Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20125-x

A new four-tier technique for efficient multiple images encryption

Khalid M. Hosny[1] · **Sara T. Kamal**[2]

Received: 7 June 2023 / Revised: 30 June 2024 / Accepted: 18 August 2024

Abstract
People transmit millions of digital images daily over various networks, where securing these images is a big challenge. Image encryption is a successful approach widely used in securing digital images while transmitting. Researchers developed different encryption techniques that focus on securing individual images. Recently, encryption of multiple images has gained more interest as an emerging encryption approach. In this paper, we proposed a four-tier technique for multiple image encryption (MIE) to increase the transmission speed and improve digital image security. First, we attached the plain images to create an augmented image. Second, the randomized augmented image is obtained by randomly changing the position of each plain image. Third, we scrambled the randomized augmented image using the zigzag pattern, rotation, and random permutation between blocks. Finally, we diffuse the scrambled augmented image using an Altered Sine-logistic-based Tent map (ASLT). We draw a flowchart, write a pseudo-code, and present an illustrative example to simplify the proposed method and make it easy to understand. Many experiments were performed to evaluate this Four-Tier technique, and the results show that this technique is extremely effective and secure to withstand various attacks.

Keywords Multiple image encryption · Augmented image · Block-scrambling · Altered sine-logistic-based tent map

Abbreviations

ASLT	Altered Sine-Logistic-based Tent map
MIE	Multiple image encryption
DNA	Deoxyribonucleic acid
1D	One dimensional
2D	Two dimensional
NPCR	Number of pixels changes rate
UACI	Unified average changing intensity
DA	Differential attack

✉ Khalid M. Hosny
k_hosny@yahoo.com

[1] Department of Information Technology, Faculty of Computers and Information, Zagazig University, Zagazig, Egypt

[2] Department of Computer Sciences, Faculty of Computers and Information, Assiut University, Assiut, Egypt

Published online: 13 September 2024

1 Introduction

Securing digital images during their transmission via various networks is a big challenge. There are three main approaches for securing digital images: image watermarking [1, 2], image steganography [3], and image encryption [4–6]. Different studies and surveys [7, 8] clearly show that encryption of digital images is preferable over other image security approaches.

Securing digital images using an encryption approach is based on two main processes: the plain input image is encrypted using a key and converted to an unpredictable, unreadable, and unrecognizable image of the same size. The sender transmits the encrypted image via networks. The receiver needs the encryption key to decrypt the encrypted image [9].

In the era of big data, people worldwide need to secure and transmit many digital images daily. Although successful image encryption techniques are available, most are designed to encrypt individual images. Securing n-images using encryption techniques initially designed to secure individual images requires the repetition of the encryption and decryption processes n tines, which is an impractical, time-consuming process. Also, this repetition is risky, where attackers have multiple chances to attack the transmitted images.

Researchers designed algorithms to encrypt and decrypt multiple images. Zhang and Wang [10] presented a pioneering algorithm to encrypt multiple images based on piecewise linear chaotic maps. Again, Zhang and Wang [11] used a chaotic system with DNA encoding and developed an MIE method. Xingyuan et al. [12] proposed an MIE algorithm based on cyclic shift and mixed hash functions, where this algorithm can resist various attacks. Karawia [13] introduced a similar algorithm using the combination of mixed image elements and a 2D economic map. The experiments that were conducted show that this algorithm is secure. Patro and Acharya [14, 15] encrypt multiple images using different methods. Both methods indicate robustness against various common attacks.

Enayatfar et al. [16] employed a DNA sequence and image matrix indexes in an MIE algorithm, where a single image is created by attaching multiple gray-level plain images. Then, the 2D augmented images are transformed into a 1D array. The pixels' positions are permutated using the indexes of the obtained 1D array. Zarebnia et al. [17] used hybrid chaotic systems to construct fast encryption algorithms for multiple grayscale images. They used cyclic shift and XOR operations with the Arnold cat map. This algorithm resists various attacks. In [18], Zhang and his co-authors used coupled map lattices with DNA to design an MIE algorithm. Again, Tanveer ul Haq and Tariq Shah [19] used the DNA with a 12-bit random sequence to develop an MIE algorithm.

Gao et al. [20] used a fractional-order hyperchaotic system in an MIE technique. Again, Gao and his co-authors in [21] proposed an algorithm to encrypt multiple images using single-channel scrambling, where these encryption methods show robustness against common attacks. Zhang and Zhang [22] combined chaos and gene fusion to encrypt multiple images.

We proposed a four-tier technique for MIE to increase encryption effectiveness and make it easier to send numerous images securely. In the first tier, the plain input images ($i = 1$ to n^2) are attached to create the augmented image. Then, in the second tier, each plain image's position is randomly displaced to obtain the randomized augmented image. The augmented image is scrambled in the third tier using a zigzag pattern, rotation, where the augmented image's blocks are randomly displaced. Finally, the scrambled augmented image is diffused in the fourth tier using ASLT [23]. Based on the experiments, the Four-Tier MIE algorithm is effective, secure, and resilient to attacks.

The rest of this paper describes the proposed MIE method in the second section. Experiments, results, and discussion are presented in Sect. 3. The conclusion is presented in Sect. 4.

2 The four-tier encryption algorithm

2.1 Encryption process

Here, we described the proposed Four-Tier MIE algorithm. This algorithm can encrypt n^2 images simultaneously, as n take a value of 2, 3, 4,…. .

2.1.1 Tier #1: augmented image creation

The augmented image consists of n^2 plain images are illustrated in Fig. 1. The images from Image #1 to Image #n are displayed in the first row. The elements of the second row are the images from Image #$(n+1)$ to Image #2n. Similarly, the plain input images are arranged and displayed in the augmented image. In the n^{th} (Last) row, the displayed images start from Image #$(n^2 - n + 1)$ to Image # (n^2).

For easier understanding, we present an illustrative example (Fig. 2) of the value of $n = 4$; the augmented image contains sixteen plain images. Each row contains four images, where Image #1, Image #2, Image #3, and #4 are displayed in the first row. The second row contains the images from Image #4 to Image #8. Similarly, the images from Image #9 to Image #12 are displayed in the third row. In the last row, we displayed the images from Image #13 to Image #16. In Fig. 2, we used various colors just for illustration.

2.1.2 Tier #2: randomized augmented image

We generate a vector R of size, N^2, which contains random numbers from 1 to N^2 with no duplication of numbers. Each plain image's position in the augmented image is assigned

Image 1	Image 2	Image 3	.	.	.	.	.	.	.	.	Image n
Image n+1	Image n+2	Image n+3	.	.	.	.	.	.	.	.	Image 2n
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	Image n^2-n
Image n^2-n+1	Image n^2-n+2	.	.	.	.	.	.	.	.	.	Image n^2

Fig. 1 Creation of the augmented image

Image 1	Image 2	Image 3	Image 4
Image 5	Image 6	Image 7	Image 8
Image 9	Image 10	Image 11	Image 12
Image 13	Image 14	Image 15	Image 16

Fig. 2 An illustrative example of the augmented image. It augmented image creation from 16 images

to a randomly selected element of the vector R, as shown in Fig. 3, where each row of the randomized augmented image contains n images and each column contains n Images.

2.1.3 Tier #3: scrambled augmented image

We decompose the randomized augmented image into blocks with different sizes, where the position of the decomposed blocks is changed using a zigzag pattern, rotation, and random permutation between blocks. The zigzag pattern can change the block's positions by scanning the blocks in the "Z" shape from the top left corner of the image. Figure 4(a) shows the scrambled augmented image.

Image 16	Image 5	Image 9	Image 8
Image 4	Image 15	Image 2	Image 12
Image 10	Image 11	Image 3	Image 6
Image 7	Image 1	Image 14	Image 13

Fig. 3 Randomized augmented image

2.1.4 Tier #4: encrypted augmented image

We performed the diffusion process using ASLT as defined by Eq. (1) to change the pixel value.

$$x_{n+1} = \begin{cases} \frac{4-r}{4}\sin(\pi x_n) + \frac{r}{2}x_n, & x_n < 0.5 \\ (4-r)x_n(1-x_n) + \frac{r}{2}(1-x_n), & x_n \geq 0.5 \end{cases} \quad (1)$$

The map is chaotic for the control parameter $r \in [0,4]$.
A flow chart of the proposed MIE algorithm is displayed in Fig. 5.
The following pseudo-code is used to encrypt K plain images simultaneously.

Step1: the augmented image is divided into blocks with a same size selected by the user. The block (B) takes a value from (16, 32, and 64).

Step 2: compute the block size (BS), and number of blocks (NB) as follows:

$$BS = B^2 \quad (2)$$

$$NB = N_a^2/BS \quad (3)$$

Where N_a is the width and height of the augmented image.

Step 3: generate a random vector with a size equals to NB in the augmented image, and the values are taken from (4,8, 16,…, and B) randomly.

Step 4: then each block is either divided into smaller blocks or kept as it is according to the vector generated in the previous step.

Step 5: the pixel positions in each small block are changed using zigzag pattern and rotation with angle 90° respectively.

Step 6: generate a random vector with a size equals to NB in the augmented image, and the values in the vector start from 1 to NB (no duplication of the values).

Step 7: the position of each block is changed randomly using the vector generated in the previous step.

Step 8: a vector (CV) is generated from the ASLT map with the same size of the augmented image (The equation (1) is iterated $N_a^2 + N_0$ and then N_0 numbers are ignored), where the initial condition of the map is calculated by:

$$x_0 = \frac{\sum_{i=1}^{N_a}\sum_{j=1}^{N_a} AI(i,j)}{N_a^2 \times 255} \quad (4)$$

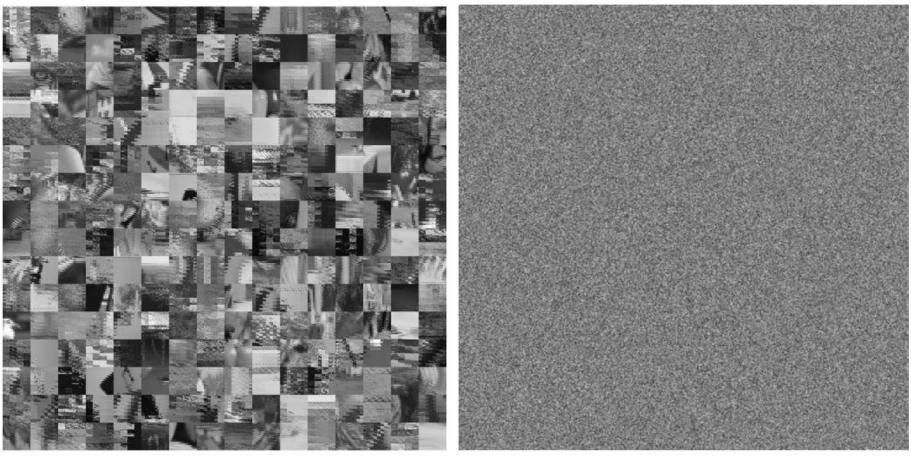
Step 9: Key generation (CV vector values are converted into range [0:255] using)

$$K(i) = mod(floor(CV(i) \times 10^{14}), 256), \quad i = 1{:}N_a^2 \quad (5)$$

Step 10: the augmented image is converted into vector (AIV).

Step 11: XOR operation is applied between AIV and K.

Step 12: the vector generated from step 11 is converted into matrix (the encrypted augmented image).



(a) (b)

Fig. 4 **a** Scrambled augmented image. **b** Encrypted augmented image

2.2 Decryption process

The decryption process restores the augmented image using the encrypted one by performing the reverse steps of the encryption process. First, diffusion is performed by applying an XOR operation between the encrypted augmented image and the key. The key is generated using the initial condition of the ASLT map. Then, confusion steps are done to retrieve the augmented image as follows:

1. The diffused image is decomposed into blocks as in encryption steps.
2. The blocks are rearranged using the encryption algorithm's vector generated in step 6.
3. Each block is split into smaller blocks or kept as it is using the vector generated in step 3 in the encryption algorithm.
4. Apply the reverse of rotation and a zigzag pattern for each small and undivided block.

Finally, the augmented image is restored. Figure 6 displays the flowchart of the decryption process.

3 Experiments

Various experiments were conducted to evaluate the performance of the Four-Tier MIE algorithm. Four test images with a size of 512×512 shown in Fig. 7, selected from the USC-SIPI image database, are used for all security tests. In the experiment, we used four images to create the augmented image, as shown in Fig. 8(a). Figure 8(b) displays the encrypted augmented image. In our experiments, Matlab R2021b is used to create the source code, executed using a computer machine equipped with Intel$^{(R)}$, Core$^{(TM)}$ i7-1165G7 @ 2.80 GHz CPU, and 16 GB RAM with Windows 10 operating system.

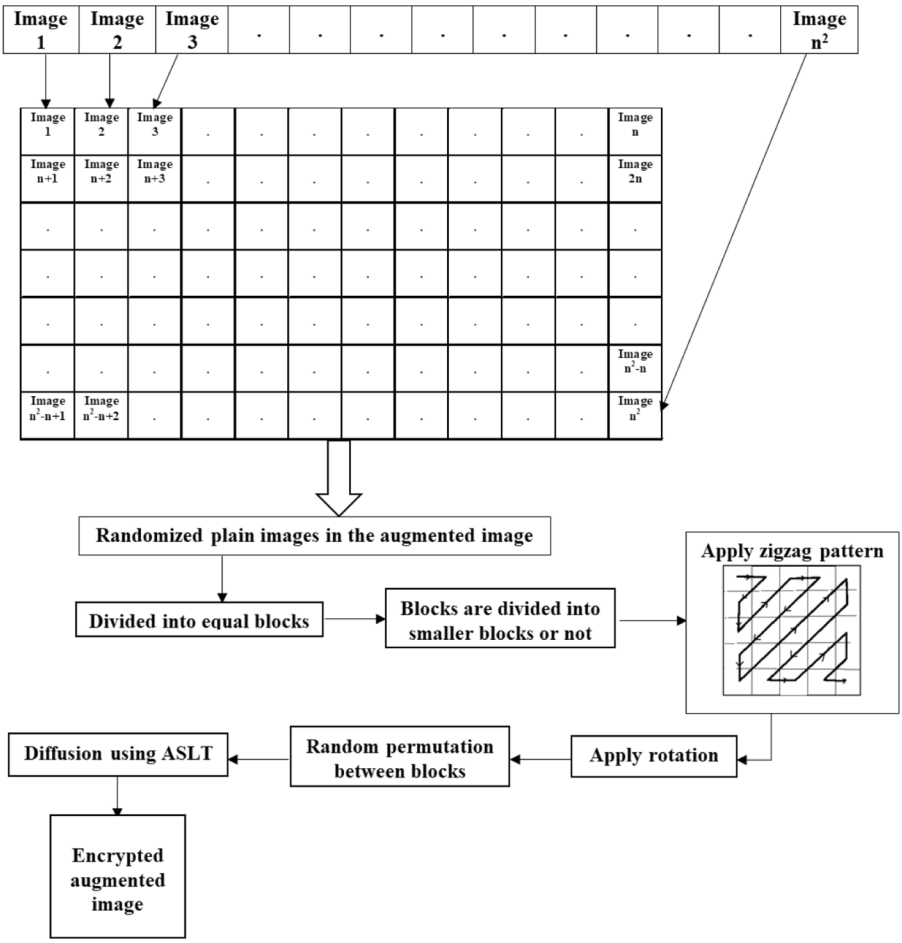


Fig. 5 Flow chart of the proposed MIE algorithm

3.1 Uniformity of the histogram

A histogram graphically represents the frequency of pixels in an image. The uniformity of the histogram means all pixel values have the same frequency. It is an indicator of the success of the image encryption algorithm. Figure 9 shows the histograms of the original and the encrypted augmented image. The histograms of the four test images are completely different from the histogram of the encrypted augmented image. We notice that the histograms of the plain images are non-uniform, while the histogram of the encrypted augmented image is uniform. Therefore, our proposed algorithm altered the original images' features.

The histogram's variance is defined using the following equation:

$$Var(Y) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} (Y_i - Y_j)^2 \tag{6}$$

It calculates the uniformity of encrypted images` histograms. Based on Eq. (6), the variance of the encrypted image should be less than the variance of the original image.

We calculated the variances of all test images and the encrypted image; their values are listed in Table 1. Quantitatively, we proved that our algorithm guaranteed to generate an encrypted image with a consistent histogram.

4 Correlation of adjacent pixels

Generally, the correlation between neighboring pixels in the plain image is high. The adjacent pixel correlation in the encrypted image should be close to zero. The correlation between adjacent pixels is computed as follows:

$$r_{x,y} = \frac{Q((x - Q(x))(y - Q(y)))}{\sqrt{H(x)H(y)}} \tag{7}$$

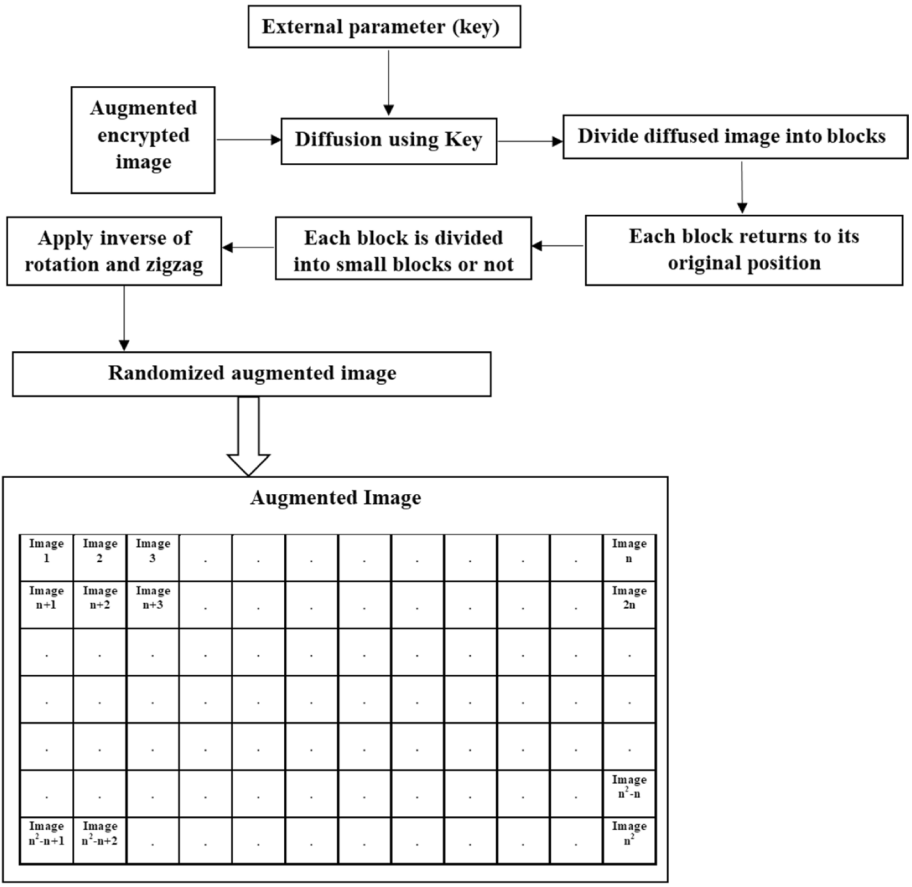
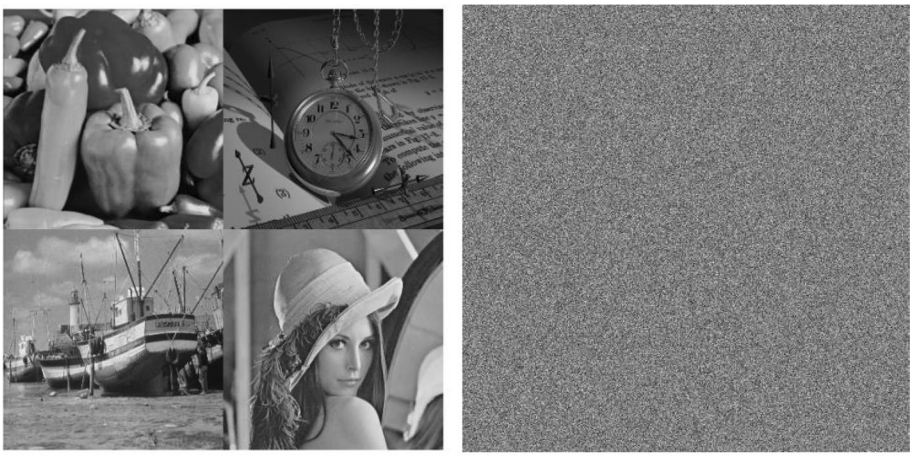


Fig. 6 Flow chart of the decryption process

(a) Lena (b) Peppers (c) boat (d) Watch

Fig. 7 Test images



(a) (b)

Fig. 8 **a** Augmented Image. **b** Encrypted augmented Image

where

$$Q(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \ , \ H(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - Q(x))^2 \tag{8}$$

The x and y are the values of two neighboring pixels, and N is the number of pixels selected from the image. Using our proposed algorithm, Table 2 shows the correlation of pixels in the augmented encrypted image in three directions (Vertical, Diagonal, and Horizontal). Then, we compared the results with recent works [24, 17, 19, 25, 16] and showed these results in Figs. 10 and 11, respectively. We can see that the Four-Tier method reduces the correlation of adjacent pixels in the encrypted augmented image.

4.1 DA analysis

We make a small modification in the plain image and then encrypt it with the proposed algorithm. After that, the modified and original files are encrypted and compared to crack

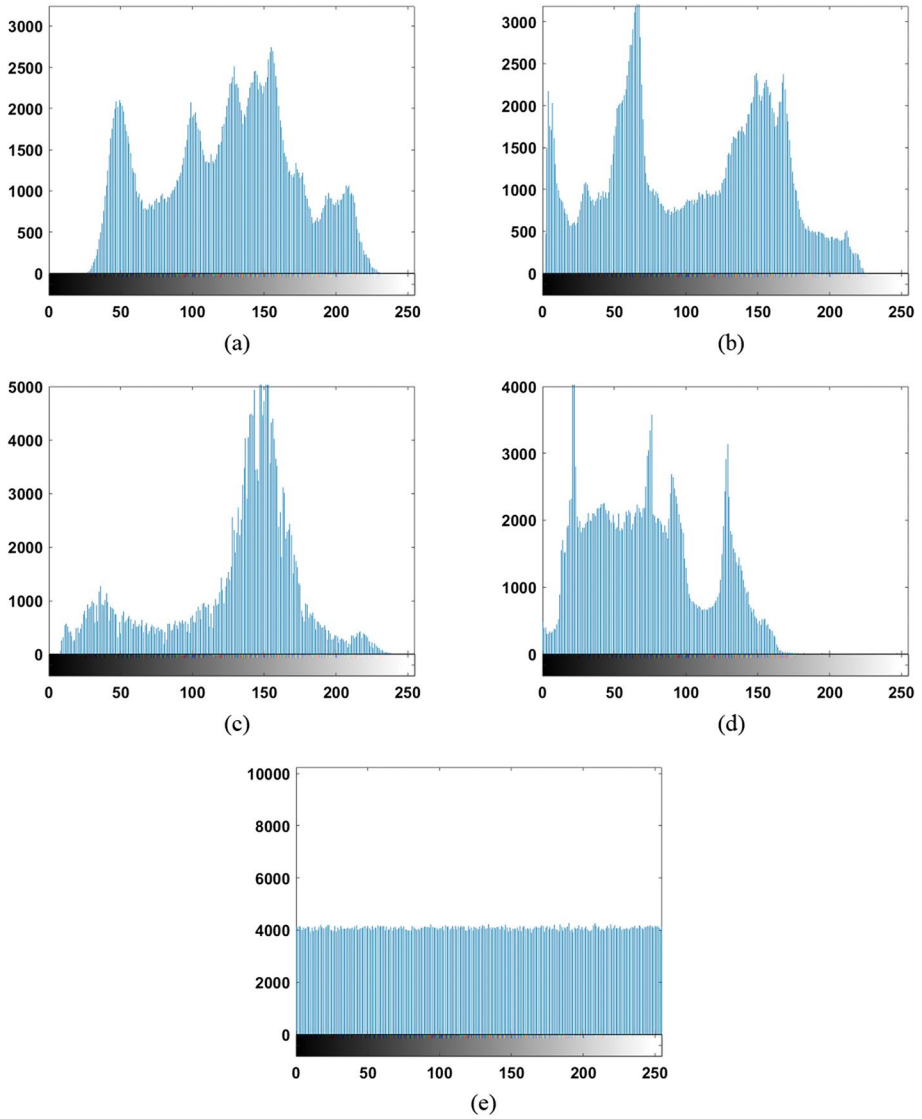


Fig. 9 Histogram analysis of the test images: **a** Lena. **b** Peppers. **c** Boat. **d** Watch. **e** Encrypted augmented image

the algorithm. Thus, Small changes in the plain image should make a big difference in the encrypted image. The NPCR and the UACI measure the ability of the algorithm to resist a DA that is calculated between the two images im_1 and im_2 by:

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i=1}^{N} \sum_{j=1}^{M} D(i,j) \times 100(\%) \tag{9}$$

Table 1 The variance of the histogram

Image	variance
Lena	6.3588e+05
Peppers	5.7631e+05
Boat	1.5419e+06
Watch	1.3498e+06
Encrypted	3578

where M & N refer to the width & height of the image:

$$\mathrm{D(i,\ j)} = \begin{cases} 0 \text{ if } \mathrm{im_1(i,\ j)} = \mathrm{im_2(i,\ j)} \\ 1 \text{ if } \mathrm{im_1(i,\ j)} \neq \mathrm{im_2(i,\ j)} \end{cases} \tag{10}$$

$$\mathrm{UACI} = \frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{\mathrm{j}=1}^{\mathrm{M}} \frac{\left|\mathrm{c_1(i,j)} - \mathrm{c_2(i,j)}\right|}{255} \times 100(\%) \tag{11}$$

Here, a random pixel is changed in the plain image. We encrypt the plain and the changed plain images using the Four-Tier encryption technique and calculate UACI & NPCR. The calculated values for the proposed Four-Tier and the recent existing encryption techniques [24, 17, 19, 25, 16] are shown in Table 3. According to [26], the algorithm can resist this attack if the NPCR & UACI are 99.61% & 33.46%, respectively. Based on the results in Table 3, we conclude that the proposed algorithm is superior to other existing techniques.

4.2 Entropy

The entropy reflects the randomness of an image. The ideal value for an encrypted image is 8, which is calculated from the following equation:

$$H(s) = -\sum_{i=1}^{k} P(s_i) \log_2 P(s_i) \tag{12}$$

We calculate, $H(s)$, the encrypted augmented image and the results are shown in Table 4. The entropy value of the proposed algorithm is similar to that of the algorithm in [24] and better than all other algorithms, which indicates high randomness.

4.3 Noise attack

The efficiency of the encryption algorithm depends on its ability to recover the attacked encrypted image with high quality. Images may be contaminated with noise to noise during transmission over channels. Figures 12 and 13 show the decrypted images after adding salt and pepper noise and Gaussian noise, respectively, with different intensities to the encrypted image. All the contents of the decrypted images are visible.

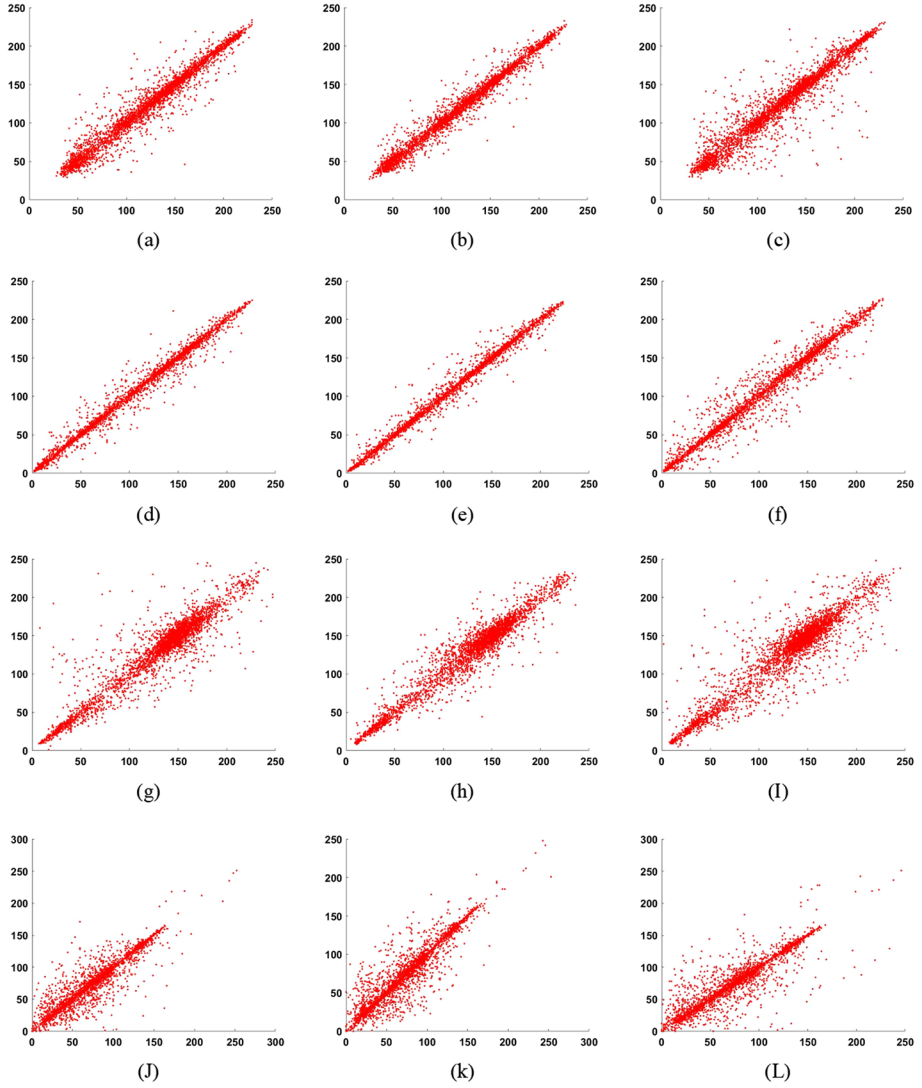


Fig. 10 Correlation coefficient analysis. **a** Lena #Horizontal (**b**) Lena #Vertical (**c**) Lena #Diagonal (**d**) Peppers #Horizontal (**e**) Peppers #Vertical (**f**) Peppers #Diagonal (**g**) Boat #Horizontal (**h**) Boat #Vertical (**I**) Boat #Diagonal (**J**) Watch #Horizontal (**k**) Watch #Vertical (**L**) Watch #Diagonal

4.4 Data cut attack

Resistance to data cut attack is tested by removing parts from the encrypted image. In this experiment, the encrypted image is attacked with 10%, 20%, 30%, and 40% cropping, as shown in Fig. 14 (a-d). Then, the proposed algorithm decrypts the attacked encrypted images; the results are shown in Fig. 14 (e-h). The pixel values in the cropped part are zero. Increasing the cropped part's size decreases the decrypted image's quality. Though large

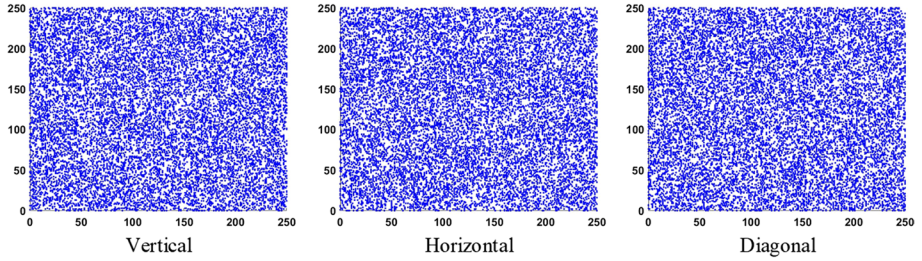


Fig. 11 Correlation of the encrypted augmented image

Table 2 The correlation coefficient of the encrypted augmented image in the three directions

Method	H	V	D
Proposed	-0.0031	0.0016	-0.0081
[24]	-.0003	0.0011	0.0013
[17]	-0.0036	0.0026	0.0012
[19]	-0.0016	0.0057	-0.0189
[25]	0.0034	0.0015	0.0008
[16]	0.0009	0.0016	0.0007

Table 3 UACI & NPCR of different methods

Method	NPCR	UACI	Deviation error	
			NPCR	UACI
Proposed	99.6102	33.4691	-0.0002	-0.0091
[24]	99.6060	33.5126	0.004	-0.0526
[17]	99.5907	33.4811	0.0193	-0.0211
[19]	99.6250	33.4510	-0.015	0.009
[25]	99.1841	33.5284	0.4259	-0.0684
[16]	99.6534	33.6772	-0.0434	-0.2172

parts are removed in Fig. 14 (g and h), the contents of the images are still visible. Thus, the proposed algorithm succeeds in resisting data-cut attacks.

4.5 JPEG compression attack

The efficiency of the proposed algorithm in restoring the compressed, encrypted augmented image is presented in this section. The encrypted augmented image in Fig. 8b is compressed with different quality factors (20, 40, and 60). We apply our proposed algorithm to restore the compressed, encrypted augmented image, and the results are shown in Fig. 15(a-c). As shown in Fig. 15, the content of the decrypted images is recognized. Thus, our proposed algorithm is highly robust against JPEG compression attacks.

Table 4 Entropy values of the encrypted image using different methods

Method	Entropy
Proposed	7.9998
[24]	7.9998
[17]	7.9995
[19]	7.9994
[25]	7.9996
[16]	7.9994

4.6 Key sensitivity

Efficient encryption algorithms must be very sensitive to every small change in the key. To test the key sensitivity of the proposed algorithm, we modified the initial condition, x_0

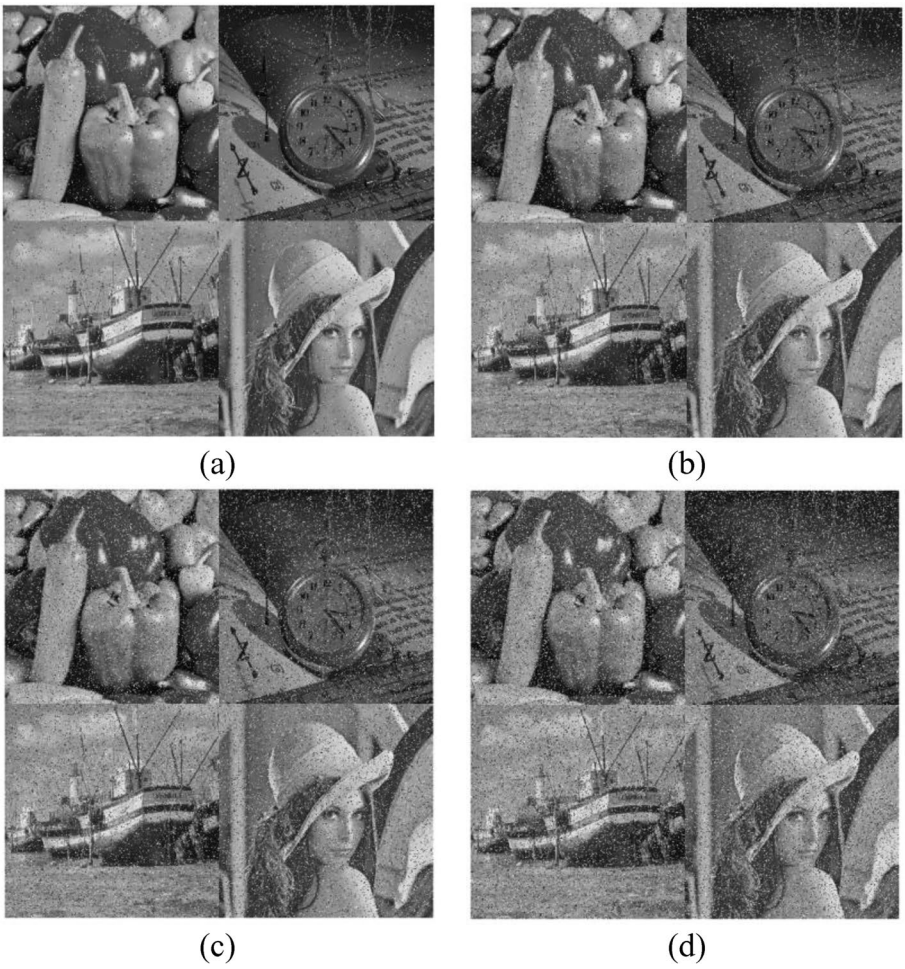


Fig. 12 Salt and pepper noise attack. **a** Decrypted image after adding 0.05 salt and pepper noise. **b** Decrypted image after adding 0.1 salt and pepper noise. **a** Decrypted image after adding 0.15 salt and pepper noise. **a** Decrypted image after adding 0.2 salt and pepper noise

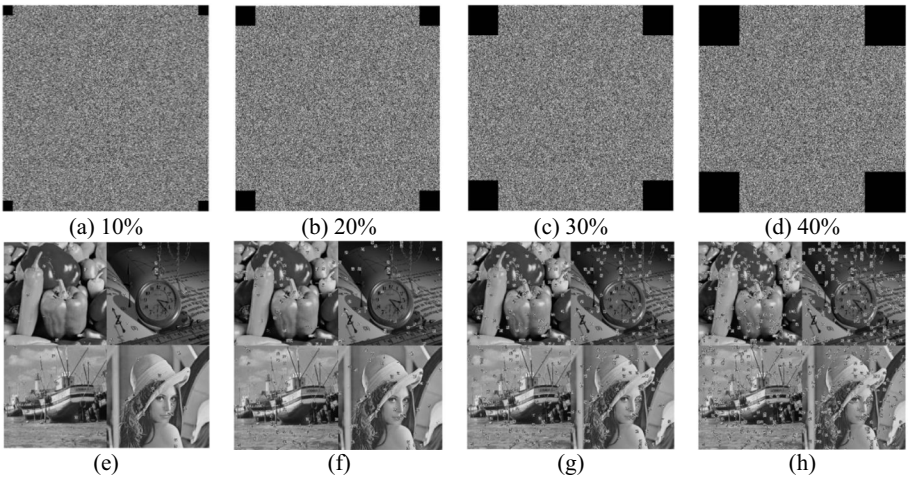


Fig. 13 Data cut attack. **a-d** cropped encrypted images with 10, 20, 30, and 40 %, respectively. **e-f** Decrypted images

of the ASLT map to be $x_0 + 10^{-14}$. Then, these two values are used to generate two keys (k_1, *and* k_2). The original key (k_1) is generated using the initial condition x_0, and the modified key (k_2) is generated using the initial condition $x_0 + 10^{-14}$. We decrypt the encrypted image with the two keys (k_1, *and* k_2) and the decrypted images are shown in Fig. 16. The decryption result using k_2 is unacceptable; therefore, the proposed algorithm is very sensitive to any small change in the key.

4.7 Keyspace

To avoid brute force attacks, the key space of encryption algorithms should be larger than 2^{100}. In this attack, attackers try all possible keys until they reach the correct one. The larger the key space, the harder it is for attackers to predict the correct key. In this algorithm, the key space is based on the initial condition x_0, the control parameter r, and the initial iteration number $N_0 = 10^3$ of the map. Thus, the total key space is 10^{35} that is large enough to resist brute force attack.

4.8 Encryption quality

The quality of the encrypted image is measured by mean square error (MSE) and peak signal-to-noise ratio (PSNR). The larger the value of the MSE, the greater the difference between the original image (I_O) and the encrypted image (I_E). Also, a small value of the PSNR confirms that the encrypted image is completely different from the original image. MSE and PSNR are defined by:

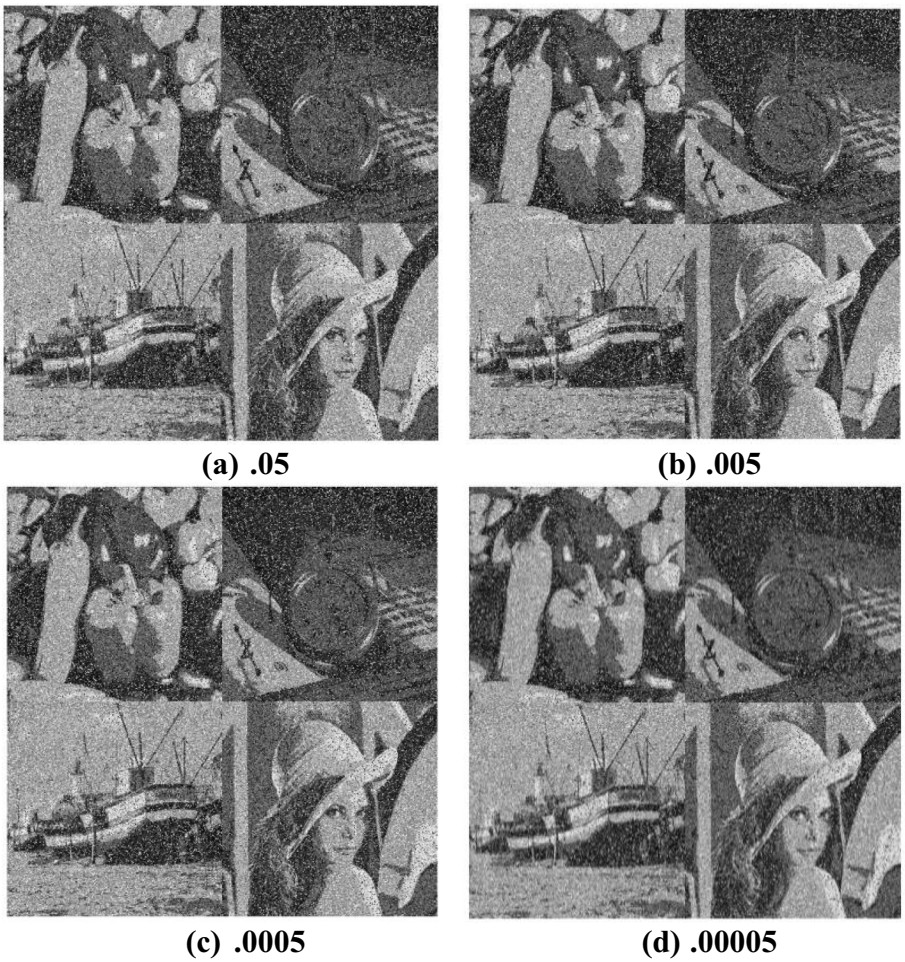


Fig. 14 Decryption images with different white Gaussian noise intensities

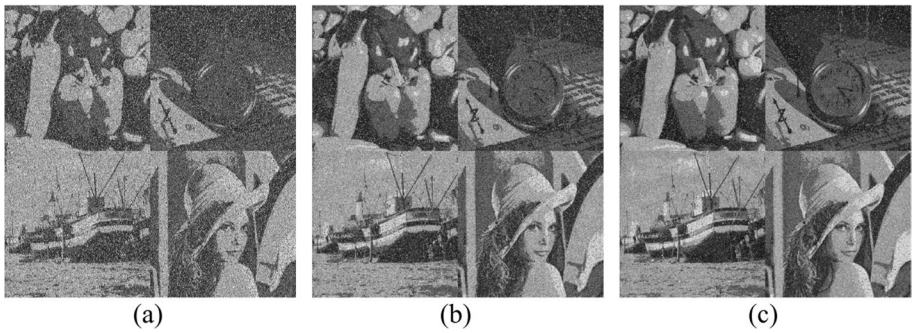


Fig. 15 JPEG compression attack: **a** After the JPEG was compressed with the quality factor 20, the image was decrypted. **b** After the JPEG was compressed with the quality factor 40, the image was decrypted. **c** The image was decrypted after the JPEG was compressed with a quality factor of 60

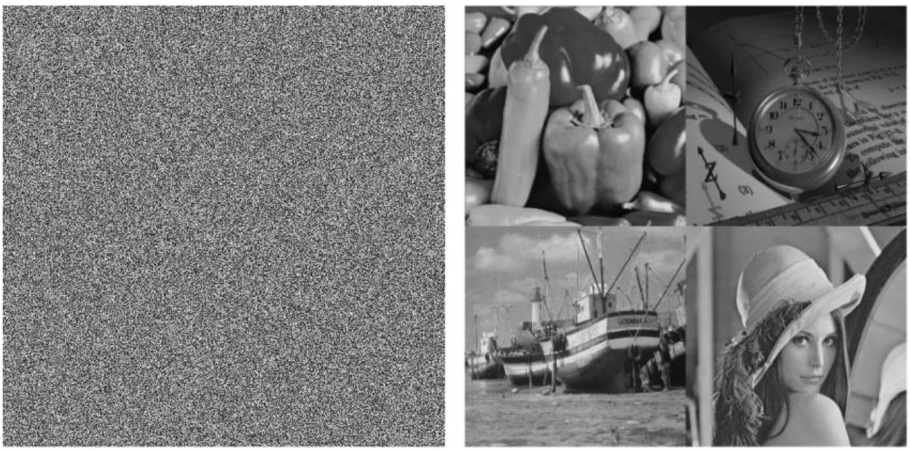


Fig. 16 key sensitivity. **a** Decrypted image with modified key (k_2). **b** decrypted image with original key (k_1)

Table 5 Encryption quality analysis

Method	MSE	PSNR
Proposed	8708	8.7317
[24]	8522	8.8260

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \left| \mathrm{I}_O(\mathrm{i},\mathrm{j}) - I_E(\mathrm{i},\mathrm{j}) \right|^2 \tag{13}$$

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right) \ (db) \tag{14}$$

Table 5 shows the MSE and PSNR values between the original and encrypted images using the proposed algorithm and the algorithm in [24]. We can see that the proposed algorithm has excellent encryption quality.

4.9 Discussion

Our proposed method used a four-tier technique for MIE to send several images securely and to increase encryption effectiveness. Our proposed algorithm can encrypt multiple images simultaneously instead of encrypting one image. Also, in the confusion step of the augmented image, all pixels of different images are confused instead of scrambling pixels of the same image. This step increases the randomization of pixels in the image as the pixels come from different images. The ASLT is used to change the pixel values (diffusion). When applied to the confused image, the diffusion process achieves a higher security level. We perform ten tests and achieve the ideal values of (entropy, NPCR, and UACI) for the encrypted image. Also, we can resist different

kinds of attacks (noise, data cut, and JPEG compression). Regarding the key space, our proposed method can resist brute force attacks as the key space is larger than 2^{100}.

Compared with other MIE methods [16, 17, 19, 24, and 25] the proposed algorithm has the higher entropy value of the encrypted image. Also, the deviation error of the NPCR and UACI of the proposed method is lower than that of other methods.

5 Conclusion

This paper presented a new efficient technique to encrypt multiple images simultaneously. The proposed algorithm is based on four tiers in tier #1; the plain images are located in the augmented image from the upper left corner row by row. Tier #2, the position of each plain image in the augmented image is randomized. In tier #3, the augmented image is divided into blocks; scrambling is performed using a zigzag pattern, rotation, and random permutation between blocks. Tier #4, the ASLT map, changes the value of pixels in the scrambled augmented image. Our proposed algorithm is compared with other recent methods, and the results prove that the proposed algorithm can encrypt multiple images with high efficiency. Also, the key space of the proposed algorithm is large enough to resist brute-force attacks.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). No funding is available for all authors.

Data availability Data will be available upon request.

Declarations

Conflict of interests No conflict of interests.

References

1. Hosny KM, Darwish MM, Fouda MM (2021) Robust color images watermarking using new fractional-order exponent moments. IEEE Access 9:47425–47435
2. Hani Alquhayz, Raza B (2022) Watermarking techniques for the security of medical images and image sequences. Arab J Sci Eng 47:9471–9488
3. Aiman Jan SA, Parah M, Hassan, Bilal A, Malik (2023) Realization of efficient Steganographic Scheme using hybrid edge detection and Chaos. Arab J Sci Eng 48:1859–1872
4. Hosny KM, Kamal ST, Darwish MM, Papakostas GA (2021) New image encryption algorithm using hyperchaotic system and Fibonacci q-matrix. Electronics 10(9):1066. https://doi.org/10.3390/electronics10091066

5. Hosny KM, Kamal ST, Darwish MM (2022) Novel encryption for color images using fractional-order hyperchaotic system. J Ambient Intell Humaniz Comput 13(2):973–988
6. Gabriel Armand Gakam Tegue, Nkapkop JDD, Abdel MA, Tsafack N, Musheer A, Signing FV (2023) Joseph Yves Effa, and Jean Gaston Tamba, a novel image encryption Scheme combining a dynamic S-Box generator and a New Chaotic Oscillator with Hidden Behavior. Arab J Sci Eng. https://doi.org/10.1007/s13369-023-07715-x)
7. Kaur M, Kumar V (2020) A comprehensive review on image encryption techniques. Arch Comput Methods Eng 27:15–43
8. Magdy M, Hosny KM, Ghali NI, Ghoniemy S (2022) Security of medical images for telemedicine: a systematic review. Multimedia Tools Appl 81(18):25101–25145
9. Hosny KM, Kamal ST, Darwish MM (2023) A novel color image encryption based on fractional shifted Gegenbauer moments and 2D logistic-sine map. Visual Comput 39:1027–1044
10. Zhang X, Wang X (2017) Multiple-image encryption algorithm based on mixed image element and chaos. Comput Electr Eng 62:401–413
11. Zhang X, Wang X (2019) Multiple-image encryption algorithm based on DNA encoding and chaotic system. Multimed Tools App 78:7841–7869
12. Wang Xingyuan Z, Xiaoqiang W, Xiangjun Z, Yingqian (2018) Image encryption algorithm based on multiple mixed hash functions and cyclic shift. Opt Lasers Eng 107:370–379
13. Karawia AA (2018) Encryption Algorithm of Multiple-Image Using Mixed Image Elements and Two Dimensional Chaotic Economic Map. Entropy 20(10):801. https://doi.org/10.3390/e20100801
14. K Abhimanyu Kumar Patro and Bibhudendra Acharya (2018) Secure multi-level permutation operation based multiple colour image encryption. J Inform Secur Appl 40:111–133
15. Patro KAK, Acharya B (2020) A novel multi-dimensional multiple image encryption technique. Multimed Tools App 79:12959–12994
16. Rasul Enayatfar FG, Guimarães, Siarry P (2019) Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt Lasers Eng 115:131–140
17. Zarebnia M, Pakmanesh H, Parvaz R (2019) A fast multiple-image encryption algorithm based on hybrid chaotic systems for grayscale images, Optik. 179:761–773
18. Zhang H, Wang Xiao-qing, Wang Xing-yuan, Peng-fei Y (2020) Novel multiple images encryption algorithm using CML system and DNA encoding, IET Image Processing, pp. 518–529
19. Tanveerul Haq and Tariq Shah (2020) Algebra-chaos amalgam and DNA transform based multiple digital image encryption. J Inform Secur Appl, 54, Article 102592
20. Gao X, Yu J, Banerjee S, Huizhen Yan, and, Mou J (2021) A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion. Sci Rep, 11, Article 15737
21. Xinyu Gao J, Mou L, Xiong Y, Sha HY, Cao Y (2022) A fast and efficient multiple images encryption based on single-channel encryption and chaotic system. Nonlinear Dyn 108:613–636
22. Zhang X, Zhang L (2022) Multiple-image encryption algorithm based on chaos and gene fusion. Multimed Tools App 81:20021–20042
23. Aouissaoui I, Bakir T, Sakly A, Femmam S (2022) Improved one-Dimensional Piecewise Chaotic maps for Information Security. J Commun 17(1):11–16
24. Zhang X, Hu Y (2021) Multiple-image encryption algorithm based on the 3D scrambling model and dynamic DNA coding. Opt Laser Technol 141:107073
25. Ye H, Zhou N, Gong L (2020) Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion. Signal Process 175:107652–107666
26. Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J Sel Areas Telecommunications (JSAT), 31–38