Multimedia Tools and Applications (2023) 82:19969–19991
https://doi.org/10.1007/s11042-022-14160-9

Multiple-image encryption algorithm based on the bit plane and superpixel

Xiaoqiang Zhang[1] · Tiancong Gao[1]

Received: 14 January 2022 / Revised: 7 April 2022 / Accepted: 27 October 2022 /
Published online: 25 November 2022

Abstract
Image security is becoming more and more important in recently years. To improve the efficiency and security, this paper defines the concepts of the superpixel and super image, and proposes a multiple-image encryption (MIE) algorithm based on the bit plane and superpixel. The superpixel is an integer formed by connecting the binary values of multiple pixels head to tail and then converting them into a decimal number. The proposed algorithm adopts the classical scrambling-diffusion structure. At the scrambling stage, our algorithm uses the scrambling operation among bit planes and the extended Zigzag transformation in the bit plane. At the diffusion stage, our algorithm performs the exclusive OR operation on the scrambled super image. Different from other MIE algorithms, our algorithm can encrypt multiple images with the workload of processing one image. The experiments and comparative analysis, i.e., the results such as key space, differential attack, etc., show that the proposed algorithm has excellent encryption efficiency and high security.

Keywords Image security · MIE · Bitplane · Pixelbit depth · Superpixel

1 Introduction

With the rapid development of digital technology and the wide use of multimedia acquisition tools, the image, video, audio and other multimedia data become the main carriers of information dissemination, which play an irreplaceable role in many applications. A large number of images and videos are transmitted in Internet. In our daily life, the image is the commonly used multimedia. For example, some scholars use images for information retrieval [7, 17, 18, 20]. The security incidents, such as information leakage, content tampering and content forgery emerge one after another. Scholars have studied the technologies of image

✉ Xiaoqiang Zhang
grayqiang@163.com

[1] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China

watermarking, image steganography, image encryption, etc., to protect the image security. The technology of image watermarking is an essential tool for protecting the multimedia copyright. The watermark, imperceptible information such as ownership identities, serial numbers, and secret words, is embedded into the host signal, and it could be extracted later at the decoder side for the proof of the ownership or other applications [35, 36, 38, 42, 47, 48]. Image steganography and image encryption are technologies to protect the security of image content. The technology of image steganography is embedding the secret data in another innocuous data/object in such a way that only the sender and intended recipient are aware of the secret existence [2, 15, 19]. However, this technology hides a limited capacity. The technology of image encryption is converting the original image into a noise-like image by the scrambling or diffusion operations.

The digital image has the inherent characteristics of large data capacity, strong correlation between pixels and high redundancy. It is inefficient to encrypt images with the traditional encryption algorithms, such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA) and RSA [14]. The reason why chaotic image encryption has become so popular is that chaotic systems have some characteristics that are very suitable for image encryption, such as extreme sensitivity to initial values, unpredictability, pseudorandomness, and ergodicity [6, 22, 49]. Since 1989, Matthews proposed that chaotic systems can be used in cryptography [24]. Since then, many scholars have devoted themselves to the research of encryption schemes based on the chaotic system. After decades of development, various excellent image encryption schemes based on the chaotic system have been proposed in recent years [21, 26, 28]. Most of encryption algorithms follow the scramble-diffusion architecture. In the scrambling process, pixels are shuffled from their positions. Thus, the visual perception of the original image gets changed. Chaotic maps such as Arnold Cat map, Logistic map, and Hénon map are some scrambling techniques [8, 32, 39]. In the diffusion process, the intensity values of the pixels are changed. Mostly chaotic maps and some operations like the addition, subtraction, and exclusive OR (XOR) are used to diffuse the original image to remove its statistical information [1]. With the continuous expansion of image data, traditional single-image encryption (SIE) algorithms are difficult to meet the requirements of security and efficiency in practical applications. The multiple-image encryption (MIE) algorithm has gradually attracted researchers' attention. The main difference between the MIE algorithm and SIE algorithm is that the MIE algorithm makes full use of the characteristics of multiple-image data and encrypts multiple images in batch at the same time. MIE algorithms have greatly improved the efficiency and security of simultaneous transmission for multiple images. Some scholars have proposed many MIE algorithms [3, 4, 10, 16, 23]. For example, Chen et al. proposed an MIE algorithm based on DNA coding [5]. This algorithm has high security, but due to the DNA encoding and decoding operations, its encryption efficiency can be further improved. Song et al. proposed an MIE algorithm based on the cascaded fractional Fourier transform [13]. This algorithm is designed in the transform domain, and combined with the compression technology. However, the decrypted image has obvious distortion. Tang et al. proposed an MIE algorithm based on bit planes and chaotic images [34]. This algorithm has a good encryption effect, but it can only encrypt 4 images at once. Zhang et al. proposed two MIE algorithms based on mixed image elements [43, 44]. These two algorithms have high security and efficiency. However, when the size of the mixed image elements is large, they have the blocking effect to a certain extent. In total, the current MIE algorithms exist some drawbacks, such as the low efficiency and weak security.

To improve encryption efficiency and security and increase encryption capacity, this paper proposes an MIE algorithm based on the bit plane and superpixel. The main contributions are listed as follows: (1) To reduce the amount of encrypted data, we define the concepts of super image and superpixel by connecting the binary values of multiple pixels head to tail; (2) This paper designs a new MIE algorithm based on the bit plane and superpixel, which encrypts plain images with the unit of superpixels, not the commonly used 8-bit pixels; (3) Experiments and comparative analyses, show that our algorithm is feasible, secure and efficient.

The rest of the paper are organized as follows. Theoretical principles related to the new MIE algorithm are given in Section 2. Section 3 proposes the new MIE algorithm. Experiments are carried out in Section 4. Section 5 evaluates the performance of the proposed algorithm. Finally, Section 6 draws the conclusions of the whole paper.

2 Theoretical principles

2.1 Sine-tent system

The chaotic system used in this paper is the Sine-Tent piecewise chaotic System (STS) proposed in Ref. [50]. Its mathematical model is shown in Eq. (1).

$$x_{i+1} = \begin{cases} \frac{4-\mu}{4} \times \sin(\pi x_i) + \frac{\mu}{2} \times x_i & x_i < 0.5 \\ \frac{4-\mu}{4} \times \sin(\pi x_i) + \frac{\mu}{2} \times (1-x_i) & x_i \geq 0.5 \end{cases} \tag{1}$$

where μ is the control parameter. When $\mu \in [0, 4]$, this system is in a chaotic state. Figure 1 is the phase diagram of the STS (the initial vale: $x_0 = 0.2147$). When the parameter μ is changed in a large range, the output of the system shows excellent chaos. In addition, in a nonlinear chaotic system, the Lyapunov Exponent (LE) λ can characterize the speed of convergence or

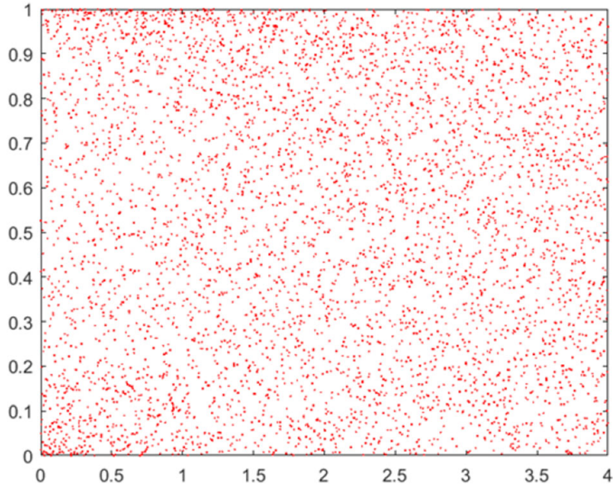


Fig. 1 Phase diagram of the STS

divergence of the system trajectory [33]. According to the definition of LE, the LE of the STS is calculated. When $\lambda > 1$, even if the initial conditions of the system change very slightly, the output will vary greatly with the iteration, and the system shows chaotic characteristics. If a chaotic system has two LE values and they are greater than 0, this system is a hyperchaotic system. Figure 2 is the LE spectrum of the STS, and it shows that when the chaotic parameters are changed, the LE is greater than 0. The STS has a large LE value and an extremely wide range of chaos. Therefore, the STS can iterate complex chaotic trajectories, and exhibit extremely strong unpredictability.

2.2 Extended zigzag transformation

The traditional Zigzag transformation is used for the square matrix [37]. The Zigzag transformation is a procedure to scan the elements of a matrix following the 'Z' shape and store the scanned elements into a one-dimensional (1D) array sequentially. Then, the 1D array can be rearranged as a two-dimensional (2D) matrix according to specific requirements. As shown in Fig. 3, the 4×4 matrix starts from the upper-left corner and ends at the lower-right corner.

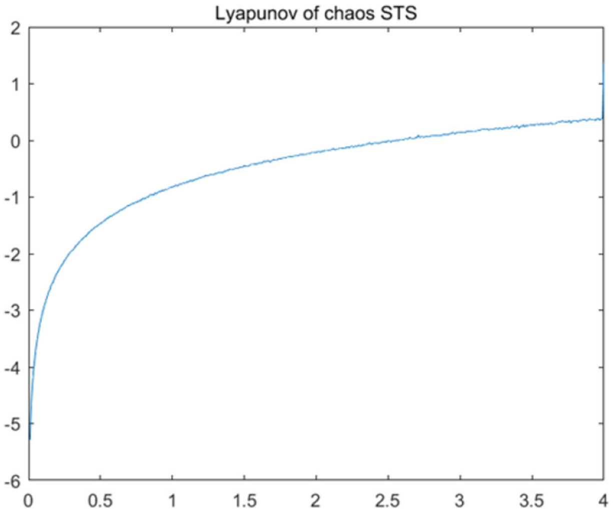
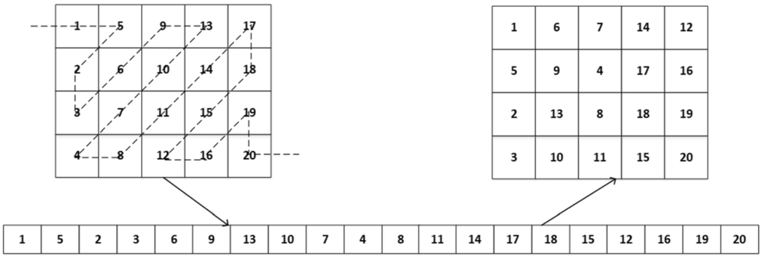


Fig. 2 LE diagram of the STS



1	5	9	13	17
2	6	10	14	18
3	7	11	15	19
4	8	12	16	20

1	6	7	14	12
5	9	4	17	16
2	13	8	18	19
3	10	11	15	20

1	5	2	3	6	9	13	10	7	4	8	11	14	17	18	15	12	16	19	20

Fig. 3 Traditional Zigzag transformation for the 4×5 matrix

Therefore, we can extend the traditional Zigzag transformation to the rectangle with arbitrary dimensions. Meanwhile, the extended Zigzag transformation used in this paper can be started from four different corners, i.e., the top-left, top-right, bottom-left or bottom-right corners, and ended at the opposite corners. Compared with the traditional Zigzag transformation, the extended Zigzag transformation has better scrambling effect, as shown in Fig. 4.

2.3 Hash algorithm

Secure Hash Algorithm 2 (SHA-2) is a set of cryptographic hash functions, issued by the National Institute of Standards and Technology in 2001, and it is widely used to provide the security service of integrity. SHA-256, SHA-384 and SHA-512 are named after the original name by their digest length in bits. SHA-256 is one of the most commonly used hash functions in SHA-2 family [30]. They have the same functional structure with some variation in the internal operations, such as the message size, word size, block size, as shown in Table 1 [51]. SHA-256 algorithm is widely used in the digital signature, message authentication and other fields because of its high anti-collision ability and irreversibility. The proposed algorithm uses SHA-256 and external parameters to generate the keys.

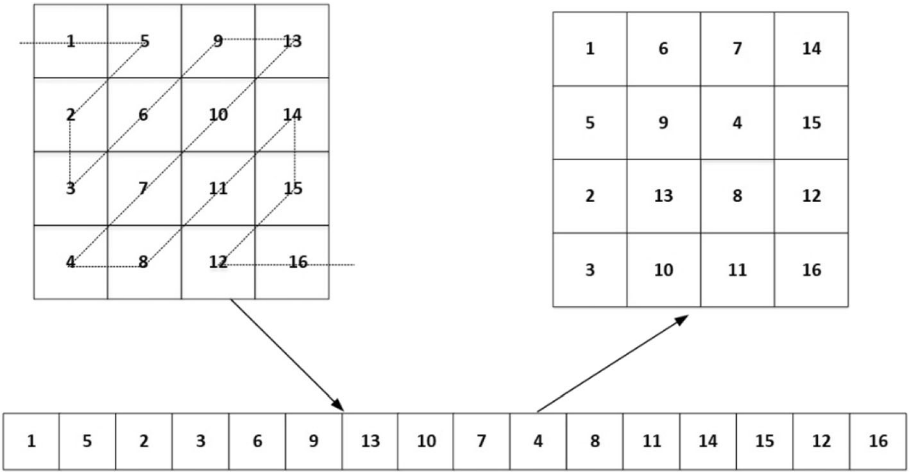


Fig. 4 Extended Zigzag transformation for 4×5 matrix

Table 1 Secure hash algorithms

Algorithms	Message size	Word size	Block size	Rounds	Output size	Security bits
SHA-1	$<2^{64}$	32	512	80	160	80
SHA-224	$<2^{64}$	32	512	64	224	112
SHA-256	$<2^{64}$	32	512	64	256	128
SHA-384	$<2^{128}$	64	1024	80	384	192
SHA-512	$<2^{128}$	64	1024	80	512	256

2.4 Bit plane

The pixel value of a gray image is between 0 and 255, which can be converted into an 8-bit binary number. The bit plane consists of 0 or 1 of the same bit of each 8-bit binary number. Let a gray image be $I_{m \times n}$. Therefore, each pixel value of $I_{m \times n}$ can be represented by 8-bit binary number. Figure 5 shows the 8 bit planes of Lena.

It can be seen from Fig. 5 that the information of Lena is divided into 8 different bit planes. The 8th bit plane contains a large quantity of the information, but the main features of Lena are hardly seen in the 1st bit plane. The information weights of these bit planes can be calculated by

$$I(i) = \frac{2^{i-1}}{255}, i = 1, 2, \cdots, 8 \tag{2}$$

where i represents the i^{th} bit plane. According to the Eq. (2), the information weights of the 8 bit planes are 50.196%, 25.098%, 12.549%, 6.275%, 3.137%, 1.568%, 0.784% and 0.393%, respectively. For the advantages of the bit-plane decomposition, lots of image encryption algorithms based on the bitplane have been proposed [9, 29, 46].

2.5 Superpixel

The bit depth of pixels refers to the number of bits expressed by each pixel. It is due to the number of each pixel colors for the color image, or due to the gray level of each pixel for the gray image. For example, if each pixel value of a gray image is 0–255, then its pixel depth is 8 bits.

In this sense, the bit depth of pixels is often referred to as the image depth. The more bits a pixel has, the more colors it can express. Although the pixel depth or image depth can be very deep in theory, due to the limitations of the device itself and the human eye's

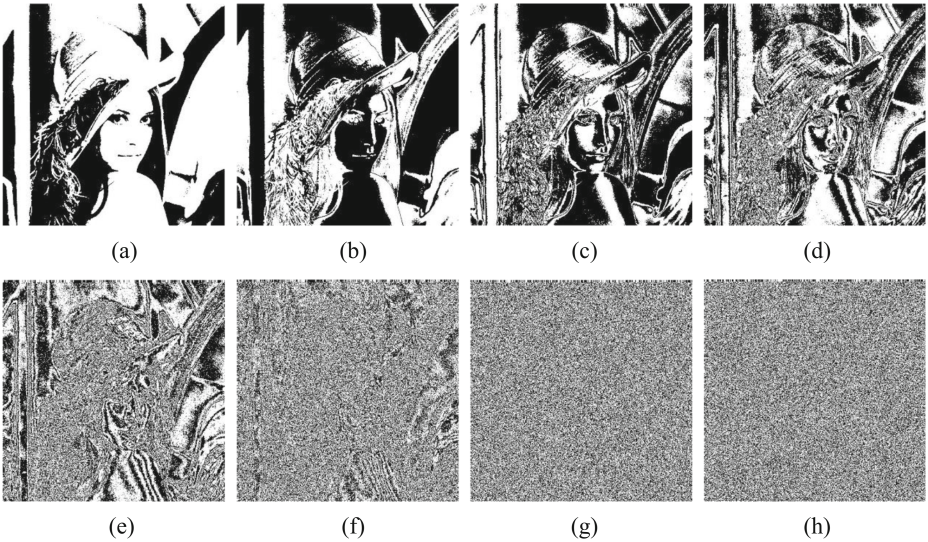


Fig. 5 8 bit planes of Lena (**a**) The 8th bit plane (**b**) The 7th bit plane (**c**) The 6th bit plane (**d**) The 5th bit plane (**e**) The 4th bit plane (**f**) The 3rd bit plane (**g**) The 2nd bit plane (**h**) The 1st bit plane

resolution, it is meaningless to pursue the particularly deep bit depth of pixels. Meanwhile, the deeper the bit depth of pixels is, the more data the image has. On the contrary, if the bit depth of pixels is too shallow, the image quality will become bad, which looks rough and unnatural.

In this paper, we define the superpixel value as an integer within $[0,\ 2^{8k}-1]$, so the bit depth of the superpixel is $8k$ bits, where k is a positive integer. Images composed of superpixels are called super images. Taking four 8-bit pixels 168, 30, 0 and 30 as an example, the process to generate the superpixel 2,820,538,398 is described in Fig. 6. Firstly, these four pixels are converted into 8-bit binary numbers, i.e., 1010 0000, 0001 1110, 0000 0000 and 0001 1110, respectively; secondly, connecting these four binary numbers head to tail, the result is 1010 0000 0001 1110 0000 0000 0001 1110; finally, it is converted into a decimal number 2820538398, i.e., the superpixel value.

3 Proposed MIE algorithm

3.1 Key generation

256-bit hash value h of the plain images is obtained by the SHA-256 hash function. h is divided into 8-bit blocks, i.e., k_1, k_2, $\cdots$, k_{32},

$$h = k_1, k_2, \cdots, k_{32} \tag{3}$$

where $k_i = k_{i,\ 1}, k_{i,\ 2}, \cdots, k_{i,\ 8}$ and $i = 1, 2, \cdots, 32$ denotes the block number.

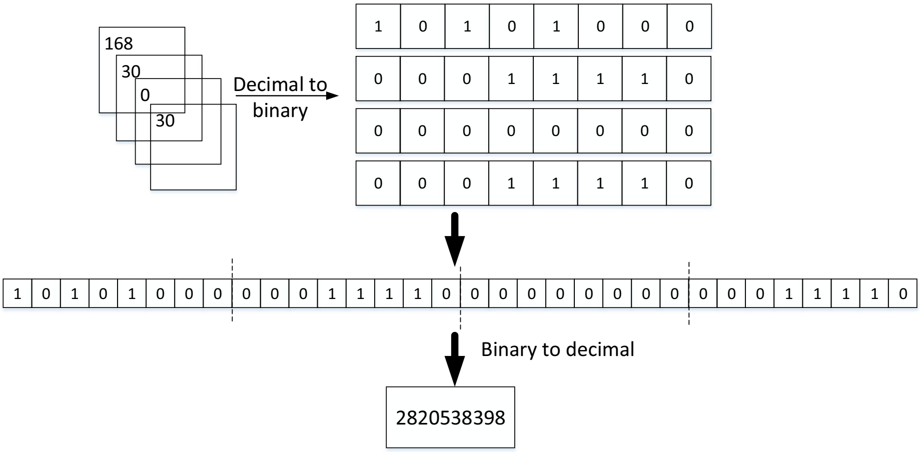


Fig. 6 Generation process of the superpixel

The initial values x_0, y_0 and the control parameter μ of the STS can be derived by

$$\begin{cases} x_0 = c_1 + \dfrac{k_1 \oplus k_2 \oplus \cdots \oplus k_{16}}{512} \\ y_0 = c_2 + \dfrac{k_{17} \oplus k_{18} \oplus \cdots \oplus k_{32}}{512} \\ \mu = c_3 + \dfrac{k_1 \oplus k_2 \oplus \cdots \oplus k_{32}}{256} \end{cases} \tag{4}$$

where $c_1, c_2 \in (0,\ 0.5)$ and $c_3 \in \{0,\ 1,\ 2,\ 3\}$ are the external key parameters.

3.2 Encryption process

For the proposed algorithm, the flowchart of the image encryption process is shown in Fig. 7. The detailed encryption steps are described as follows.

Step 1: Filling images

Let k plain images be $I_1, I_2, \cdots, I_k$, whose sizes are $m_1 \times n_1, m_2 \times n_2, \cdots, m_k \times n_k$, respectively. The plain images are enlarged from $m_i \times n_i$ to $m \times n$, and the enlarged area is filled with the pixel value "0", where $i = 1, 2, \cdots, k$, $m = \ max\ \{m_1,\ m_2,\ \cdots,\ m_k\}$ and $n = \ max\ \{n_1,\ n_2,\ \cdots,\ n_k\}$. The enlarged images are $I_1^{'}, I_2^{'}, \cdots, I_k^{'}$.

Step 2: Converting image

The pixels of each image are converted into 8-bit binary numbers, and each image can form 8 bit planes. Therefore, $8k$ bit planes formed from $I_1^{'}, I_2^{'}, \cdots, I_k^{'}$ are stored in a three-dimensional (3D) matrix P_b.

Step 3: Generating chaotic sequences

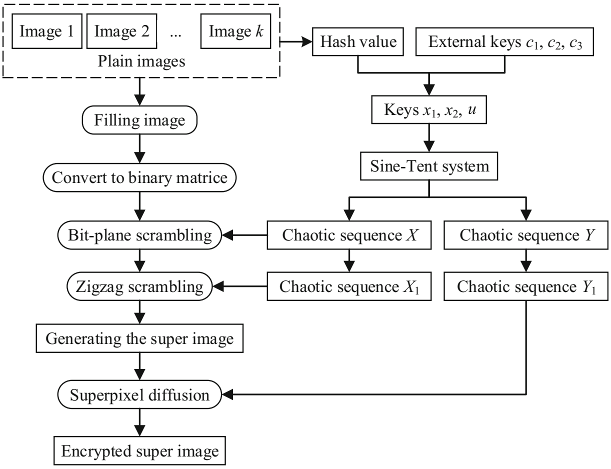


Fig. 7 The flowchart of the image encryption process

The chaotic sequence X can be generated by iterating the Eq. (1) $8k$ times with the initial value x_0 and control parameter μ generated in Subsection 3.1. The index sequence W is obtained by sorting X in descending order. Similarly, the chaotic sequence Y can be generated by iterating the Eq. (1) mn times with the initial value y_0 and control parameter μ generated in Subsection 3.1.

Step 4: Chaotic sequence integralization

The integer chaotic sequences X_1, Y_1 can be obtained by

$$X_1 = \mathrm{mod}\left(floor\left(X \times 10^{10}\right), 4\right) + 1 \tag{5}$$

$$Y_1 = \mathrm{mod}\left(floor\left(Y \times 10^{10}\right), 2^{8k}\right) \tag{6}$$

where ***floor***(•) rounds toward negative infinity, and *mod*(•) is the modulo operation.

Step 5: Scrambling operation among bit planes

Perform the scrambling operation among bit planes on P_b with W by

$$P_b^{'}(:,:,i) = P_b(:,:,W(i)), i = 1, 2, \cdots, 8k \tag{7}$$

Step 6: Scrambling operation in the bit plane

Perform the scrambling operation in each bit plane of $P_b^{'}$ with the extended Zigzag transformation by

$$P_b^{''}(:,:,i) = zigzag\left(P_b^{'}(:,:,i), x_1(i)\right), i = 1, 2, \cdots, 8k \tag{8}$$

where zigzag(•) is the extended Zigzag transformation, and $x_1(i) \in X_1$ denotes the start corner, i.e., the top-left, top-right, bottom-left or bottom-right corners.

Step 7: Constituting superpixels

The 2D matrix S with the size of $mn \times 8k$ is converted by $P_b^{''}$. The $8k$-bit binary numbers in each row of S are converted into decimal numbers to obtain a 1D column vector P_0 with the size of mn. Each element of P_0 is a superpixel, whose bit depth is $8k$ bits.

Step 8: Image diffusion

The 1D row vector P_1 with the size of mn is converted by P_0. The diffusion rule is designed as

$$\begin{cases} C_1(i) = P_1(i) \oplus Y_1(i) & i = 1 \\ C_1(i) = P_1(i) \oplus Y_1(i) \oplus C_1(i{-}1) & i > 1 \end{cases} \tag{9}$$

The final encrypted super image C can be obtained by converting C_1 into a 2D matrix with the size of $m \times n$, whose bit depth is $8k$ bits.

3.3 Decryption process

The decryption process is the inverse process of image encryption. For the proposed algorithm, the flowchart of the image decryption process is shown in Fig. 8. The detailed decryption steps are described as follows.

Step 1: Generating chaotic sequences

The chaotic sequences X and Y are generated with the keys x_1, x_2, μ and the STS.

Step 2: Chaotic sequence integralization

The integer chaotic sequences X_1, Y_1 can be obtained by Eqs. (5) and (6).

Step 3: Image diffusion

The 1D row vector C_1 with the size of mn is converted by the final encrypted super image C. The diffusion rule is designed as,

$$\begin{cases} P_1(i) = C_1(i) \oplus Y_1(i) & i = 1 \\ P_1(i) = C_1(i) \oplus Y_1(i) \oplus P_1(i-1) & i > 1 \end{cases} \tag{10}$$

Step 4: Converting image

A 3D binary matrix P_b'' with the size of $m \times n \times 8k$ is generated by converting P_1 into $8k$-bit binary numbers.

Step 5: Scrambling operation in the bit plane.

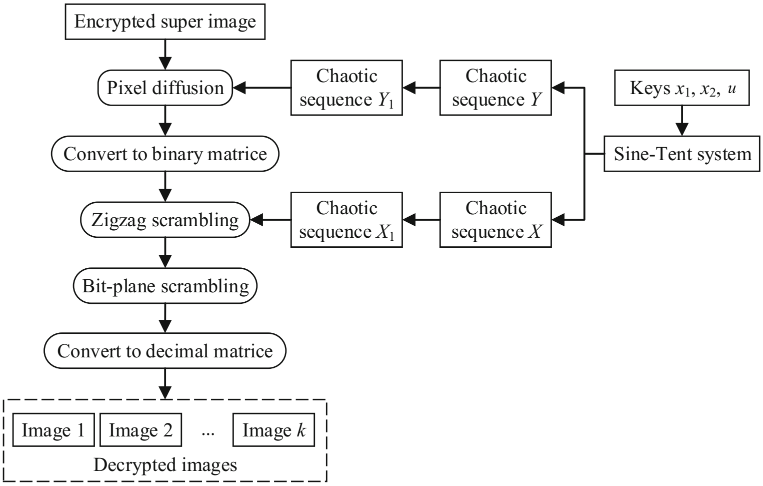


Fig. 8 The flowchart of the image decryption process

Perform the scrambling operation in each bit plane of P''_b by

$$P'_b(:,:,i) = antiZigzag\left(P''_b(:,:,i), X_1(j)\right), i = 1, 2, \cdots, 8k \tag{11}$$

where antiZigzag(•) is the extended anti-Zigzag transformation.

Step 6: Scrambling operation among bit planes

Perform the scrambling operation among bit planes on P'_b with W by

$$P_b(:,:,i) = P'_b(:,:,W(i)), i = 1, 2, \cdots, 8k \tag{12}$$

where W is the index sequence obtained by sorting X in descending order.

Step 7: Getting plain images

According to the Eq. (13), the enlarged images $I'_1, I'_2, \cdots, I'_k$ can be obtained by converting 8 bit planes of P_b into a 2D decimal matrix with the size of $m \times n$.

$$\begin{cases} I'_1 = bi2de(P_b(:,:,1:8)) \\ I'_2 = bi2de(P_b(:,:,9:16)) \\ \qquad\qquad \vdots \\ I'_k = bi2de(P_b(:,:,8(k-1)+1:8k)) \end{cases} \tag{13}$$

Step 8: Cutting images

The plain images $I_1, I_2, \cdots, I_k$ can be generated by cutting $I'_1, I'_2, \cdots, I'_k$ with the sizes of $m_1 \times n_1$, $m_2 \times n_2$, $\cdots$, $m_k \times n_k$, respectively.

4 Experiments

To evaluate the algorithm performance, our algorithm is simulated by both Matlab and Java programming languages on the PC with Intel(R) Core (TM)i5-7200U CPU @2.50GHz, 12G running memory and 64-bit Windows 10 system. The simulation softwares are Matlab R2016b and MyEclipse 2017 CI. The test images are from the USC-SIPI image database (http://sipi.usc.edu/database) and commonly used standard test images. Four plain images are shown in Fig. 9, and their sizes are 720×480, 512×512, 720×576 and 480×480, respectively. The hash value of these four images is h=6c9e97c11e808ba6321649 0bb257db7b2b545 caa63520661f1ba0ab5c2a25d75, and the external keys $c_1 = 0.1531$, $c_2 = 0.4586$, $c_3 = 3$. Therefore, we can calculate $x_1 = 0.2574$, $x_2 = 0.6520$, $\mu = 0.3948$ by the Eq. (4). The encrypted image is shown in Fig. 10. The decrypted images are completely the same with the plain images.

(a) (b)

(d) (d)

Fig. 9 Four plain images (**a**) Barbara (**b**) Lena (**d**) Goldhill (**d**) Columbia

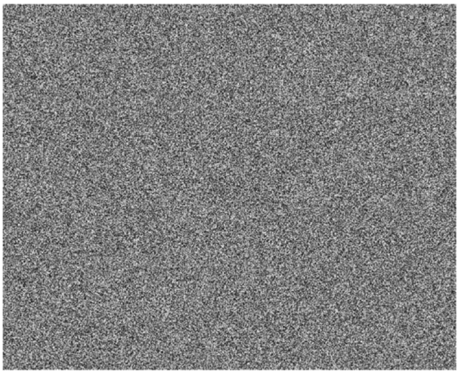


Fig. 10 Encrypted image

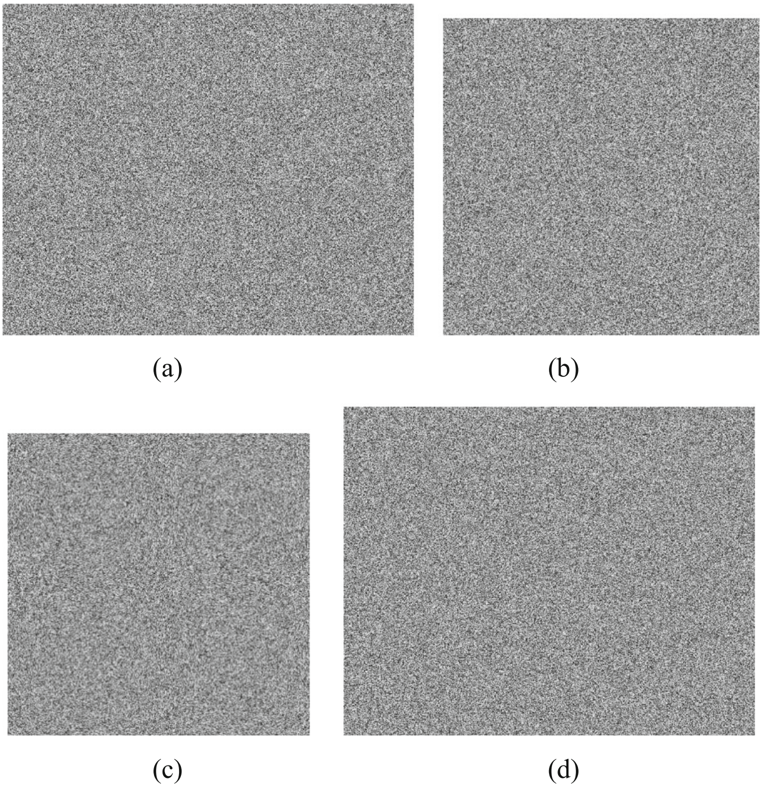
Table 2 Key space analysis

Algorithm	Key Space
Propose	2^{352}
Ref. [44]	2^{190}
Ref. [41]	2^{245}
Ref. [11]	2^{180}

5 Algorithm analyses

5.1 Key space analysis

The key space is the collection of all potential keys that can be used in the image encryption algorithm. An excellent encryption algorithm needs a large key space to resist the exhaustive attack. Generally speaking, if the key space is greater than 2^{100}, it is an excellent encryption algorithm. The keys of the proposed algorithm include 256-bit hash value h and external parameters c_1, c_2, c_3. If the computing precision of the computer is 10^{-14}, then the key space of our algorithm is about $2^{256} \times 10^{14 \times 2} \times 2^2 \approx 2^{352}$. Table 2 shows that the key space of the



(a)

(b)

(c)

(d)

Fig. 11 Decrypted images with wrong keys (**a**) Decrypted Babara (**b**) Decrypted Lena (**c**) Decrypted Columbia (**d**) Decrypted Goldhill

proposed algorithm is the largest of all the similar algorithms. Therefore, the proposed algorithm has a large enough key space to resist the brute-force attack.

5.2 Key sensitivity analysis

For a secure encryption system, the keys should be sensitive. A slight change to some keys will cause the failure of decrypting encrypted images. In our algorithm, the keys consist of the external parameter c_1, c_2, c_3 and the hash value h of plain images. To test the key sensitivity, we made c_1, c_2 add a very small number 10^{-10} when decrypted encrypted images. Figure 11 shows decrypted images by wrong keys, which are completely different with the plain images.

5.3 Histogram analysis

The histogram can count the pixel distribution of the image and represent its statistical information. Generally speaking, an excellent image encryption algorithm needs to completely destroy the statistical features of plaintext information. To resist the statistical attack, the histograms of encrypted images should be uniform without the statistical information of plain images. Synthesis of four plain images into a super plain image, the 3D histograms of the plain image and its corresponding encrypted image are shown in Fig. 12. It can be seen that the histogram of the encrypted image is uniform and no longer contains any statistical information of plain image.

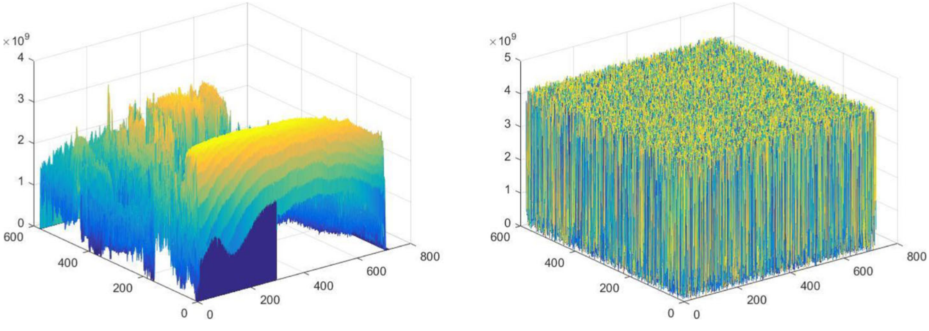
5.4 Correlation of adjacent pixels

The strong correlation of adjacent pixels is an important feature of digital images. Therefore, the correlation of adjacent pixels is one of the important criteria to evaluate the performance of an image encryption algorithm.

The image correlation is defined by

$$r_{x,y} = \frac{E\{[x-E(x)][y-E(y)]\}}{\sqrt{D(x)D(y)}} \tag{14}$$

where $E(x)$ and $D(x)$ are the mathematical expectation and variance of the data x, respectively. They are defined by

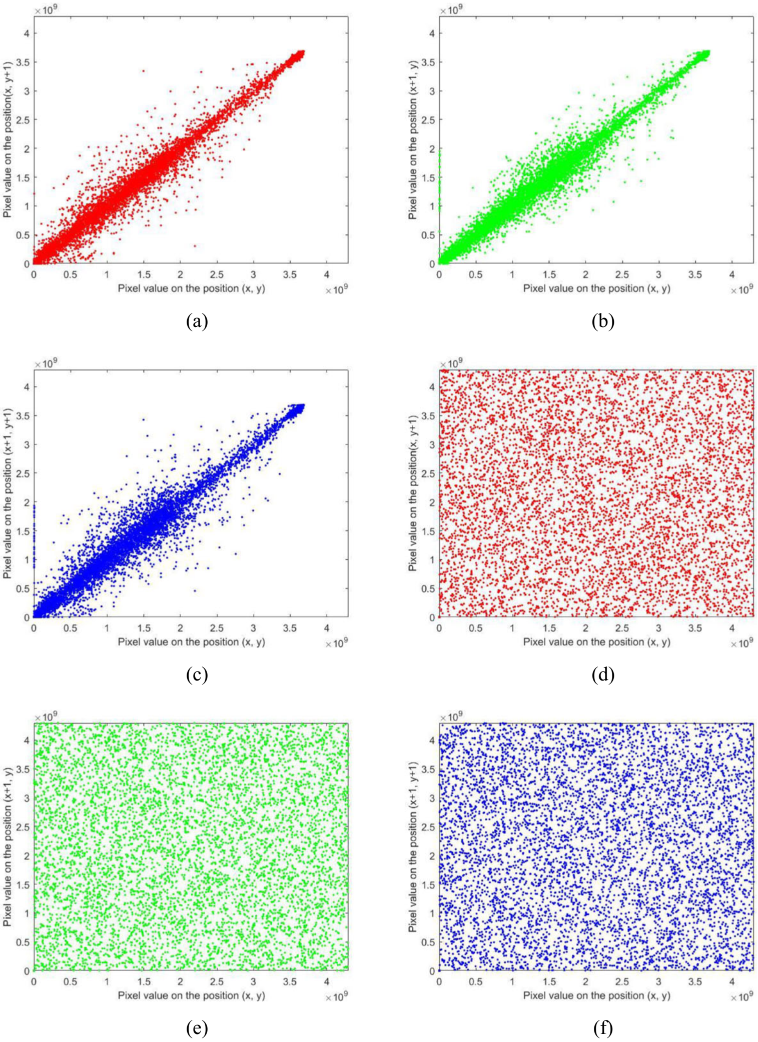


(a) (b)

Fig. 12 Histograms of Babara and its corresponding encrypted image (**a**) Histogram of Babara (**b**) Histogram of the encrypted image

Table 3 Correlation coefficients of original images

Directions	Horizontal	Vertical	Diagonal
Barbara	0.9185	0.9538	0.8971
Lena	0.9710	0.9847	0.9587
Columbia	0.9739	0.9716	0.9456
Goldhill	0.9766	0.9731	0.9571



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 13 Correlation analysis of original image Barbara and its corresponding encrypted image (**a**) Horizontal correlation of Babara (**b**) Vertical correlation of Babara (**c**) Diagonal correlation of Babara (**d**) Horizontal correlation of the encrypted image (**e**) Vertical correlation of the encrypted image (**f**) Diagonal correlation of the encrypted image

$$E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i \tag{15}$$

$$D(x)=\frac{1}{N}\sum_{i=1}^{N}[x_i-E(x)]^2 \tag{16}$$

To calculate the image correlation, many pairs of adjacent pixels are selected from the original images and the encrypted images. Generally speaking, the original image has the high pixel correlation, and its correlation coefficient is close to 1. However, the encrypted image has the low pixel correlation, and its correlation coefficient tends to 0. The correlation coefficients of the original images are listed in Table 3. We calculated their correlation coefficients in the horizontal, vertical and diagonal directions. It can be seen from Tab. 3 that the correlation coefficients of original images are close to 1. Taking Babara as an example, Fig. 13 is the correlation analysis of the plain image and encrypted image (i.e., Fig. 10). It can be seen from Fig. 13 that the plain image has a strong correlation, but the pixel correlation of the encrypted image is broken and presents a random distribution. Therefore, the proposed algorithm has an excellent encryption effect. Table 4 shows the comparison with other algorithms. It can be seen from Table 4 that all the algorithms can destroy the correlation between adjacent pixels well and protect the content of plain images well.

5.5 Differential attack analysis

The Number of Pixel Changes Ratio (NPCR) and the Unified Average Change Intensity (UACI) are two important indicators for evaluating the differential attack. NPCR reflects the number of changed pixels in the encrypted image after the original image is changed. The large NPCR value reflects the strong ability to resist the plaintext attack. The UACI measures the average difference intensity of pixel values between two encrypted images, which correspond to the original image and the changed original image. The large UACI value reflects the strong ability to resist the differential attack. NPCR and UACI are defined by [40].

$$NPCR=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}f(i,j)}{m\times n}\times 100\% \tag{17}$$

Table 4 Correlation coefficient of the encrypted image

Algorithms	Horizontal	Vertical	Diagonal
Ref. [25]	−0.0036	0.0026	0.0012
Ref. [31]	−0.0036	0.0016	0.0058
Ref. [12]	−0.0538	0.0389	0.0307
Ref. [11]	−0.0016	0.0057	−0.0189
Ref. [41]	0.0034	0.0015	0.0008
Ref. [10]	0.0009	0.0016	0.0007
Ref. [45]	−0.0003	0.0011	0.0013
Proposed	0.0034	−0.0010	0.0009

Table 5 NPCR and UACI values

Algorithm	NPCR	UACI
Ref. [25]	99.5907%	33.4811%
Ref. [31]	99.9100%	33.4800%
Ref. [11]	99.6250%	33.4510%
Ref. [41]	99.1841%	33.5284%
Ref. [10]	99.6534%	33.6772%
Ref. [45]	99.6060%	33.5126%
Proposed	99.6062%	33.4522%

$$UACI = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|I^{'}(i,j)-I^{''}(i,j)\right|}{m\times n}\times 100\% \tag{18}$$

where m and n represent the size of the original image, and $I'(i, j)$, $I''(i, j)$ respectively represent the encrypted images corresponding to the original image and the changed original image. $f(i, j)$ is defined by

$$f(i,j)=\begin{cases}0 & I^{'}(i,j)=I^{''}(i,j)\\ 1 & I^{'}(i,j)\neq I^{''}(i,j)\end{cases} \tag{19}$$

For the 8-bit gray image, the ideal values of NPCR and UACI are 99.6094% and 33.4635% [40], respectively. To test the ability of the proposed algorithm to resist the differential attack, we changed the pixel value 112 at the position (30, 30) of the plain image Babara for 224. After that, we encrypt the changed Babara and other three plain images with the same encryption algorithm and keys. Finally, the values of NPCR and UACI are listed in Table 5. The data in Tab. 5 shows that the NPCR and UACI values are close to their ideal values for the proposed algorithm and similar algorithms. Therefore, the proposed algorithm has strong resistance to the differential attack.

5.6 Encryption efficiency and computational complexity

We carried out experiments on the time-consuming of the proposed algorithm. Table 6 shows the time consumption of several MIE algorithms. It can be seen from Table 6 that four plain images can be encrypted with Matlab programming language in 1.41 seconds, and Java programming language in 1.07 seconds for the proposed algorithm. Meanwhile, the proposed algorithm is faster than other similar algorithms. Therefore, the proposed algorithm is efficient, which can encrypt multiple images in a short time.

Table 6 Time consumption of encryption algorithms

Algorithm	Time (second)	system characteristics
Ref. [31]	1.78	CPU @ 3.60 GHz, 32 GB RAM
Ref. [41]	1.89	CPU @2.00 GHz, 4GB RAM
Ref. [10]	1.86	CPU @2.30 GHz, 8 GB RAM
Ref. [45]	1.71	CPU @1.20 GHz, 8 GB RAM
Proposed with Matlab	1.41	CPU @2.50 GHz, 4GB RAM
Proposed with Java	1.07	CPU @2.50 GHz, 4GB RAM

Table 7 MSE and PSNR analysis

Test images	Barbara	Lena	Goldhill	Columbia
MSE	8547	12,627	14,881	10,519
PSNR	8.8122	7.1179	6.4045	7.9109

The proposed algorithm adopts the scrambling-diffusion structure. The cost of time consumption mainly includes the generation of chaotic sequences, the conversion between decimal numbers and binary numbers, the scrambling operation among bit planes, the extended Zigzag transformation, and the XOR operation. In our algorithm, the size of plain images is $m \times n$. The computational complexity of generating two chaotic sequences X, Y is $O(8k) + O(mn)$. The computational complexity of the conversion between decimal numbers and binary numbers is $O(kmn) + O(mn)$. The computational complexity of the scrambling operation among bit planes is $O(8k)$. The computational complexity of the extended Zigzag transformation is $O(8kmn)$. The computational complexity of the XOR operation is $O(1 + 2(mn - 1))$. In total, the computational complexity of the proposed algorithm is $O(16k + 4mn + 9kmn)$. Therefore, the computation complexity of our algorithm depends on the size and number of the plain images at once.

5.7 Encryption quality analysis

In statistics, the Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) can characterize the encryption quality of image encryption algorithms. MSE and PSNR are defined by [27].

$$PSNR = 10 \times \lg\left(\frac{255^2}{MSE}\right) \tag{20}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|I_O(i,j) - I_E(i,j)\right| \tag{21}$$

where I_O is the original image, and I_E is the encrypted image. The lower value of PSNR indicates a significant difference between the original image and the encrypted image. In general, the encryption image with a large MSE value and a PSNR value less than 10 dB means an efficient pseudo-random ciphertext, and it is different in structure from the corresponding plain image. Table 7 lists the MSE and PSNR values for different gray images. From these results, we can conclude that our algorithm has high encryption quality.

5.8 Occlusion attack analysis

When the image information is transmitted over the network, the grays changes and some details of the image information are easily weakened. Therefore, the loss of image information is a common phenomenon in the data transmission. For the proposed MIE algorithm, the encrypted image contains the main information of the plain image. When a part of the encrypted image is assumed to be cropped, the pixel values of the corresponding part are replaced with zeros. The occlusion attack and corresponding decrypted results are shown in Figs. 14, 15, 16, 17 and 18.

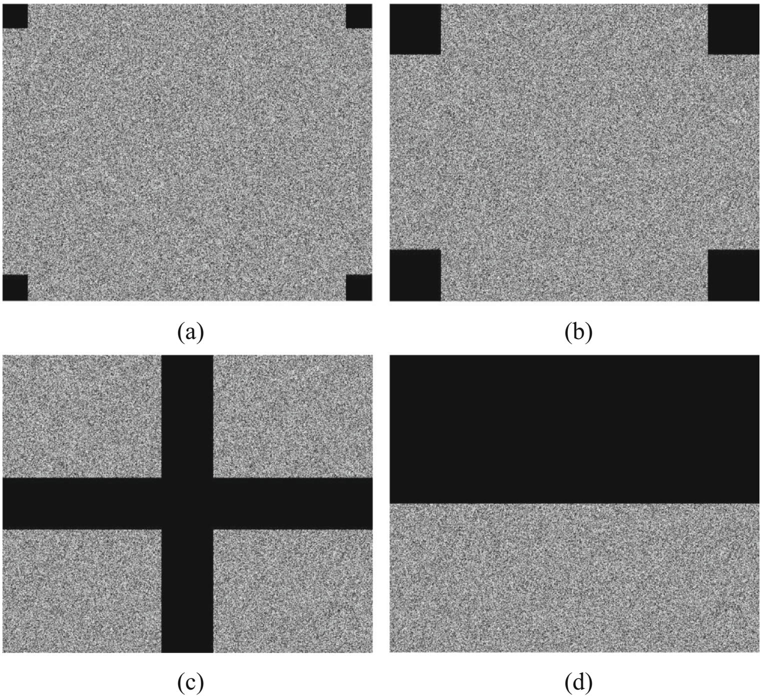


Fig. 14 Occlusion attack (**a**) 10% occlusion (**b**) 20% occlusion (**c**) 40% occlusion (**d**) 50% occlusion

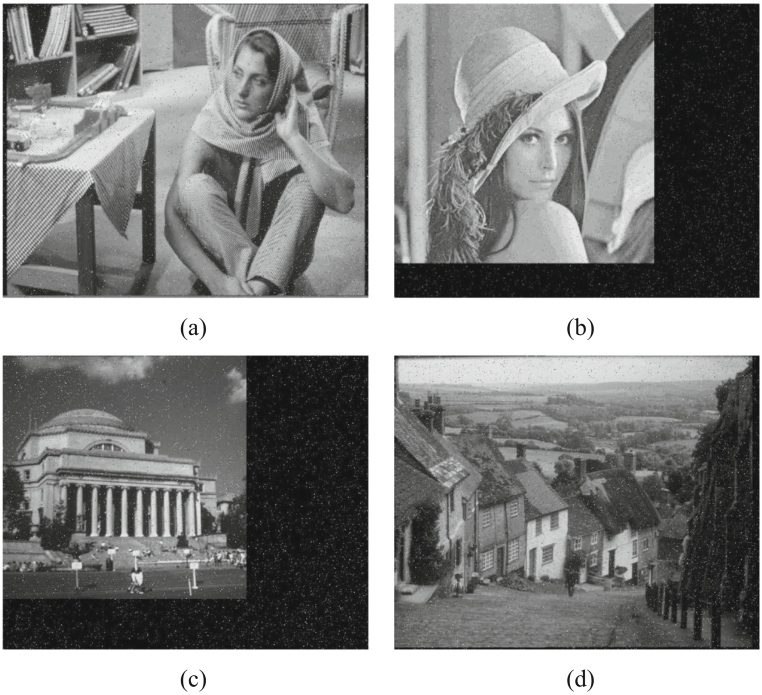
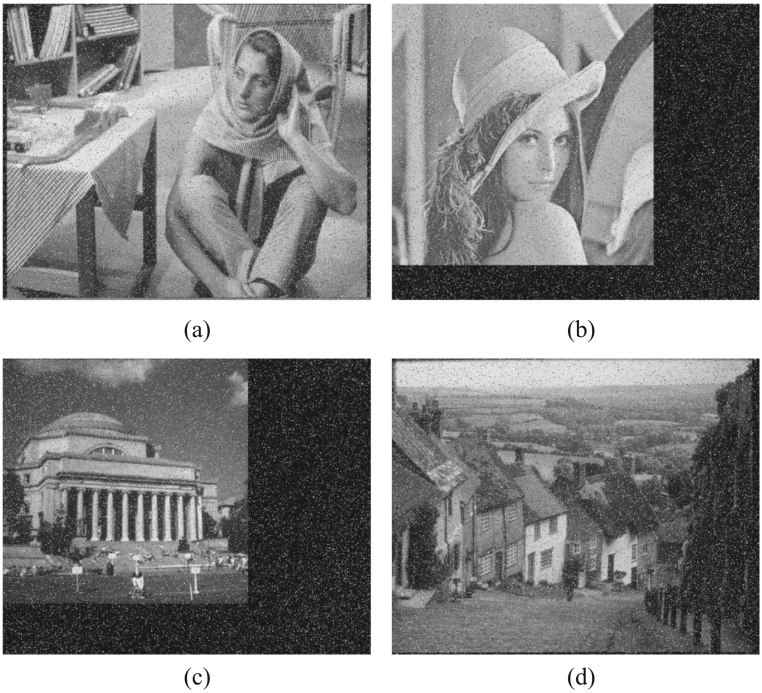


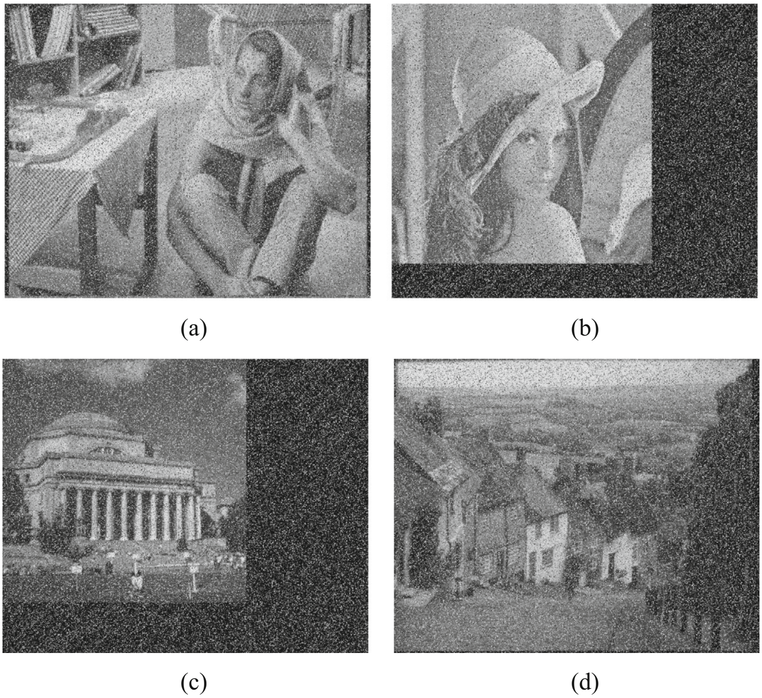
Fig. 15 Decryption images of 10% occlusion (**a**) Decrypted Babara (**b**) Decrypted Lena (**c**) Decrypted Goldhill (**d**) Decrypted Columbia



(a) (b)

(c) (d)

Fig. 16 Decrypted images of 20% occlusion (**a**) Decrypted Babara (**b**) Decrypted Lena (**c**) Decrypted Goldhill (**d**) Decrypted Columbia



(a) (b)

(c) (d)

Fig. 17 Decrypted images of 40% occlusion (**a**) Decrypted Babara (**b**) Decrypted Lena (**c**) Decrypted Columbia (**d**) Decrypted Goldhill

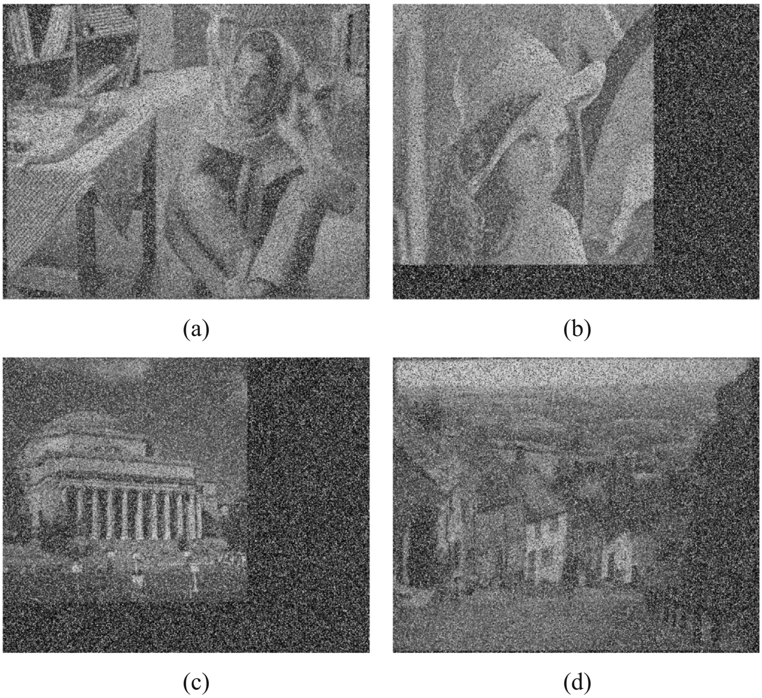


Fig. 18 Decrypted images of 50% occlusion (**a**) Decrypted Babara (**b**) Decrypted Lena (**c**) Decrypted Goldhill (**d**) Decrypted Columbia

6 Conclusions

This paper defines the superpixel and super image, and proposes an MIE algorithms based on the bit plane and superpixel. Different from other MIE algorithm, our algorithm encrypts multiple images with the unit of superpixels. The proposed algorithm adopts the classical scrambling-diffusion structure. At the scrambling stage, our algorithm includes the scrambling operation among bit planes and the extended Zigzag transformation in the bit plane. At the diffusion stage, our algorithm designs the diffusion rule with the XOR operation. Through key space analysis, histogram analysis, correlation analysis, etc., the proposed algorithm can effectively resist the brute-force attack, statistical attack and differential attack. Our algorithm has strong security, efficiency and robustness.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Akkasaligar PT, Biradar S (2020) Selective medical image encryption using DNA cryptography. Inform Sec J: A Global Perspec 29(2):91–101
2. Abdulla Alan (2019) Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography. Doctoral thesis, University of Buckingham, 12(149): 1–230
3. Chandra BR, Suvamoy C (2021) A novel and efficient amino acid codon based medical image encryption scheme colligating multiple chaotic maps. Multimed Tools Appl 80(7):10723–10760
4. Chen Q, Shen X, Cheng Y, Liu J, Wu W, Liu Y (2020) Optical multiple images encryption system based on ellipsometry reconstruction. J Mod Opt 67(17):1388–1397
5. Chen H, Liu Z, Camel T et al (2021) A novel chaos based optical cryptosystem for multiple images using DNA-blend and gyrator transform. Opt Lasers Eng 138(5):1–9
6. Cheng G, Wang C, Chen H (2019) A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture. Int J Bifurca Chaos 29(9):1–17
7. Chhabra P, Kumar NG, Kumar M (2020) Content-based image retrieval system using ORB and SIFT features. Neural Comput Applic 32(7):2725–2733
8. Dai J, Ma Y, Zhou N (2021) Quantum multi-image compression-encryption scheme based on quantum discrete cosine transform and 4D hyper-chaotic Henon map. Quantum Inf Process 20(7):1–24
9. Dasari S, Abhimanyu PK, Kumar AB et al (2019) A secure chaotic image encryption based on bit-plane operation. Soft Compu Data Analy 758:717–726
10. Enayatifar R, Guimaraes FG, Siarry P (2019) Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt Lasers Eng 115(4):131–140
11. Haq TU, Shah T (2020) Algebra-chaos amalgam and DNA transform based multiple digital image encryption. J Inform Sec App 54(4):102592–1025609
12. Hasheminejad A, Rostami MJ (2019) A novel bit level multiphase algorithm for image encryption based on PWLCM chaotic map. Optik 184(10):205–213
13. Jaehun S, Lee Y (2021) Optical image encryption using different twiddle factors in the butterfly algorithm of fast Fourier transform. Opt Commun 485(4):1–9
14. Jain M, Kumar A (2017) RGB channel based decision tree grey-alpha medical image steganography with RSA cryptosystem. Int J Mach Learn Cybern 8(5):1695–1705
15. Jain M, Kumar A, Choudhary RC (2017) Improved diagonal queue medical image steganography using Chaos theory, LFSR, and Rabin cryptosystem. Brain Inform 4(2):95–106
16. Jain K, Aji A, Krishnan P (2021) Medical image encryption scheme using multiple chaotic maps. Pattern Recogn Lett 152(11):356–364
17. Kashif S, Mao A, Imran Q et al (2022) DS-CNN: a pre-trained Xception model based on depth-wise separable convolutional neural network for finger vein recognition. Expert Syst Appl 191(4):1–18
18. Kaur RP, Jindal MK, Kumar M (2021) Text and graphics segmentation of newspapers printed in Gurmukhi script: a hybrid approach. Vis Comput 37(7):1637–1659
19. Kumar A, Ghose MK (2011) Extended substitution-diffusion based image cipher using chaotic standard map. Commun Nonlinear Sci Numer Simul 16(1):372–382
20. Kumar M, Chhabra P, Kumar NG (2018) An efficient content based image retrieval system using BayesNet and K-NN. Multimed Tools Appl 77(16):21557–21570
21. Laiphrakpam DS, Waikhom LS, Brahma D, Baruah P, Biswas S (2021) Image compression-encryption scheme using SPIHT and chaotic systems. J Inform Sec App 63(12):1–12
22. Li T, Zhang D (2021) Hyperchaotic image encryption based on multiple bit permutation and diffusion. Entropy 23(5):1–22
23. Luan G, Chen Z, Huang C (2020) Silhouette-free multiple image encryption using coherent superposition and Fresnel transform. Optik 224(9):1–8
24. Matthews R (1989) On the derivation of a “chaotic” encryption algorithm. Cryptologia 13(1):29–42
25. Mohammad Zarebnia H (2019) Pakmanesh, Reza Parvaz. A fast multiple-image encryption algorithm based on hybrid chaotic systems for gray scale images. Optik 179(9):761–773
26. Munir N, Khan M, Jamal SS, Hazzazi MM, Hussain I (2021) Cryptanalysis of hybrid secure image encryption based on Julia set fractals and three-dimensional Lorenz chaotic map. Math Comput Simul 190(12):826–836
27. Murillo-Escobar MA, Meranza M, Lopez-Gutierrez RM (2019) Suggested integral analysis for chaos-based image cryptosystems. Entropy 21(8):815–839
28. Qiao Z, El Assad S, Taralova I (2020) Design of secure cryptosystem based on chaotic components and AES S-box. AEU-Intern J Electro Commu 121(7):1–13
29. Ram R, Arvind Y (2021) Security analysis of bit-plane level image encryption schemes. Def Sci J 71(2): 209–221

30. Ramzi Guesmi MA, Farah B (2021) A new efficient medical image cipher based on hybrid chaotic map and DNA code. Multimed Tools Appl 80(2):1925–1944
31. Sahasrabuddhe A, Laiphrakpam DS (2021) Multiple images encryption based on 3D scrambling and hyper-chaotic system. Inf Sci 550(5):252–267
32. Shazia S, Vandana G (2021) Multi-layer color image encryption using random matrix affine cipher, RP2DFrHT and 2D Arnold map. Multimed Tools Appl 80(18):27829–27853
33. Shevchenko II (2014) Lyapunov exponents in resonance multiples. Phys Lett A 378(1):34–42
34. Tang Z, Song J, Zhang X, Sun R (2016) Multiple-image encryption with bit-plane decomposition and chaotic maps. Opt Lasers Eng 80(5):1–11
35. Wan W, Wang J, Li J, Sun J, Zhang H, Liu J (2020) Hybrid JND model-guided watermarking method for screen content images. Multimed Tools Appl 79(7–8):4907–4930
36. Wan W, Wang J, Li J, Meng L, Sun J, Zhang H, Liu J (2020) Pattern complexity-based JND estimation for quantization watermarking. Pattern Recogn Lett 130:157–164
37. Wang X, Chen X (2021) An image encryption algorithm based on dynamic row scrambling and zigzag transformation. Chaos, Solitons Fractals 147(6):1–22
38. Wang J, Wan W, Li X et al (2020) Color image watermarking based on orientation diversity and color complexity. Expert Syst Appl 140(2):1–16
39. Wang X, Guan N, Yang J (2021) Image encryption algorithm with random scrambling based on one-dimensional logistic self-embedding chaotic map. Chaos, Solitons Fractals 150(9):1–16
40. Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. Cyber J 4(4): 31–38
41. 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(10):761–773
42. Zhang X, Wang JZ (2013) Spread spectrum image data hiding in the encrypted discrete cosine transform coefficients. J Electro Imag 22(4):1171–1180
43. Zhang X, Wang X (2017) Multiple-image encryption algorithm based on mixed image element and permutation. Opt Lasers Eng 92(5):6–16
44. Zhang X, Wang X (2017) Multiple-image encryption algorithm based on mixed image element and chaos. Compu Elect Engin 62(8):401–413
45. Zhang X, Yangming H (2021) Multiple-image encryption algorithm based on the 3D scrambling model and dynamic DNA coding. Opt Laser Technol 141(4):1–16
46. Zhang L, Zhang X (2020) Multiple-image encryption algorithm based on bit planes and chaos. Multimed Tools Appl 79(29–30):20753–20771
47. Zhang X, Wang JZ, Wang X (2014) Correlation-and-bit-aware additive spread spectrum data hiding for Laplacian distributed host image signals. Sign Proc-Image Commu 29(10):1171–1180
48. Zhang Y, Wang Z, Zhan Y, Meng L, Sun J, Wan W (2021) JND-aware robust image watermarking with tri-directional inter-block correlation. Int J Intell Syst 36(12):7053–7079
49. Zheng J, Liu L (2020) Novel image encryption by combining dynamic DNA sequence encryption and the improved 2D logistic sine map. IET Image Process 14(11):2310–2320
50. Zhou K, Fan J, Fan H, Li M (2020) Secure image encryption scheme using double random-phase encoding and compressed sensing. Opt Laser Technol 121(1):1–12
51. Zhu C, Gan Z, Lu Y, Chai X (2020) An image encryption algorithm based on 3-D DNA level permutation and substitution scheme. Multimed Tools Appl 79(12):7227–7258

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.