Nonlinear Dyn (2021) 106:3583–3599
https://doi.org/10.1007/s11071-021-06941-y

ORIGINAL PAPER

Image Encryption Using Value-Differencing Transformation and Modified ZigZag Transformation

Zhongyun Hua · Jiaxin Li · Yuanman Li · Yongyong Chen

Received: 1 April 2021 / Accepted: 24 September 2021 / Published online: 15 October 2021

Abstract Image encryption is an efficient technique to protect the contents of an image. However, many existing image encryption algorithms have low efficiency and are weak to resist many commonly used security attacks such as the chosen-plaintext attack. To address these issues, in this paper, we propose a new image encryption scheme using value-differencing transformation (VDT) and modified ZigZag transformation. First, we propose a new transform called VDT. It can split a plain image into four subbands based on value-differencing and this can greatly help to diffuse the image pixels. Second, we develop a modified ZigZag transformation that can separate the image pixels with a much higher efficiency. Besides, to improve the ability to defense the chosen-plaintext attack, we extract part information of plain image as parameters to encrypt the image. Notice that these information don't need when decrypting the image so that the proposed image encryption scheme is a symmetric key encryption algorithm. Simulation results and security analysis show that the proposed image encryption scheme can encrypt different kinds of plain images into unrecognized cipher images with a high security level, and it can outperform several existing image encryption schemes.

Keywords Chaotic system · Image encryption · Secure communication

1 Introduction

With the rapid development of information technology, information security becomes more and more important. Thus, researchers have developed a large number of information security technologies such as the well-known Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The image is a widely used data format since it has a straightforward visual effect and can show much potential information [35]. Recently, to securely communicate the digital images, many image security techniques have been proposed, such as the data hiding [6,29,44], image encryption [10,37,46] and watermarking [30,31,40].

Among all the image security techniques, image encryption is one of the most straightforward and effective methods and it can encrypt a plain image into an completely unrecognized cipher image with high security level [1,12]. Only with the correct secret key, one can recover the original image completely. Generally, all the existing image encryption methods can be clas-

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11071-021-06941-y.

Z. Hua · J. Li · Y. Chen (✉)
School of Computer Science and Technology, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China
e-mail: YongyongChen.cn@gmail.com

Z. Hua
e-mail: huazyum@gmail.com

Y. Li
College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China

sified into four categories, including optical encryption [18], spatial domain encryption, frequency domain encryption and compressive sensing [14,34,43]. For example, the spatial domain encryption is to directly encrypt the image pixels using DNA coding [3], cellular automata coding [32], chaotic theory [13,17,19,21,41] and many other techniques. The frequency domain encryption first transforms a plain image from spatial domain to the frequency domain using the fast Fourier transform [8], discrete cosine transform [24], discrete Wavelet transform [34], etc., and then encrypts the image in the frequency domain. Among all the techniques used in image encryption, the chaotic theory is a widely used and effective one, since it has many similar properties to the principles of image encryption. However, when a chaotic system is simulated in digital platform, chaos degradation will occur because of the limitation of precision. This may result in serious security problems for the chaos-based image encryption algorithms [15]. Thus, when used in image encryption, chaotic systems are usually combined with other techniques to improve the security level of cipher images.

Since a permutation–diffusion structure for image encryption was firstly proposed by Feistel [11], it receives increasing attention of researchers [7,20,27,45,48]. In the permutation–diffusion structure, the permutation operation changes the pixel positions and reduces the correlations between the adjacent pixels, while the diffusion operation randomly changes the pixel values using pixel dependence manner. However, many image encryption algorithms using traditional permutation–diffusion structure cannot resist many security attacks such as the chosen-plaintext attack [9,22,23]. This is because by choosing some plaintexts to encrypt and analyze their corresponding ciphertexts, the cryptanalysts can build the equivalent key for the permutation and diffusion operations [25,49]. Using the equivalent key, the cryptanalysts can reconstruct the plain image without the secret key [36]. One efficient way to solve this problem is to use the hash result of the plain image as a part of secret key. Using this strategy, the encrypted result is related to the plain image, which may make the chosen-plaintext attack ineffective [5]. However, this strategy causes serious problem for the key distribution, since the decrypted key must contain the hash result of the plain image.

To address the weaknesses of existing image encryption algorithms, in this paper, we introduce a new image encryption algorithm considering the contents of plain image. First, a new value-differencing transformation (VDT) is presented to decompose the plain image to enhance the image diffusion efficiency. Then, a modified ZigZag transformation is developed to quickly separate the image pixels. Our proposed algorithm can efficiently resist the chosen-plaintext attack [26]. This is because some features of the plain image are used as parameters to encrypt the image. Then, the permutation and diffusion results are sensitive with the plain image. Besides, the diffusion operation is performed from both forward and backward directions, which also enhances the security level. Since the features of the plain image are unnecessary when decrypting the image, the proposed image encryption scheme is a symmetric key encryption algorithm and the decryption key is the same with the encryption key. The main contributions and novelty of this paper are summarized as follows.

(1) We propose a new image decomposition technique, the VDT, which can split an image into four different subbands: low-low (LL), horizontal difference (HD), vertical difference (VD) and diagonal difference (DD). The VDT can greatly help to improve the encryption efficiency since it can selectively process the image contents in the diffusion process.
(2) We introduce a modified ZigZag transformation and propose a new permutation method using it. The new permutation method can improve the permutation efficiency using less pseudo random numbers.
(3) We use some features of the plain image as parameters to encrypt the image. Then, the encryption algorithm can well defense the chosen-plaintext attack. Because these features are unnecessary when decrypting the image, the proposed image encryption scheme is a symmetric key encryption algorithm and thus can overcome the key distribution problem of existing similar algorithms.
(4) Simulation results and security analysis show that the developed image encryption algorithm has a high security level to resist many security attacks and it has better performance than some state-of-the-art image encryption algorithms.

The rest of this paper is organized as follows. Section 2 introduces the proposed VDT and modified ZigZag transformation. Section 3 presents a new image encryption algorithm using the VDT and modified ZigZag transformation. Section 4 analyzes the security performance of the encryption algorithm, and Sect. 5 concludes this paper.

2 VDT and Modified ZigZag Transformation

This section presents the proposed VDT and modified ZigZag transformation. The VDT splits an image into different subbands to change the pixel values selectively, while the modified ZigZag transformation changes the positions of the pixels in a high efficiency.

2.1 VDT

Here, we develop the VDT that can diffuse the image selectively. Specially, the VDT splits a plain image into one original subband and three difference subbands by calculating the differences between four neighbor pixels of an image block. The detail procedures of the VDT are shown as follows:

Step 1: Divide a digital image of size $M \times N$ into 2×2 image blocks.

Step 2: Calculate the differences between the pixels on (1,1) with (1,2),(2,1),(2,2) in every blocks, and put the results to the subbands HD, VD and DD, respectively. Then put the pixels on (1,1) of every block to the subband LL.

Step 3: Put the values in the subbands LL, HD, VD and DD to upper right position, low left position and low right position, respectively.

Step 4: Finally, perform a modulo 256 operation to all the generated subbands.

A numeral example with image size 4×4 is demonstrated to better show the VDT, and Fig. 1 shows the detailed process. As can be seen, the image is first divided into four blocks of size 2×2. Then, calculate the differences between the first pixels with the other three pixels in each block. Taking the first block as an example, the differences between the pixels on (1,1), (1,2), (2,1) and (2,2) are $75 - 115 = -40$, $200 - 115 = 85$, and $205 - 115 = 90$, respectively. After the calculation, put the first pixel of each block in the upper left position, and put the values in the HD, VD and DD to the upper left position, upper right position, low left position and low right position, respectively. Finally, perform a modulo 256 operation to all the values. The negative values -40, -3, -3 and -9 become 216, 253, 253 and 247, respectively.

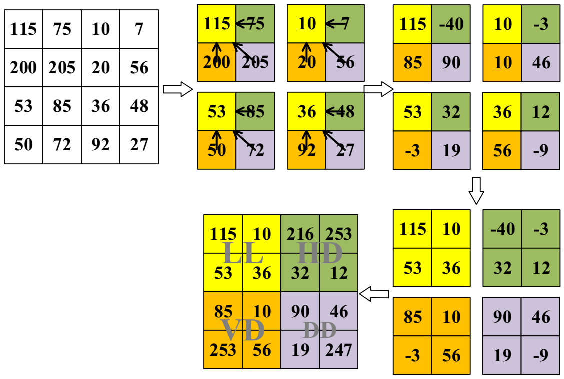

Fig. 1 A numeral example of the VDT

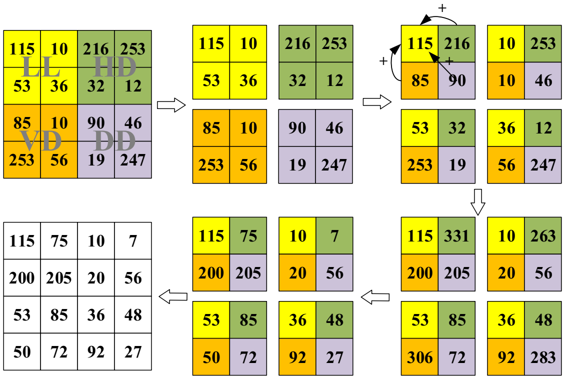

Fig. 2 The IVDT of the VDT in Fig. 1

It is obvious that the VDT is a reversible operation and the inverse VDT (IVDT) is to perform the reversible of each operation in the VDT. Figure 2 demonstrates the IVDT of the example in Fig. 1. It can be seen that all the original image pixels can be totally recovered.

It is obvious that if we change the positions of the pixels in different subbands, the IVDT cannot correctly recover the original image pixels. Using this strategy, we can diffuse the image pixels and Fig. 3 shows an example about it. Firstly, change the positions of pixels in HD, VD and DD randomly. Then, add the pixels in HD, VD and DD with the pixels in LL of the same position. For example, after permutation, the pixels on (1,1) of HD, VD and DD are 46, 32 and 253, respectively. Add them with the pixel on (1,1) of LL; then put the calculation results 199, 185 and 406 on the positions of (1,2), (2,1) and (2,2). Finally, all the pixels in the result are performed the operation modulo 256. As can be seen from Fig. 3, most pixel values are changed.

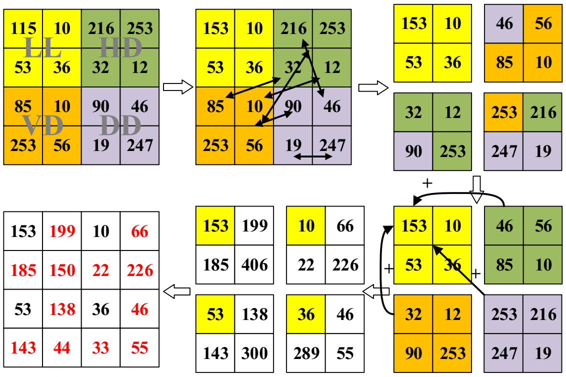

Fig. 3 An example of IVDT with the positions changed

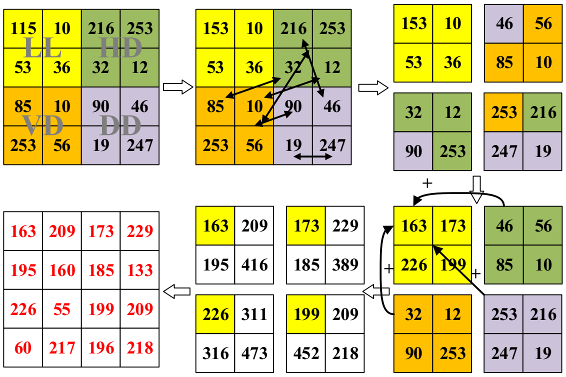

Fig. 4 An example of IVDT with the positions changed and LL subband diffused

However, the above strategy has a problem that the value of the pixels in LL subband cannot be changed. To solve this issue, we separately diffuse these pixels. Compared with the traditional diffusion method, we only need to diffuse one-fourth pixels of the image and this can improve the efficiency of the encryption algorithm. Figure 4 shows the IVDT result when randomly changing the positions of LL subband in the VDT result. It can be seen that all the image pixels are changed, comparing with the original image.

2.2 Modified ZigZag Transformation

The ZigZag transformation is a procedure to scan the elements of a matrix following the ‘Z’ shape and store the scanned elements into an one-dimensional (1D) array sequentially. Then, the 1D array can be rearranged as a two-dimensional (2D) matrix according to specific requirements [4]. When used in image encryption, the

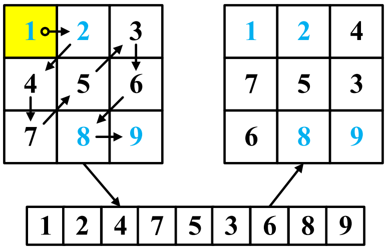

Fig. 5 The standard ZigZag mode, and the yellow block is the starting point

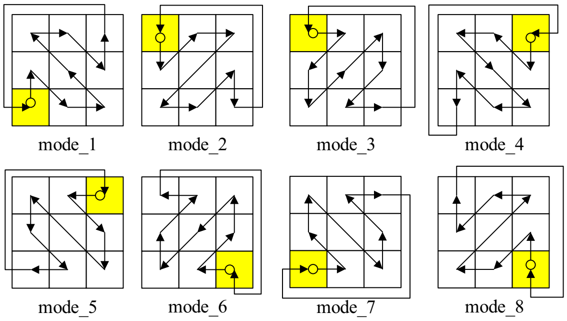

Fig. 6 Eight ZigZag modes

ZigZag transformation can permute the image pixels and Fig. 5 shows the process of the ZigZag transformation. Generally, there are eight ZigZag transformation modes, which are shown in Fig. 6. As can be seen, one can start from each of the four corners and either the horizontal or vertical direction. Thus, the scanning order array can form a ring as is shown in Fig. 6. Besides, a shifting can be used to the 1D array to further improve the randomness.

However, the ZigZag transformation has some problems. First, as shown in Fig 6, it only has eight modes and thus the key space is not large enough to resist brute force attack. Moreover, it cannot efficiently decorrelate the high correlations between the neighbor elements.

To address these issues and obtain more random result, we introduce a modified ZigZag transformation, which can permute an image of size $N \times N$ using $4 \times N$ pseudo-random numbers. Besides, compared with the traditional ZigZag transformation, the scanning order in the modified ZigZag transformation is controlled by a parameter and this can greatly improve the transformation efficiency and security level. Different from the existing ZigZag transformation that puts every pixel to its new coordinate in a fixed order, the modified ZigZag transformation generates a mapping

Table 1 The mapping between p, a and scan modes

a	p	
	1	2
1	Mode_1	Mode_5
2	Mode_2	Mode_6
3	Mode_3	Mode_7
4	Mode_4	Mode_8

between the old coordinates and new coordinates in random order determined by chaotic sequences. Here the used chaotic system is the enhanced logistic map introduced in [15], and it is mathematically defined as

$$x_{i+1} = E(L(x_i)) = \sin(\pi a x_i(1 - x_i)), \tag{1}$$

where a is a control parameter and $a \in (0, +\infty)$. The detailed steps of the modified ZigZag transformation on a matrix $\mathbf{M}$ of size $N \times N$ are shown as follows:

Step 1: Generate four chaotic sequences of length N. By sorting each chaotic sequence, four index vectors $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{C}_1$ and $\mathbf{C}_2$ can be obtained. Generate coordinate matrices $\mathbf{O}_1$ and $\mathbf{O}_2$ by separately using $\mathbf{R}_1$ and $\mathbf{R}_2$ as two row indices and $\mathbf{C}_1$ and $\mathbf{C}_2$ as two column indices.

Step 2: Set $p(i) = (\mathbf{R}_i(1) \mod 2) + 1$, $a(i) = (\mathbf{C}_i(1) \mod 4) + 1$, $x(1) = \mathbf{R}_1(2)$, $y(1) = \mathbf{C}_1(N)$, $x(2) = \mathbf{R}_2(2)$, $y(2) = \mathbf{C}_2(N)$, where $p(i)$ and $a(i)$ determine the scan modes of $\mathbf{O}_i$ shown in Fig 6, while $x(i)$ and $y(i)$ determine the beginning points in the two scans. Table 1 lists the mappings between p, a and the scan modes.

Step 2: Using the scan mode selected in *Step* 2 to scan $\mathbf{O}_1$ and $\mathbf{O}_2$. Then, record every coordinates on their scan paths into $\mathbf{c}_1$ and $\mathbf{c}_2$.

Step 3: Exchange the elements whose coordinates are $\mathbf{c}_1(i)$ and $\mathbf{c}_2(i)$ in the matrix M.

To better explain the modified ZigZag transformation, a numeral example of size 3 × 3 is provided and Fig. 7 shows the details of the process. First, obtain the $\mathbf{R}_1$={2,3,1}, $\mathbf{C}_1$={2,1,3}, $\mathbf{R}_2$={2,1,3}, $\mathbf{C}_2$={1,2,3}, and the coordinate matrices $\mathbf{O}_1$, $\mathbf{O}_2$. Then, calculate the scanning parameters $p(1) = (\mathbf{R}_1(1) \mod 2) + 1 = 1$, $p(2) = (\mathbf{R}_2(1) \mod 2) + 1 = 1$, $a(1) = (\mathbf{C}_1(1) \mod 4) + 1 = 3$, $a(2) = (\mathbf{C}_2(1) \mod 4) + 1 = 2$, $x(1) = \mathbf{R}_1(2) = 3$, $x(2) = \mathbf{R}_2(2) = 1$, $y(1) = \mathbf{C}_1(3) = 3$, $y(2) = \mathbf{C}_2(3) = 3$. According to Table 1, scan the $\mathbf{O}_1$ with mode_3 and scan the $\mathbf{O}_2$ with mode_2. Then, we can get the coordinate mappings $(3, 1) \to (1, 3)$, $(2, 2) \to (2, 1)$, $(2, 3) \to (3, 2)$, $(1, 2) \to (3, 1)$, $(3, 2) \to (2, 3)$, $(1, 3) \to (3, 3)$, $(2, 1) \to (1, 2)$, $(1, 1) \to (2, 2)$ and $(3, 3) \to (1, 1)$. Finally, swap the pixels in matrix $\mathbf{M}$ according to the coordinate mappings to get the swapped result $\mathbf{W}$.

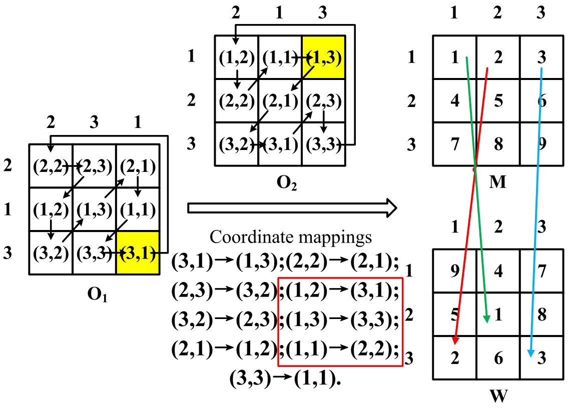

Fig. 7 An example modified ZigZag transformation

Compared with the existing ZigZag transformation, the modified ZigZag transformation can randomly select the scan order by modifying the initial states of the chaotic system, which can enhance the security level of the transformation result. Moreover, only four chaotic sequences of length N are required to permutate a matrix of size $N \times N$. However, most permutation algorithms using chaos require a chaotic sequence of length $N \times N$. Thus, the modified ZigZag transformation algorithm can greatly improve the efficiency.

3 New Image Encryption Algorithm

Using the VDT and modified ZigZag transformation, this section proposes an image encryption algorithm and Fig.8 shows the structure of the proposed encryption scheme. The algorithm mainly contains the VDT, cross-subband permutation, plaintext-related diffusion, IVDT and global permutation. The VDT divides a plain image into four subbands LL, HD, VD and DD. The cross-subband permutation randomly changes the element positions of the HD, VD and DD subbands, while the plaintext-related diffusion randomly diffuses the pixel values of LL subband. Then, after IVDT, the pixel values of the whole image are changed. The global permutation further changes the pixel positions to reduce the correlations between adjacent pixels. Two rounds of encryption are used to obtain a higher security level. Next, we will introduce each of the encryption steps in detail.

3.1 Key schedule

To resist the brute force attack, the secret key of an image encryption algorithm must have a large key space; otherwise, the attacker can crack the secret key with enumeration. The secret key of our encryption scheme is of 256 bits, which have a sufficiently large key space to resist the brute force attack. The secret key in our proposed algorithm has eight parts, namely $\mathbf{K} = \{x_0^{(1)}, x_0^{(2)}, r_0^{(1)}, r_0^{(2)}, a^{(1)}, a^{(2)}, t_0^{(1)}, t_0^{(2)}\}$. Every part is 32-bit length. The $x_0^{(1)}$ and $x_0^{(2)}$ are 32-bit float number within $[0, 1)$, and they are used to generate original initial values $\{x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}\}$ $(i = 1, 2)$ of the chaotic system in two encryption rounds. The $r_0^{(1)}$ and $r_0^{(2)}$ contain 7-bit integer and 25-bit float numbers and both of them are within $[0, 128)$. They are used to generate original control parameters $\{r_1^{(i)}, r_2^{(i)}, r_3^{(i)}, r_4^{(i)}\}(i = 1, 2)$ of the chaotic system in two encryption rounds. The $a^{(1)}$ and $a^{(2)}$ are two vectors, and each of them has four elements, namely $\{a_1^{(i)}, a_2^{(i)}, a_3^{(i)}, a_4^{(i)}\}$ (i=1, 2), and every element is an 8-bit integer. The $t_0^{(1)}$ and $t_0^{(2)}$ have the same format with $r_0^{(1)}$ and $r_0^{(2)}$. The $t_0^{(i)}$ and $a^{(i)}$ are used to disturb the original initial states and control parameters. The detailed steps to generate eight groups of original initial states and control parameters are shown as follows

$$\begin{aligned} x_j^{(i)} &= (\log_2(a_j^{(i)} + j) \times x_{j-1}^{(i)} + t_0^{(i)}) \bmod 1, \\ r_j^{(i)} &= a_j^{(i)} + r_{j-1}^{(i)} + t_0^{(i)}. \end{aligned} \tag{2}$$

where $j = 1, 2, 3, 4$, and $i = 1, 2$.

Thus, five original initial states $(x_0^{(i)}, r_0^{(i)})$, $(x_1^{(i)}, r_1^{(i)})$, $(x_2^{(i)}, r_2^{(i)})$, $(x_3^{(i)}, r_3^{(i)})$ and $(x_4^{(i)}, r_4^{(i)})$ can be obtained in each encryption round. They will be combined with the features of plain image to generate the initial states of the chaotic system in each encryption step.

3.2 Cross-Subband Permutation

The cross-subband permutation can randomly permute the elements within the HD, VD, and DD subbands. Using this strategy, one can significantly reduce the correlations among the elements in difference subbands to increase the security level. The permutation process is under the control of chaotic sequences generated by the enhanced Logistic map. To resist the chosen-plaintext attack, the average values of the three subbands are used as a part of the initial states of the chaotic map. Since the average values cannot be changed by the permutation process, one can calculate the average value in the permutation result when doing the inverse operation. Thus, the backward process has the same secret key with the forward process. For an image of size $N \times N$, the detailed steps of the cross-subband permutation in the first encryption round are shown as follows:

Step 1: Calculate the average value avg_1 of the three difference subbands, $avg_1 = \frac{\sum \text{HD} + \sum \text{VD} + \sum \text{DD}}{3 \times \frac{N}{2} \times \frac{N}{2}}$. An initial value is calculated as $y_0^{(1)} = avg_1 - \lfloor avg_1 \rfloor + x_0^{(1)}$, where $\lfloor x \rfloor$ is the largest integer that smaller than or equal to x.

Step 2: Use the initial state $(y_0^{(1)}, r_0^{(1)})$ to iterate the enhanced logistic map to generate the chaotic sequence Q_0 of length $3 \times \frac{N}{2} \times \frac{N}{2}$.

Step 3: Perform the permutation operation among HD, VD and DD subbands. Firstly, rearrange Q_0 into a three-dimensional(3D) chaotic matrix and sort the 3D chaotic matrix by the third dimension to obtain a 3D index matrix A. Merge the three difference subbands into a 3D matrix P, where $P(i, j, 1) = \text{HD}(i, j)$, $P(i, j, 2) = \text{VD}(i, j)$, $P(i, j, 3) = \text{DD}(i, j)$, $(i \in [1, \frac{N}{2}], j \in [1, \frac{N}{2}])$. Then, shuffle the elements in the three subbands using A to get the result matrix P', where $P'(i, j, k) = P(i, j, A(i, j, k)), (i \in [1, \frac{N}{2}], j \in [1, \frac{N}{2}], k \in [1, 3])$. Finally, split the result 3D matrix P' back to three difference subbands HD, VD and DD, where $\text{HD}(i, j) = P'(i, j, 1)$, $\text{VD}(i, j) = P'(i, j, 2)$, and $\text{DD}(i, j) = P'(i, j, 3)$, $(i \in [1, \frac{N}{2}], j \in [1, \frac{N}{2}])$.

Step 4: Permute the elements inside HD, DD and VD subbands. Specifically, calculate the average values avg_h, avg_v and avg_d of subbands HD, VD and DD, and

$$\begin{aligned} y_1^{(1)} &= avg_h - \lfloor avg_h \rfloor + x_1^{(1)}, \\ y_2^{(1)} &= avg_v - \lfloor avg_v \rfloor + x_2^{(1)}, \\ y_3^{(1)} &= avg_d - \lfloor avg_d \rfloor + x_3^{(1)}. \end{aligned} \tag{3}$$

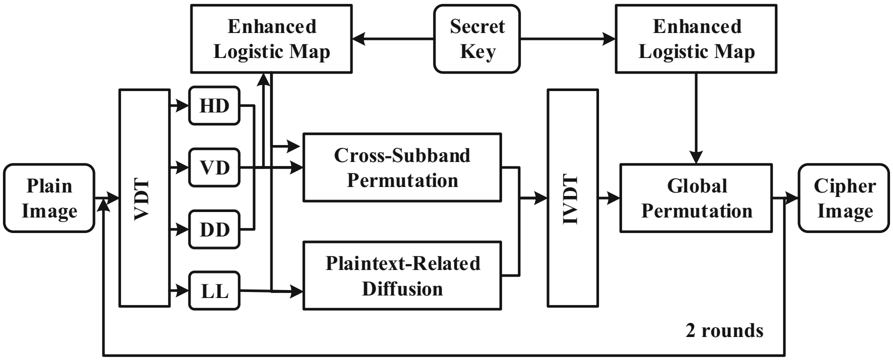

Fig. 8 The structure of the proposed image encryption scheme

Use each of the initial states $(y_1^{(1)}, r_1^{(1)})$, $(y_2^{(1)}, r_2^{(1)})$, and $(y_3^{(1)}, r_3^{(1)})$ to iterate the enhanced logistic map to generate three chaotic sequences Q_1, Q_2 and Q_3 of length $4 \times \frac{N}{2}$. Permute the HD, VD and DD subbands using the modified ZigZag transformation with the Q_1, Q_2 and Q_3 as the index sequences, respectively.

After the cross-subband permutation, the elements positions in the three subbands are totally changed, which can avoid information leakage and reduce the correlations between adjacent elements among difference subbands.

3.3 Plaintext-Related Diffusion

The diffusion process can change the pixel values and spread little change in the plaintext to the whole ciphertext. Many existing diffusion methods use the addition or XOR operation between the current plain pixel and the adjacent one or two cipher pixels. However, most of them cannot resist the chosen-plaintext attack, for the equivalent key can be constructed by choosing plaintext to encrypt [49]. To overcome this, we propose a plaintext-related diffusion. The diffusion parameters are generated by the enhanced logistic map, while the initial states are the combination of the secret key and the plaintext information. Because the LL subband contains the most important information of a plain image, we just perform the diffusion operation to the LL subband. For an image of size $N \times N$, the detailed steps of the plaintext-related diffusion are shown as follows:

Step 1: Use the initial state $(y_0^{(1)}, r_0^{(1)})$ to iterate the enhanced Logistic map to generate the chaotic sequence Q_5 of length $\frac{N}{2} \times \frac{N}{2}$. Sort Q_5 to get the index sequence $\mathbf{I}$, which is used as the diffusion sequence.

Step 2: Reshape the LL subband to a sequence of length $\frac{N}{2} \times \frac{N}{2} = \frac{N^2}{4}$, and then perform the diffusion process to the sequence using

$$\mathbf{T}(i) = \begin{cases} (\mathrm{LL}(i) + \mathbf{I}(i)) \bmod 256, & i = 1 \\ (\mathrm{LL}(i) + \mathbf{T}(i-1) + \mathbf{I}(i)) \bmod 256, & i \in [2, \frac{N^2}{4}] \end{cases} \quad (4)$$

Step 3: Perform the diffusion process to $\mathbf{T}(i)$ using Eq. (4) from the backward direction.

The plaintext-related diffusion can spread the tiny change in one pixel to the whole LL subband. Besides, the Q_5 is related to the plain image, which helps to resist the chosen-plaintext attack.

3.4 IVDT

The IVDT is used to merge the four subbands together and spread the tiny change of the plain image to the whole cipher image. The detailed steps are introduced in Sect. 2.1. As is shown in Fig. 3, after cross-subband permutation and IVDT, the pixel values of the four subbands can be changed even without any change of original pixel values. Thus, if we make a tiny change in the difference subbands, the change will also spread along the process of cross-subband permutation and IVDT, and all the pixels in difference subbands can be changed. Moreover, we also changed the pixel values of LL subbands using the plaintext-related diffusion. If we change one of the pixel values on the position of LL subband in the first encryption round, the adjacent pixel values on the difference subbands can be changed. Then, the difference can be spread all over the cipher image in the second encryption round.

3.5 Global Permutation

The global permutation is used to change the pixel positions and reduce the correlations between adjacent pixels. Firstly, we calculate the average value avg_g of the merged image. The initial value of the enhanced logistic map is calculated as $y_4^{(1)} = \lceil avg_g \rceil - avg_g + x_4^{(i)}$. Then, iterate the enhanced Logistic map using the initial states $(y_4^{(1)}, r_4^{(1)})$ to generate a chaotic sequence Q_4 with length $4 \times N$. Perform the permutation operation to the image using the modified ZigZag transformation with the chaotic sequence as the index sequence. Since the permutation result is related to the information of the image, the global permutation process also has high security level. Moreover, most existing permutation schemes need a permutation sequence of size $N \times N$. But the length of the permutation sequence in our proposed algorithm is only $4 \times N$. Thus, the proposed permutation scheme can reduce the iteration times of chaotic system, which can greatly improve the efficiency.

In the second encryption round, another five original initial states $(x_0^{(2)}, r_0^{(2)})$, $(x_1^{(2)}, r_1^{(2)})$, $(x_2^{(2)}, r_2^{(2)})$, $(x_3^{(2)}, r_3^{(2)})$ and $(x_4^{(2)}, r_4^{(2)})$ are used and the procedures are the same with that in the first encryption round.

3.6 Discussion

In the proposed image encryption scheme, the permutation scheme can permute the pixels of plain image with a high efficiency, a new diffusion method using the VDT is proposed, and the plaintext-related information is combined with the secret key. With these techniques, the proposed scheme can obtain the following advantages:

(1) The proposed scheme can achieve a high efficiency, because the permutation scheme using the modified ZigZag transformation can reduce the iteration times of chaotic system. Besides, a selective diffusion strategy is used. The diffusion only needs to diffuse the LL subband, because the LL subband contains most information of the image.

(2) The proposed encryption scheme has a high ability to resist many security attacks such as the chosen-plaintext attack, static and differential attacks. This is because we combine the plaintext-related information into the secret key. This makes it difficult to construct the equivalent key. Note that these plaintext-related information is unnecessary when decrypting the image so that the proposed image encryption scheme is a symmetric key encryption algorithm.

4 Experimental results and security analysis

In this section, we simulate the proposed image encryption scheme and analyze its security level.

4.1 Simulation results

An effective image encryption scheme should be able to encrypt a plain image into an unrecognized cipher image. One cannot get any useful information about the plain image from the cipher image without the correct secret key. Figure 9 simulates the encryption and decryption processes for the proposed encryption scheme using different grayscale, binary and color images. The used binary images are all-one and all-zero images. Figure 9(a) shows the plain images, and Figure 9(b) plots their histograms. As can be seen, the histograms contain much significant information about the plain images. However, the proposed image encryption scheme can encrypt the plain images into unrecognized cipher images which are shown in Fig. 9(c). Fig. 9(d) plots the histograms of these cipher images. Obviously, the distribution of the pixels in the cipher images is random-like, which can well protect the contents of the plain images. Finally, using the correct secret key, one can completely recover the plain images and the recovered images are shown in Fig. 9(e). Thus, the proposed image encryption scheme can encrypt different kinds of images into unrecognized cipher images with a high visual security, which can help to resist the statistical attacks effectively.

4.2 Efficiency analysis

To adapt the rapid increment of image data capacity, an image encryption algorithm must have a high encryption speed. This means that it is necessary to keep a balance between the encryption efficiency and security level. Our proposed encryption scheme can achieve fast encryption speed because of the following reasons:

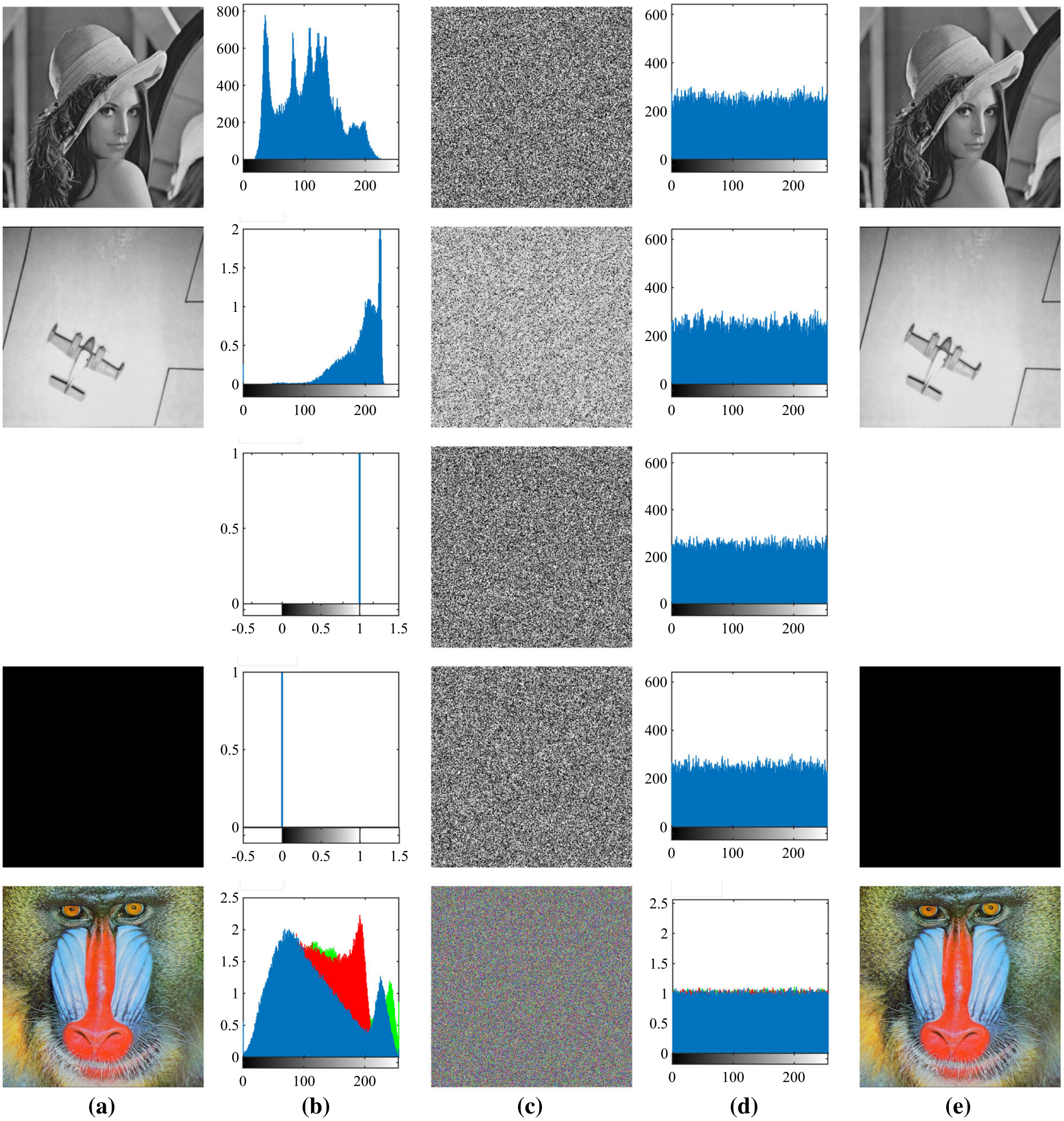

Fig. 9 Simulation results of the proposed image encryption algorithm: **a** plain images include the grayscale, binary, and color images; **b** histograms of **a**; **c** encrypted results of **a**; **d** histograms of **c**; **e** decrypted results of **c**

(1) It can achieve a good diffusion performance by performing the diffusion operation to only a part of image. On the contract, traditional diffusion operations are performed to the whole image pixels.

(2) When doing the permutation process to an image of size $M \times N$, our proposed scheme only need four chaotic sequences. Two of them are of length $\frac{M}{2}$, and the other two are of length $\frac{N}{2}$. This can greatly reduce the iteration numbers of the used chaotic system.

To demonstrate the efficiency of our proposed image encryption scheme, we compare its efficiency with

Table 2 Average encryption times (s) of different image encryption algorithms for different sizes of plain images

Image size	Algorithms				
	Ours	Hua [16]	Xu [42]	Zhou [47]	Wang [38]
128 × 128	0.0178	0.0201	0.0199	0.1934	0.1253
256 × 256	0.0689	0.0791	0.0832	0.7314	2.3261
512 × 512	0.3395	0.3706	0.3778	2.8625	10.8785

several state-of-the-art encryption algorithms. All the algorithms are simulated on a computer with the following environment: Intel(R) Core(TM) i5-8300H CPU running at 2.30 GHz, 8 GB random-access memory, and a Windows 10 operating system. Table 2 shows the encryption time of several image encryption algorithms for different sizes of plain images. The results are obtained by calculating the average encryption time of 100 experiments. It can be seen that our proposed encryption scheme requires the shortest encryption time, which means it has relatively higher efficiency.

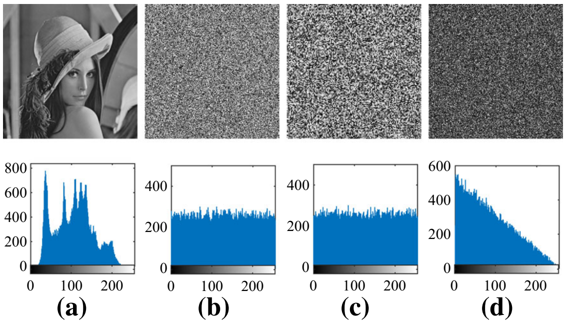

Fig. 10 Key sensitivity analysis in encryption process: **a** plain image P; **b** cipher image $\mathbf{C}_1$=$Enc(\mathbf{P}, K_1)$; (c) cipher image $\mathbf{C}_2$ = $Enc(\mathbf{P}, K_2)$; (d) difference between $\mathbf{C}_1$ and $\mathbf{C}_2$, $|\mathbf{C}_1 - \mathbf{C}_2|$

4.3 Key sensitivity

An encryption algorithm should have very sensitive secret key. Otherwise, its actual key space may far smaller than the theoretical one, which can increase the risk of being broken by brute force attack. If an encryption algorithm is sensitive to its secret key, a tiny change on the secret key can cause total difference to the cipher image in the encryption process and cause total difference to the recovered image in the decryption process. Several experiments are designed to test the key sensitivity of the proposed image encryption scheme. First, we randomly generate a secret key $\mathbf{K}_1$ and then obtain two other secret keys $\mathbf{K}_2$ and $\mathbf{K}_3$ by changing one bit in $\mathbf{K}_1$. The three obtained secret keys are expressed as

$$\begin{aligned}\mathbf{K}_1 &= E794\mathbf{E}FE7F092B5613033D363D3CCCA\\&79CB0BCA0315885EC08F7AAF823B4BE2B\mathbf{1},\\\mathbf{K}_2 &= E794\mathbf{F}FE7F092B5613033D363D3CCCA\\&79CB0BCA0315885EC08F7AAF823B4BE2B1,\\\mathbf{K}_3 &= E794EFE7F092B5613033D363D3CCCA\\&79CB0BCA0315885EC08F7AAF823B4BE2B\mathbf{2}.\end{aligned}$$

Figure 10 shows the key sensitive analysis results in encryption process. Figure 10(a) shows the plain image and its histogram, and Fig. 10(b) and (c) shows the encryption results of Fig. 10(a) using the secret key $\mathbf{K}_1$ and $\mathbf{K}_2$, respectively. Figure 10(d) shows the difference between the two cipher images. As shown that the cipher images encrypted by the two slightly different keys are completely different, which indicates the high sensitivity of the secret key in encryption process. Figure 11 shows the key sensitivity analysis results in decryption process. Figure 11(a) shows the cipher image encrypted by $\mathbf{K}_1$ and its histogram. Figure 11(b) shows the recovered image of (a) using the secret key $\mathbf{K}_1$. Figure 10(c) and (d) shows the decryption results of Fig. 10(a) using $\mathbf{K}_2$ and $\mathbf{K}_3$, respectively. Figure 11(e) shows the difference of Figs 10(c) and (d). It can be seen that only using the correct key, one can recover the plain image completely. Even a bit change in the secret key can lead to a completely different result. Therefore, the secret key of the proposed image encryption scheme is highly sensitive.

Besides, we quantitatively test the key sensitivity of our proposed image encryption scheme using the number of bit change rate (NBCR). For two data sequences Z_1 and Z_2 with the same size, their NBCR is defined as

$$\text{NBCR} = \frac{H(Z_1, Z_2)}{L_b} \times 100\%, \tag{5}$$

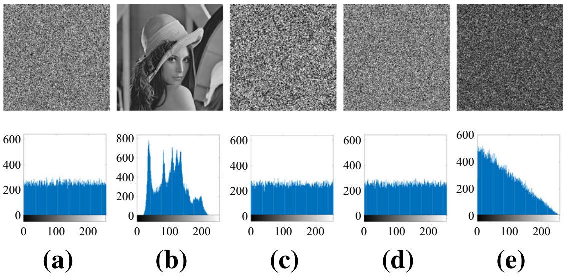

Fig. 11 Key sensitivity analysis in decryption process: **a** cipher image $\mathbf{C}_1$; **b** decrypted result $\mathbf{D}_1 = Dec(\mathbf{C}_1, \mathbf{K}_1)$; **c** decrypted result $\mathbf{D}_2 = Dec(\mathbf{C}_1, \mathbf{K}_2)$; **d** decrypted result $\mathbf{D}_3 = Dec(\mathbf{C}_1, \mathbf{K}_3)$; **e** difference between $\mathbf{D}_2$ and $\mathbf{D}_3$, $|\mathbf{D}_2 - \mathbf{D}_3|$

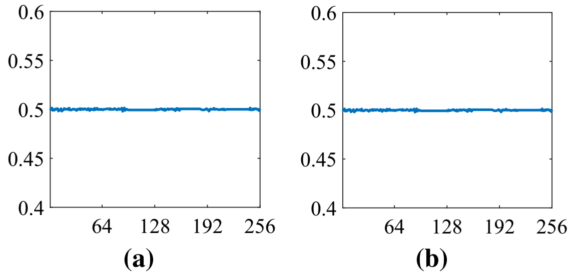

Fig. 12 The NBCR results of the proposed image encryption scheme in encryption and decryption processes

where L_b is the length of Z_1 or Z_2, and $H(Z_1, Z_2)$ calculates the Hamming distance of Z_1 and Z_2. The ideal value of NBCR is 50%, and it can be achieved when the two data sequences are completely independent.

Our experiment is designed as follows. For a given secret key $\mathbf{K}_1$, we change one of its 256 bits to obtain $\mathbf{K}_4$. To calculate the NBCR of encryption process, we encrypt the same image using $\mathbf{K}_1$ and $\mathbf{K}_4$ and obtain the cipher images $\mathbf{C}_1$ and $\mathbf{C}_4$. Then, the NBCR of $\mathbf{C}_1$ and $\mathbf{C}_4$ is calculated. To calculate the NBCR in decryption process, we firstly decrypt $\mathbf{C}_1$ using $\mathbf{K}_1$ and $\mathbf{K}_4$ to obtain the decryption results $\mathbf{D}_1$ and $\mathbf{D}_4$. Similarly, the NBCR of $\mathbf{D}_1$ and $\mathbf{D}_4$ are calculated. Every bit of $\mathbf{K}_1$ is changed separately to measure the NBCR in encryption and decryption process, and Fig. 12 shows the test result. It is shown that the NBCRs in the encryption and decryption processes are all close to the ideal value 0.5 for every bit, which also means that the proposed encryption scheme has highly sensitive secret key.

4.4 Ability to Resist Differential Attack

The differential analysis is a commonly used and effective cryptanalysis technique. An image encryption algorithm should have high ability to resist this attack. Our proposed scheme can resist the differential attack due to the following properties: (1) The cross-subband permutation and IVDT can efficiently spread the tiny change of plain image to the whole cipher image. (2) The combination of plaintext-related diffusion and global permutation further spread the tiny change.

The number of pixel change rate (NPCR) and uniform average change intensity (UACI) are two criteria to test the ability of an encryption algorithm in resisting the differential attack [33]. Suppose that C_1 and C_2 are two cipher images with only one bit difference in their plain images, then their NPCR and UACI can be calculated as

$$\mathrm{NPCR} = \frac{\Sigma_{j,k} A(j,k)}{L \times M} \times 100\% \tag{6}$$

and

$$\mathrm{UACI} = \frac{1}{L \times M} \Sigma_{j,k} \left(\frac{C_1(j,k) - C_2(j,k)}{255}\right) \times 100\% \tag{7}$$

respectively. The $A(j,k) = 1$ if $C_1(j,k) \neq C_2(j,k)$; otherwise $A(j,k) = 0$. If the NPCR result is greater than the threshold value ϑ_α, it can pass the NPCR test [39]. The threshold is calculated as

$$\vartheta_\alpha = \frac{L - \phi^{-1}(\alpha)\sqrt{\frac{L}{MN}}}{L+1} \tag{8}$$

where α is a significance level and L is the largest allowed value of the image, $L = 255$ in an 8-bit grayscale image, M and N are the height and width of an image, respectively. A greater NPCR value means stronger ability to resist differential attack. Meanwhile, in UACI test, if the result UACI value falls into the interval $(\theta_\alpha^{*-}, \theta_\alpha^{*+})$, it passes the UACI test. The interval is calculated as

$$\begin{cases} \theta_\alpha^{*-} = \frac{L+2}{3L+3} - \phi^{-1}(\frac{\alpha}{2})\frac{(L+2)(L^2+2L+3)}{18(l+1)^2 LMN} \\ \theta_\alpha^{*+} = \frac{L+2}{3L+3} + \phi^{-1}(\frac{\alpha}{2})\frac{(L+2)(L^2+2L+3)}{18(l+1)^2 LMN} \end{cases} \tag{9}$$

A closer UACI value to the median value of the interval indicates a stronger ability to resist the differential attack. Following the setting given in [39], our experiments set the significance level $\alpha = 0.05$. Thus, for different sizes of images, the criteria are

Table 3 The NPCR and UACI results of our proposed image encryption scheme for different sizes of images

Image size	File name	NPCR	UACI	Test results
128 × 128	lena128	99.6704	33.4788	Pass
	flower	99.6460	33.5205	Pass
	workshop	99.6521	33.4135	Pass
256 × 256	lena256	99.6689	33.4590	Pass
	5.1.09	99.6017	33.4446	Pass
	5.1.10	99.6170	33.4425	Pass
512 × 512	lena512	99.6227	33.4959	Pass
	5.2.08	99.6090	33.4653	Pass
	5.2.09	99.6235	33.4335	pass
1024 × 1024	lena1024	99.6084	33.4254	Pass
	5.3.01	99.6146	33.4514	Pass
	5.3.02	99.6120	33.4615	Pass

Table 4 The NPCR results of different image encryption schemes with the pixel change in the border (α =0.05)

Position	Methods					
	Ours	Cao [2]	Ping [28]	Hua [16]	Wang [38]	Xu [42]
(1, 1)	99.6307	99.5575	99.6033	99.5987	99.6460	99.5605
(256, 256)	99.6292	99.6017	99.5987	99.6277	99.6490	99.6262
(1, 256)	99.6216	99.6292	99.6216	99.5712	99.5926	99.6262
(256, 1)	99.6307	99.6155	99.5895	99.6246	94.7647	99.6140
(127, 128)	99.6216	99.6445	99.6094	99.6246	99.6201	99.6078
Mean	99.6268	99.6097	99.6045	99.6094	98.6545	99.6069

Table 5 The UACI results of different image encryption schemes with the pixel change in the border (α =0.05)

Position	Methods					
	Ours	Cao [2]	Ping [28]	Hua [16]	Wang [38]	Xu [42]
(1,1)	33.4604	33.4867	33.5353	33.5732	33.3683	33.3448
(256, 256)	33.4559	33.5465	33.6461	33.5167	33.4740	33.5155
(1, 256)	33.4477	33.4768	33.4004	33.4105	33.3579	33.5146
(256, 1)	33.5709	33.4169	33.4339	33.4397	31.7191	33.6503
(127, 128)	33.4477	33.5189	33.4289	33.5438	33.5116	33.5840
Offset	0.0300	0.0442	0.0764	0.0640	0.4008	0.1057

as follows: for an image of size 128 × 128, $\vartheta_\alpha = 99.5292\%$ and $(\theta_\alpha^{*-}, \theta_\alpha^{*+}) = (33.1012\%, 33.8259\%)$; for an image of size 256 × 256, $\vartheta_\alpha = 99.5693\%$ and $(\theta_\alpha^{*-}, \theta_\alpha^{*+}) = (33.2824\%, 33.6447\%)$; for an image of size 512 × 512, $\vartheta_\alpha = 99.5893\%$ and $(\theta_\alpha^{*-}, \theta_\alpha^{*+}) = (33.3730\%, 33.5541\%)$; and for an image of size 1024 × 1024, $\vartheta_\alpha = 99.5994\%$ and $(\theta_\alpha^{*-}, \theta_\alpha^{*+}) = (33.4183\%, 33.5088\%)$. Twelve grayscale images are tested, and the results are shown in Table 3. It is shown that all the images pass the tests, which means that our proposed image encryption algorithm can effectively resist the differential analysis.

Then, we choose five special positions (1,1), (1,256), (256,1), (256,256) and (127,128) in a plain image of

Table 6 The NPCR and UACI results for different image encryption schemes

Analysis	Algorithm	Images			
		Lena	Plane	FingerPrint	Mickey
NPCR	Ours	99.6826	99.6796	99.6658	99.6552
	Cao [2]	99.6292	99.6597	99.6414	99.6063
	Ping [28]	99.6277	99.6460	99.6552	99.6506
	Hua [16]	99.6231	99.5926	99.5895	99.6063
	Wang [38]	99.6140	99.6262	99.6658	99.6490
	Xu [42]	99.5941	99.6292	99.6613	99.6109
UACI	Ours	33.4518	33.4611	33.4467	33.4618
	Cao [2]	33.4769	33.4839	33.3784	33.4666
	Ping [28]	33.3175	33.4870	33.4960	33.4906
	Hua [16]	33.4791	33.5363	33.4862	33.4200
	Wang [38]	33.5311	33.5428	33.4937	33.4876
	Xu [42]	33.4794	33.4740	33.4440	33.4749

size 256 × 256, and separately change only one bit of the pixels to obtain five changed images. Encrypt the changed images, and calculate the NPCR and UACI values of the cipher images. Tables 4 and 5 show the NPCR and UACI test results of above tests for several different image encryption schemes. It is shown that our proposed scheme can almost achieve the largest NPCR values and its UACI values are closest to the median value of the interval, namely 33.46355%. Thus, our proposed scheme can spread the tiny change of plain image into the whole cipher image, even if the changed pixels are at the border of the image.

Finally, we randomly change on bit in a plain image of size 256 × 256 and calculate the NPCR and UACI values of the cipher images encrypted from the original and changed images. Table 6 shows the comparison results of the best NPCR and UACI results in 100 times of experiments. It is shown that the proposed image encryption scheme can achieve relatively higher NPCR values, and its UACI values are closest to the median value of the interval. In conclusion, the proposed algorithm shows strong ability to resist the differential attack from this aspect.

4.5 Correlation analysis

It is well known that high correlations exist in adjacent pixels of a natural image. As a result, natural images may contain much visual information. An effective image encryption algorithm should have the ability to reduce the correlations between adjacent pixels. The correlation coefficient of two pixel sequences can be calculated as

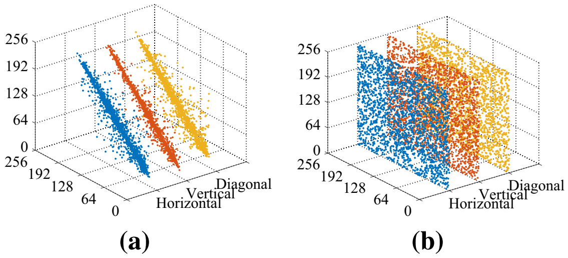

Fig. 13 Distribution of adjacent pixel pairs in **a** the plain image and **b** the cipher image encrypted by the proposed image encryption scheme

$$\mathrm{Corr}(X, Y) = \left| \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \right| \tag{10}$$

where $E(\cdot)$, μ and σ represent the mathematical expectation, mean and standard deviation, respectively. Meanwhile, X and Y are two sequences with adjacent pixels along horizontal, vertical or diagonal directions. If X and Y exhibit a high correlation, the correlation coefficient is close to 1. Otherwise, it is close to 0. Thus, two adjacent pixel sequences have weaker correlation if their correlation coefficient is lower.

To analyze the ability of reducing the high correlation coefficients between adjacent pixels in our proposed image encryption scheme, we first plot the adja-

Table 7 The correlation coefficients of the plain images with different sizes and their cipher images encrypted by the proposed encryption scheme

Image size	File name	Directions/Origin			Directions/Cipher-images		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
128 × 128	lena128	0.94300000	0.90330000	0.84180000	−0.00120000	−0.00078296	−0.00160000
	flower	0.91930000	0.94730000	0.89040000	−0.00890000	0.00170000	0.00800000
	workshop	0.95500000	0.96290000	0.95780000	0.00540000	−0.00390000	0.00220000
256 × 256	lena256	0.97040000	0.94630000	0.91250000	−0.00190000	0.00390000	0.00160000
	5.1.09	0.93390000	0.89910000	0.89040000	0.00280000	−0.00062627	−0.00190000
	5.1.10	0.85480000	0.89690000	0.81580000	0.00370000	−0.00260000	−0.00160000
512 × 512	lena512	0.99270000	0.98710000	0.97380000	0.00450000	−0.00250000	0.00820000
	5.2.08	0.87840000	0.92560000	0.85700000	0.00920000	−0.00790000	0.00300000
	5.2.09	0.85230000	0.90110000	0.79450000	0.00470000	−0.00500000	−0.00930000
1024 × 1024	lena1024	0.99790000	0.99590000	0.99350000	−0.00200000	−0.00170000	0.00120000
	5.3.01	0.98140000	0.97690000	0.96950000	0.00590000	0.00670000	0.00290000
	5.3.02	0.90380000	0.91520000	0.85740000	0.00490000	0.01190000	0.00140000

Table 8 The correlation coefficients of the plain images and their cipher images encrypted by different encryption schemes

Images	Direction	Methods						
		Origin	Ours	Cao [2]	Ping [28]	Hua [16]	Wang [38]	Xu [42]
Lena	Horizontal	0.9704	−0.00190000	0.00680000	0.00870000	0.00300000	−0.00680000	−0.00520000
	Vertical	0.9463	0.00390000	−0.00130000	−0.00074170	0.00380000	−0.00230000	−0.00200000
	Diagonal	0.9125	0.00160000	−0.00230000	0.01460000	0.00290000	0.00200000	−0.00910000
Plane	Horizontal	0.9728	−0.00043330	0.00400000	−0.00650000	−0.00660000	−0.00750000	−0.00580000
	Vertical	0.9555	0.00290000	0.00560000	0.00110000	0.00400000	0.00280000	−0.00290000
	Diagonal	0.9221	−0.00100000	−0.00500000	−0.00960000	−0.00020357	−0.00450000	0.00500000
FingerPrint	Horizontal	0.9455	−0.00071871	−0.00270000	0.00300000	0.00250000	−0.00160000	0.00700000
	Vertical	0.9191	−0.00110000	−0.00160000	−0.01190000	0.00740000	0.00630000	−0.00370000
	Diagonal	0.8199	0.00370000	−0.00730000	0.00007995	−0.00019036	0.00600000	−0.00290000
Mickey	Horizontal	0.9351	0.00410000	0.00860000	0.00600000	0.00330000	−0.00030238	0.00500000
	Vertical	0.9333	0.00310000	0.00980000	0.00680000	0.00650000	0.00370000	−0.00730000
	Diagonal	0.9291	−0.00024847	−0.00300000	0.00640000	0.00100000	0.00610000	−0.00100000

cent pixel pairs of the plain image and its cipher image and Fig. 13 shows the result. As can be seen, the pixel pairs in plain image mainly distribute on the diagonal line, which indicates the high correlation coefficients between adjacent pixels. However, the pixel pairs in cipher image are randomly distributed in the whole plane. The results show that our image encryption scheme can effectively reduce the correlations between adjacent pixels.

Table 7 exhibits the correlation values of twelve images in different sizes. All the correlation values of cipher images are far smaller than that of plain images, which means that our image encryption scheme can effectively reduce the correlations between adjacent pixels of images with different sizes. Table 8 shows the comparison results of correlation values of cipher images encrypted by different image encryption schemes. It shows that our proposed scheme has the smallest absolute correlation values, compared with other encryption schemes.

4.6 Ability to resist chosen-plaintext attack

The chosen-plaintext attack is another effective crypt-analysis technique. During the attack, attackers can construct a group of special images, such as all-one and all-zero images. After encrypting them using the encryption scheme, the attackers can get some information about the encryption scheme and further construct the equivalent key to break the encryption scheme.

In our proposed scheme, we use the plaintext-related information as a part of the initial states of chaotic system. Using this strategy, the chosen-plaintext attack may be ineffective, because different plain images may result in different initial states of the chaotic system. Note that the plaintext-related information can be extracted in the decryption process. This is because we use the average of pixels to generate the initial state and the permutation operation cannot change the average of pixels. Once the input image is changed, the average value may be also changed. Then, our proposed scheme has high ability to resist the chosen-plaintext attack.

5 Conclusion

In this paper, we first proposed a value-differencing trans-formation (VDT) and a modified ZigZag transformation. The VDT can split the plain image into four difference subbands LL, HD, VD and DD according to the relative positions and the value-differencing between the adjacent pixels. The modified ZigZag transformation can separate the image pixels with a high efficiency. Using the VDT and modified ZigZag transformation, we further proposed a new image encryption scheme. It uses the VDT to selectively diffuse the image pixels and utilizes the modified ZigZag transformation to permute the pixel positions. To improve the ability to defense the chosen-plaintext attack, we extract part information of plain image as parameters to encrypt the image. The simulation results show that our proposed algorithm can encrypt different kinds of digital images into unrecognized cipher images, and one can completely recover the plain image with the correct secret key. The comparison results show that our proposed scheme has a relatively higher efficiency and it has a high ability to resist different security attacks, including the differential attack and chosen-plaintext attack, and can outperform several other image encryption algorithms.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 62071142, 62106063 and 62001304 and the Guangdong Basic and Applied Basic Research Foundation under Grants 2021A15 15011406 and 2019A1515110410 and the Shenzhen College Stability Support Plan under Grants GXWD20201230155427003-20200824210638001 and GXWD20201230155427003-20200 824113231001.

Availability of data and material The used datasets are openly available from http://sipi.usc.edu/database/ and http://decsai.ugr.es/cvg/dbimagenes/, and the data used to support the findings of this study are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Bao, H., Hua, Z., Liu, W., Bao, B.: Discrete memristive neuron model and its interspike interval-encoded application in image encryption. Science China Technological Sciences (to be published, 2021)
2. Cao, C., Sun, K., Liu, W.: A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. Sig. Proc. **143**, 122–133 (2018)
3. Chai, X., Chen, Y., Broyde, L.: A novel chaos-based image encryption algorithm using DNA sequence operations. Opt. Las. Eng. **88**, 197–213 (2017)
4. Chai, X., Zheng, X., Gan, Z., Han, D., Chen, Y.: An image encryption algorithm based on chaotic system and compressive sensing. Sig. Proc. **148**, 124–144 (2018)
5. Chen, J., Chen, L., Zhou, Y.: Cryptanalysis of a DNA-based image encryption scheme. Inf. Sci. **520**, 130–141 (2020)
6. Chen, S., Chang, C.C., Liao, Q.: Fidelity preserved data hiding in encrypted images based on homomorphism and matrix embedding. IEEE Access **8**, 22345–22356 (2020)
7. Courtois, N., Drobick, J., Schmeh, K.: Feistel ciphers in east germany in the communist era. Cryptologia **42**(5), 427–444 (2018)
8. Dahl, M.: 2D fast Fourier transform image encryption. J. Appl. Eng. Math. 5 (2018)
9. Dong, X., Dong, B., Wang, X.: Quantum attacks on some feistel block ciphers. Des. Codes Cryptogr. pp. 1–25 (2020)
10. Farah, M.B., Guesmi, R., Kachouri, A., Samet, M.: A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. Opt. Laser Technol. **121**, 105,777 (2020)
11. Feistel, H.: Cryptography and computer privacy. Scientific American **228**(5), 15–23 (1973)
12. Gong, L., Qiu, K., Deng, C., Zhou, N.: An optical image compression and encryption scheme based on compressive sensing and RSA algorithm. Opt. Las. Eng. **121**, 169–180 (2019)
13. Hua, Z., Li, J., Chen, Y., Yi, S.: Design and application of an S-box using complete Latin square. Nonlin. Dyn. **104**, 807–825 (2021)
14. Hua, Z., Zhang, K., Li, Y., Zhou, Y.: Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing. Signal Proc. **183**, 107,998 (2021)
15. Hua, Z., Zhou, B., Zhou, Y.: Sine chaotification model for enhancing chaos and its hardware implementation. IEEE Trans. Ind. Elect. **66**(2), 1273–1284 (2018)
16. Hua, Z., Zhou, Y., Huang, H.: Cosine-transform-based chaotic system for image encryption. Inf. Sci. **480**, 403–419 (2019)
17. Hua, Z., Zhu, Z., Chen, Y., Li, Y.: Color image encryption using orthogonal Latin squares and a new 2D chaotic system. Nonlin. Dyn. **104**, 4505–4522 (2021)
18. Huang, Z.J., Cheng, S., Gong, L.H., Zhou, N.R.: Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform. Opt. Las. Eng. **124**, 105,821 (2020)
19. Lan, R., He, J., Wang, S., Gu, T., Luo, X.: Integrated chaotic systems for image encryption. Sig. Proc. **147**, 133–145 (2018)
20. Li, C.L., Zhou, Y., Li, H.M., Feng, W., Du, J.R.: Image encryption scheme with bit-level scrambling and multiplication diffusion. Multim. Tools Appl. pp. 1–23 (2021)
21. Li, H., Hua, Z., Bao, H., Zhu, L., Chen, M., Bao, B.: Two-dimensional memristive hyperchaotic maps and application in secure communication. IEEE Trans. Ind. Electron. **68**, 9931–9940 (2021)
22. Li, M., Lu, D., Xiang, Y., Zhang, Y., Ren, H.: Cryptanalysis and improvement in a chaotic image cipher using two-round permutation and diffusion. Nonlin. Dyn. **96**(1), 31–47 (2019)
23. Li, M., Zhou, K., Ren, H., Fan, H.: Cryptanalysis of permutation-diffusion-based lightweight chaotic image encryption scheme using CPA. Appl. Sci. **9**(3), 494 (2019)
24. Li, P., Lo, K.T.: Joint image encryption and compression schemes based on 16× 16 DCT. J. Vis. Commun. Image Rep. **58**, 12–24 (2019)
25. Luo, Y., Zhang, S., Liu, J., Cao, L.: Cryptanalysis of a chaotic block cryptographic system against template attacks. Int. J. Bifur. Chaos **30**(15), 2050,223 (2020)
26. Ma, Y., Li, C., Ou, B.: Cryptanalysis of an image block encryption algorithm based on chaotic maps. J. Inf. Sec. Appl. **54**, 102,566 (2020)

27. Nachef, V., Patarin, J., Volte, E.: Feistel ciphers. Springer, Cham (2017)
28. Ping, P., Xu, F., Mao, Y., Wang, Z.: Designing permutation-substitution image encryption networks with Henon map. Neurocomputing **283**, 53–63 (2018)
29. Ponuma, R., Amutha, R., Aparna, S., Gopal, G.: Visually meaningful image encryption using data hiding and chaotic compressive sensing. Multim. Tools Appl. **78**(18), 25707–25729 (2019)
30. Priya, S., Santhi, B.: A novel visual medical image encryption for secure transmission of authenticated watermarked medical images. Mobile Netw. Appl. pp. 1–8 (2019)
31. Singh, H.: Watermarking image encryption using deterministic phase mask and singular value decomposition in fractional Mellin transform domain. IET Image Proc. **12**(11), 1994–2001 (2018)
32. Su, Y., Wo, Y., Han, G.: Reversible cellular automata image encryption for similarity search. Sig. Proc. Image Comm. **72**, 134–147 (2019)
33. Ullah, A., Jamal, S.S., Shah, T.: A novel scheme for image encryption using substitution box and chaotic system. Nonlin. Dyn. **91**(1), 359–370 (2018)
34. Vaish, A., Kumar, M.: Color image encryption using MSVD, DWT and Arnold transform in fractional Fourier domain. Optik **145**, 273–283 (2017)
35. Wan, M., Li, M., Yang, G., Gai, S., Jin, Z.: Feature extraction using two-dimensional maximum embedding difference. Inf. Sci. **274**, 55–69 (2014)
36. Wang, H., Xiao, D., Chen, X., Huang, H.: Cryptanalysis and enhancements of image encryption using combination of the 1d chaotic map. Sig. Proc. **144**, 444–452 (2018)
37. Wang, X., Gao, S.: Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. Inf. Sci. **507**, 16–36 (2020)
38. Wang, X., Zhu, X., Zhang, Y.: An image encryption algorithm based on Josephus traversing and mixed chaotic map. IEEE Access **6**, 23733–23746 (2018)
39. Wu, Y., Noonan, J.P., Agaian, S., et al.: NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology, J. Select. Areas Telecomm. (JSAT) **1**(2), 31–38 (2011)
40. Xiao, B., Luo, J., Bi, X., Li, W., Chen, B.: Fractional discrete Tchebyshev moments and their applications in image encryption and watermarking. Inf. Sci. **516**, 545–559 (2020)
41. Xu, C., Sun, J., Wang, C.: An image encryption algorithm based on random walk and hyperchaotic systems. Int. J. Bifur. Chaos **30**(04), 2050,060 (2020)
42. Xu, L., Li, Z., Li, J., Hua, W.: A novel bit-level image encryption algorithm based on chaotic maps. Opt. Las. Eng. **78**, 17–25 (2016)
43. Ye, H.S., Zhou, N.R., Gong, L.H.: Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion. Sig. Proc. **175**, 107,652 (2020)
44. Yin, Z., Niu, X., Zhang, X., Tang, J., Luo, B.: Reversible data hiding in encrypted ambtc images. Multim. Tools Appl. **77**(14), 18067–18083 (2018)
45. Zhang, X., Zhou, Z., Niu, Y.: An image encryption method based on the Feistel network and dynamic DNA encoding. IEEE Photon. J. **10**(4), 1–14 (2018)
46. Zhou, M., Wang, C.: A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. Sig. Proc. **171**, 107,484 (2020)
47. Zhou, Y., Bao, L., Chen, C.P.: A new 1D chaotic system for image encryption. Sig. Proc. **97**, 172–182 (2014)
48. Zhou, Y., Li, C., Li, W., Li, H., Feng, W., Qian, K.: Image encryption algorithm with circle index table scrambling and partition diffusion. Nonlin. Dyn. pp. 1–19 (2021)
49. Zhu, C., Wang, G., Sun, K.: Cryptanalysis and improvement on an image encryption algorithm design using a novel chaos based S-box. Symmetry **10**(9), 399 (2018)

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