İçin Bir Yöntem, El-Cezeri Journal of Science and Engineering, 2016, 3(2): 179-191.

[2]. Li, Y., Wang, C., and Chen, H., A hyper-chaos-based image encryption algorithm using pixel level permutation and bit-level permutation, Optics and Lasers in Engineering, 2017, 90, 238-246.

[3]. Muthu, J. S., Murali P., Review of Chaos Detection Techniques Performed on Chaotic Maps and Systems in Image Encryption, SN Computer Science, 2021, 2(5): 1-24.

[4]. Kaur, M., Kumar, V., A Comprehensive Review on Image Encryption Techniques, Archives of Computational Methods in Engineering, 2020, 27 (1), 15-43.

[5]. Kumar, M., Saxena, A., and Vuppala, S. S., Survey on Chaos Based Image Encryption Techniques, In Multimedia Security Using Chaotic Maps: Principles and Methodologies, Springer, Cham, 2020, 1-26.

[6]. Pareek, N. K., Patidar, V., and Sud, K. K., Image Encryption Using Chaotic Logistic Map, Image and Vision Computing, 2006, 24(9): 926-934.

[7]. Ye, G., Image Scrambling Encryption Algorithm of Pixel Bit Based on Chaos Map, Pattern Recognition Letters, 2010, 31(5): 347-354.
[8]. Ismail, S. M., Said, L. A., Radwan, A. G., Madian, A. H., and Abu-Elyazeed, M. F., Generalized Double-Humped Logistic Map-Based Medical Image Encryption, Journal of Advanced Research, 2018, 10, 85-98.
[9]. Li, C., Luo, G., Qin, K., and Li, C., An Image Encryption Scheme Based on Chaotic Tent Map, Nonlinear Dynamics, 2017, 87(1): 127-133.
[10]. Naskar, P. K., Bhattacharyya, S., Nandy, D., and Chaudhuri, A., A Robust Image Encryption Scheme Using Chaotic Tent Map and Cellular Automata, Nonlinear Dynamics, 2020, 100(3): 2877-2898.
[11]. Hua, Z., Zhou, Y., Image Encryption Using 2D Logistic-Adjusted-Sine map, Information Sciences, 2016, 339, 237-253.
[12]. Zheng, J., Liu, L., Novel Image Encryption by Combining Dynamic DNA Sequence Encryption and The Improved 2D Logistic Sine Map, IET Image Processing, 2020, 14(11): 2310-2320.
[13]. Ping, P., Xu, F., Mao, Y., and Wang, Z., Designing Permutation–Substitution Image Encryption Networks with Henon Map, Neurocomputing, 2018, 283, 53-63.
[14]. Ibrahim, S., Alharbi, A., Efficient Image Encryption Scheme Using Henon Map, Dynamic S-Boxes and Elliptic Curve Cryptography, IEEE Access, 2020, 8, 194289-194302.
[15]. Abbas, N. A., Image Encryption Based on Independent Component Analysis and Arnold's Cat Map, Egyptian Informatics Journal, 2016, 17(1): 139-146.
[16]. Hariyanto, E., Rahim, R., Arnold's Cat Map Algorithm in Digital Image Encryption, International Journal of Science and Research, 2016, 5(10): 1363-1365.
[17]. Parida, R. R., Singh, B. K., and Pradhan, C., A Novel Approach for Image Encryption Using Zaslavskii Map and Arnold's Cat Map, In Data Engineering and Intelligent Computing, Springer, Singapore, 2021, 269-282.
[18]. Salleh, M., Ibrahim, S., and Isnin, I. F., Enhanced Chaotic Image Encryption Algorithm Based on Baker's Map, In Proceedings of the 2003 International Symposium on Circuits and Systems, IEEE, ISCAS'03, 2003.
[19]. Stoyanov, B., Kordov, K., Image Encryption Using Chebyshev Map and Rotation Equation, Entropy, 2015, 17(4): 2117-2139.
[20]. Liu, L., Jiang, D., Wang, X., Rong, X., and Zhang, R., 2D Logistic-Adjusted-Chebyshev Map for Visual Color Image Encryption, Journal of Information Security and Applications, 2021, 60, 102854.
[21]. Khan, M., Shah, T., A Literature Review on Image Encryption Techniques, 3D Research, 2014, 5(4): 1-25.
[22]. Mirzaei, O., Yaghoobi, M., Irani, H., A New Image Encryption Method: Parallel Sub-image encryption with hyper chaos, Nonlinear Dynamics, 2012, 67, 557-566.
[23]. Li, Y., Wang, C., and Chen, H., A Hyper-Chaos-Based Image Encryption Algorithm Using Pixel-Level Permutation and Bit-Level Permutation, Optics and Lasers in Engineering, 2017, 90, 238-246.
[24]. Zhu, Z. L., Zhang, W., Wong, K. W., and Yu, H., A Chaos-Based Symmetric Image Encryption Scheme Using a Bit-Level Permutation, Information Sciences, 2011, 181(6): 1171-1186.
[25]. Karawia, A. A., Elmasry, Y. A., New Encryption Algorithm Using Bit-Level Permutation and Non-Invertible Chaotic Map, IEEE Access, 2021, 9, 101357-101368.
[26]. Wong, K. W., Kwok, B. S. H., and Yuen, C. H., An Efficient Diffusion Approach for Chaos-Based Image Encryption, Chaos, Solitons & Fractals, 2009, 41(5): 2652-2663.
[27]. Alvarez, G., Li, S., Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, International Journal of Bifurcation and Chaos, 2006, 16(08): 2129-2151.
[28]. Shahna, K. U., Mohamed, A., A Novel Image Encryption Scheme Using Both Pixel Level and Bit Level Permutation with Chaotic Map, Applied Soft Computing, 2020, 90, 106162.

[29]. Shah, A. A., Parah, S. A., Rashid, M., and Elhoseny, M., Efficient Image Encryption Scheme Based on Generalized Logistic Map For Real Time Image Processing, Journal of Real-Time Image Processing, 2020, 17(6): 2139-2151.
[30]. Kari, A. P., Navin, A. H., Bidgoli, A. M., and Mirnia, M., A New Image Encryption Scheme Based on Hybrid Chaotic Maps, Multimedia Tools and Applications, 2021, 80(2): 2753-2772.
[31]. Wu, Y., Noonan, J. P., and Agaian, S., NPCR and UACI Randomness Tests for Image Encryption, Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), 2011, 1(2): 31-38.
[32]. Zhang, G., Ding, W., and Li, L., Image Encryption Algorithm Based on Tent Delay-Sine Cascade with Logistic Map, Symmetry, 2020, 12(3): 355.
[33]. Wang, X., Chen, S., and Zhang, Y., A Chaotic Image Encryption Algorithm Based on Random Dynamic Mixing, Optics & Laser Technology, 2021, 138, 106837.
[34]. Shengtao, G., Tao, W., Shida, W., Xuncai, Z., and Ying, N., A Novel Image Encryption Algorithm Based on Chaotic Sequences and Cross-Diffusion of Bits, IEEE Photonics Journal, 2020, 13(1): 1-15.
[35]. Kaur, M., Singh, D., Sun, K., and Rawat, U., Color Image Encryption Using Non-Dominated Sorting Genetic Algorithm with Local Chaotic Search Based 5D Chaotic Map, Future Generation Computer Systems, 2020, 107, 333-350.