Multimedia Tools and Applications (2022) 81:26535–26589
https://doi.org/10.1007/s11042-022-12139-0

A novel design of multiple image encryption using perturbed chaotic map

Thang Manh Hoang[1]

Received: 17 May 2021 / Revised: 22 November 2021 / Accepted: 3 January 2022 /
Published online: 16 May 2022

Abstract

For recent decades, the increasing volume of multimedia data has been witnessed, and the data is required technical methods to assure the security for storage and transmission. Chaos-based encryption is one of promising approaches to keep large volume of data confidential. Most of chaos-based algorithms were proposed for single image encryption. Recently, several schemes were proposed for multiple image encryption, and all of them are designed to work in a single round of encryption. In addition, the dynamics of chaotic maps therein are stationary, so it does not provide advantage of uncertainty of chaotic orbits for the security. Moreover, a chaotic map being realized in digital platforms can produce a large number of bits, and so far those bits have not been used efficiently to encrypt larger volume of data. In this paper, a novel design of chaos-based multiple image encryption is proposed using the permutation-diffusion architecture for the first time. Any chaotic map can be employed for the proposed design. Chaotic dynamics are non-stationary by means of perturbation on state variables and control parameters in bit level. Amounts of perturbation are constructed from the coordinate of pixels and the content of plain images respectively in the pixel permutation and diffusion processes, so the proposed design provides the property of authentication. Values of chaotic state variables are represented in fixed-point number, and bits generated by chaotic maps are thoroughly exploited to encrypt multiple images at the same time. The specific example will demonstrate the effectiveness of the proposed design by means of the statistical and security analyses. The simulation results will show its resistance from the attacking method of differential analysis, and those are also compared with those of other existing algorithms.

Keywords Digital perturbed chaos · Perturbed chaotic map · Chaos-based image encryption · Multiple image encryption (MIE)

✉ Thang Manh Hoang
thang.hoangmanh@hust.edu.vn

[1] School of Electrical and Electronic Engineering, and Vietnam-Japan International Institute for Science of Technology, Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam

1 Introduction

Since chaos theory were introduced by E. Lorenz, it has been intensively researched in various fields of science and engineering [23] such as physics, chemistry, biology, etc. Besides, chaos has been interested in different areas of engineering applications e.g. information security [33], communications [65, 73], electronic circuits [30]. In the application of information security, chaotic systems are employed in two main aspects, i.e. cryptography and stenography. Similar to general approaches of stegnography, chaos-based steganographic is with the use of chaotic sequences in hiding the secret data under a cover image in either the spatial domain or the transform domain. For the spatial domain, the secret data is embedded in least significant bits of pixels of the host image with the support of chaotic sequence (e.g. [27, 45, 60]). In another way with the transform domain, chaotic sequences play a role to locate transformed coefficients (e.g. [57, 71, 72]) or together with the secret data in modulating transformed coefficients [55, 69]. Many algorithms were reported in order to improve the quality, performance and security of image steganography. With the use of chaotic sequences, two prominent types of techniques are to improve the security, i.e. the combinations of chaotic sequences and other techniques, such as transformation, DNA computing, and neural networks, etc. Chaotic values are used for encrypting the secret data before introducing into the cover image in transform domains e.g. [55, 69] or for identifying the coordinate of pixels where the secret data elements being embedded, e.g. [27]. Chaotic sequences assist to determine variable bit embedding in sub-bands of transform domains [15] or it can be combined with DNA computing in construction of steganographic algorithms, e.g. [17, 28, 43]. According to recent report by Alan A. Abdulla et al. [2], there are four main requirements for a steganography which must be met, i.e. (i) stego-image quality, (ii) hiding capacity, (iii) secret detectability, and (iv) robustness. However, most of chaotic steganography schemes have not been thoroughly examined for all requirements. The embedding efficiency as defined in [2, 20] is very important for a steganography scheme. It is conversely correlated with first three requirements and related to the performance of a steganography scheme. There are very few dedicated methods focusing on the embedding efficiency. Recently, the embedding efficiency has been improved by exploiting similarities between the secret and cover images as presented in [2], by using non-adaptive LSB technique [31], or by increasing amount of transmitted data [7].

Since the first work about chaotic encryption published by Robert Matthews [42], a huge number of chaos-based algorithms have been proposed for the image encryption with two main approaches, i.e. based on a pre-defined architecture and non-structural ones. For the first approach, a chaos-based cryptographic algorithm is designed by following one of certain well-known architectures such as Feistel and substitution-permutation network (SPN) [29, 41]. Ciphers employing the architecture of SPN can be implemented in the form of permutation and diffusion or substitution-diffusion for the chaos-based image encryption, and the security level is easily controlled by number of encryption rounds. In 1998, Fridrich proposed the permutation-diffusion architecture using chaotic maps [19] for the first time. Thereafter, the chaos-based image encryption has been developed extensively [10, 80, 82]. For chaotic ciphers using the architecture of Feistel, chaotic functions are involved in generating secret keys [11, 38], and in designing round functions [44, 51, 58, 62, 74, 89, 90]. For the second approach, a chaos-based cryptographic algorithm is not based on any specific architecture, but it is designed so that the security level is assured by means of complexity such as DNA encoding [22, 47, 76, 96, 100], transform domain [14, 26, 83], neural networks [52, 77, 87], quantum information processing [81, 101, 102] and their combinations [46, 94]. In any way, a cipher must have the confusion and diffusion properties as

suggestion given by C.E. Shannon [59]. Confusion is obtained by making a relation between the ciphertext and the secret key, and it can be obtained by substitution or permutation. Diffusion is achieved by spreading the dependence of the ciphertext itself and the secret key.

For a chaos-based image encryption employing the SPN, there are two main ways to use a chaotic system in order to achieve the confusion property: chaotic substitution box and chaotic permutation. For the chaotic substitution, a psuedo-random sequence generated by a chaotic system is used for constructing a substitution table (called a chaotic S-box) and values of plain pixels are substituted by values given in a chaotic S-box, e.g. [22, 104]. Based on the sensitivity on initial values of chaotic system, dynamical S-boxes were proposed and reported e.g. [53, 66, 103]. For the chaotic permutation, pixels are permuted one another under a rule generated by a chaotic system, i.e. value of a pixel or bits of a pixel at certain location is exchanged with that of another pixel at different location. In order to get the diffusion property, chaotic values can be used for manipulating values of pixels in different ways. In most of chaotic image encryption, chaotic values, values of plain and ciphered pixels are directly involved in equations to calculate new values of pixels, and it is performed sequentially to make the diffusion.

Recent algorithms of chaos-based image encryption were proposed with two significant innovations, i.e. the image-content sensitivity encryption and fast/efficient encryption algorithms. Firstly, for the image-content sensitivity encryption, the image content is sensitively involved in the encryption process. Hash values are generated by the image data and used as initial values for chaotic maps such as in [9, 88, 91], where the encryption key does not change during the encryption process. In [10, 21], values of parameters of chaotic maps are updated and dependent on values of pixels of intermediate ciphertext during encryption of a single image. As a result, the final ciphertext is much more sensitive to the image content. However, values of only parameters of chaotic maps are changed, while the state variables are not perturbed. The final ciphertext must be most sensitive to the plaintext if both state variables and control parameters of chaotic maps are perturbed after every iteration.

Secondly, fast and efficient encryption algorithms can be achieved by several ways, i.e. combination of permutation and diffusion [12, 80], multiple image encryption [22, 34, 40, 48, 49, 56, 63, 67, 78, 79, 95, 97–101], selective or partial encryption [6, 13, 32, 64, 86], algorithms using high dimensional chaos [70], or encryption dealing with blocks of data [1, 18]. Among them, the chaos-based multiple image encryption (MIE) has been most interested recently because that performs encrypting multiple images at the same time. In fact, the MIE is also very useful for the application of multi-party authentication that requires all ciphertexts for the successful decryption. However, there are some flaws in the existing schemes as presented in Section 2.1. The designs of existing algorithms of MIE are not in the form of SPN for taking advantage of controlling the security level by means of adjusting number of encryption rounds.

Nowadays, most of images are digital, therefore algorithms of chaos-based image encryption must be implemented in digital platforms. In any digital platform, values of state variables and control parameters of chaotic map must be represented by a limited number of bits. Consequently, chaotic dynamics is deteriorated by the rounding or truncating errors, and there exists a finite cycle [36] in the sequence of values generated by digital chaos. A chaotic map with poor dynamics must not be used for the chaotic encryption [3, 4]. So, this leads to recent proposals in enriching dynamics of digital chaos for the chaotic encryption in order to lengthen the cycle in the sequence of values generated by digital chaos. As described in [85], there are several approaches to reduce the deterioration of dynamics of chaotic maps, i.e. expansion of the precision, combination of multiple maps, and perturbation to a chaotic map. The feedback control method is considered as the perturbation

by an internal source. Under the perspective of hardware implementation, the approaches by expanding the precision in representing fractional numbers and by combining multiple chaotic maps require more hardware and computational resource than the approach of perturbation does. Under the view of perturbation, a perturbed chaotic map (PCM) with the source of perturbation coming from another chaotic map must be seen as combination of chaotic maps [8, 35, 39].

Theoretically, the perturbation on state variables makes chaotic dynamics stochastic, while that on control parameters pushes the dynamics of chaotic map into a non-stationary behavior. The most complicated dynamics is obtained with the perturbation on both state variables and control parameters. A chaotic map is perturbed in the form of feedback, that is, values of state variables are utilized to perturb on the control parameters, at the same time, values of control parameters are used to perturb back on the state variables as shown in [36, 68]. The perturbation can be a feedback in a way that a delay state variable is introduced to control parameters [37]. As presented in [16, 61, 68], state variables of a chaotic map are perturbed by external sources that makes the cycle length of the chaotic sequence extended.

In recent works by T.M. Hoang et al. [24, 25], the models of perturbed chaos at the bit level have been proposed for the chaos-based image encryption. The perturbation can be made to state variables, control parameters, or to both. Perturbation amounts come from either values of state variables (i.e. feedback) or external sources. In any scheme of perturbation, the value ranges of state variables and control parameters must be kept in the valid ones to ensure the exhibition of chaotic behavior. Chaotic values are represented in the format of fixed-point number. Statistical distributions of bits "0" and "1" are uniform for low significant bits. Such low significant bits are extracted and used for the encryption. Logically, perturbation makes chaotic dynamics more complicated, and if the PCM is used, the security of chaos-based encryption must be enhanced.

In fact, chaotic values of state variables consist of large number of bits. From chaotic bit sequences, high quality bit sequences in terms of randomness can be generated and those are tested and confirmed random enough to be used for the cryptographic application, e.g. [54, 75]. However, most of chaos-based encryption use chaotic values in computation with pixel values. Values of pixels are normally represented by either 8-bit integers for grayscale images or three times of 8-bit integers for RGB color images. For existing algorithms of chaos-based image encryption, many bits in values of state variables are not used for increase the efficiency of chaos-based encryption.

This paper presents a new design of chaos-based multiple image encryption with the architecture of permutation-diffusion network. Therein, a chaotic map is perturbed by the location information of pixels in the permutation, and another chaotic map is perturbed by values of plain and intermediate ciphered pixels in the diffusion. All perturbations to chaotic maps are made at the bit level after every iteration. Two main contributions are as follows:

- For the first time, the chaos-based multiple image encryption is designed with the structure of permutation-diffusion network, in which PCMs are employed. With the proposed design, any chaotic map can be used in the design of encryption. Bits of values of state variables produced by PCMs are thoroughly exploited for the MIE.
- In both the permutation and diffusion processes are designed such that dynamics of chaotic maps is perturbed after every iteration. The source for amounts of perturbation is switched between the image content and values of state variables. Therefore, the encryption possesses the property of authentication. With the perturbation, the dynamics of chaotic maps is more complicated, and the dynamical degradation is reduced. As a result, the merit of perturbed chaos enhances the security logically.

The example is demonstrated and the simulation result shows the effectiveness of the proposed design of MIE.

The rest of the paper is organized as follows. In Section 2, the related works about the existing MIEs and the PCM are reviewed. Then, the detail about the proposed design of MIE using PCMs is presented in Section 3. Section 4 shows the exemplar simulation and detailed results of analysis. Finally, Section 5 gives the discussions and conclusions of the work.

2 Related works

In this section, the existing chaos-based multiple image encryption (MIE) is reviewed and then the drawbacks of existing algorithms of MIE are pointed out. Recent models of perturbed chaotic maps are also presented.

2.1 Multiple image encryption

As presented in [97], the big image is obtained by merging multiple smaller images, $k = K_1 * K_2$ images and each image sized $M \times N$. The permutation is carried out at blocks of pixels with the size $M_1 \times M_1$ (e.g. 8×8 pixels) with the rule generated by sorting a chaotic sequence. The scrambled blocks of pixels are merged to become a big image and then the big image is segmented into images with the original size. It is a lack of diffusion effect in the encryption. A similar method of multiple image encryption is presented in [98] by the same group of authors, in which the diffusion process is inserted into the encryption by XORing between scrambled images and the same chaotic matrix.

Inspired by the magic cube game, a 3D permutation model using the rule generated by chaotic sequence was proposed and presented in [99] to encrypt k images with the same size. There, the permutation is carried out with the internal-row, internal-column modes for rows and columns of pixels in individual images, and then with the external mode for all plain images. The XOR operation is used as the diffusion between permuted images and the same chaotic image. A similar idea to this work is presented in [56]. A limited number of original images are arranged into the form of cube image with the size $128 \times 128 \times 128$ for the pixel permutation in three directions. The XOR operation is performed between the scrambled cubic image and the cubic chaotic image. The resultant cubic image after XORed is rearranged to produce the encrypted images.

As presented in [100], the DNA computing is employed in the design. Multiple images with the same size are merged to become a big image for scrambling. Then, the big scrambled image is segmented into multiple smaller images for diffusion. The DNA operation is used for encoding. The DNA XOR operation between matrices does not make domino effect in the diffusion and all the scrambled images are masked with the same chaotic image.

In [95], the multiple image encryption deals with significant bit planes in order to reduce the volume of plaintext to be encrypted. The XOR operation between the scrambled images and the same chaotic image is performed to produce the encrypted images. In that work, four most significant bits of plain images are extracted, and bits in each plane are permuted.

Then, the scrambled bit planes are combined with four least significant bits by another permutation. Lastly, the scrambled images are XORed with the chaotic image to produce the ciphertexts.

In [67], the multiple image encryption is presented, in which plain images are decomposed into bit planes and from that bit blocks are formed. Bit blocks are swapped and then XORed with the chaotic matrix. Four plain images with the same size are demonstrated. The Henon map is employed for determining pattern of bit blocks and the Logistic map is used for generating the chaotic matrix. The disadvantage is that the random size of bit blocks makes the duration for the encryption uncertain.

The novel quantum multi-image encryption was proposed by Nanrun Zhou et al. as presented in [101]. There, the quantum representation model for multiple images and the quantum multi-image encryption scheme were devised by combining the quantum 3D Arnold transform and quantum XOR operations with the scaled Zhongtang chaotic system. There are two stages of diffusion and permutation for single round of encryption. However, a large number of computational operation are required with a series of summation and multiplication for quantum transformations.

In [49], K Abhimanyu Kumar Patro et al. proposed the multiple image encryption with various sizes, and that operates the permutation and diffusion on m blocks of pixels with the size of 2×2 and rem_size remaining pixels. Then, the shuffled blocks and pixels are XORed sequentially with chaotic values, and finally the combination is performed to produce the ciphertexts.

In the similar idea in [49] by the same group of authors, the multiple RGB-image encryption is presented in [48], but the difference is that the plain images are not divided into blocks. k plain RGB images with the same size are considered, and three color channels of each image are separately encrypted. The k components of the same color are concatenated horizontally for pixel permutation. Next, combinations of pairs of color images are concatenated vertically for pixel-row permutation and then horizontally for pixel-column permutation. Finally, the diffusion is performed by sequential XORing operation hopping over shuffled components of different colors. Multiple grayscale images are also encrypted in a similar way as presented in [50].

In [40], multiple RGB images with the equal size are merged to become a big image, and then pixels in the images of three color channels are scrambled using three sorted chaotic sequences generated by the 3D Lorenz chaotic system. The XOR operation is performed between the scrambled image and the chaotic image to produce the encrypted images.

In [22] the method of multiple image encryption using the DNA encoding is presented. Multiple RGB images are merged and the images of color channels are encrypted separately and parallel. Several S-boxes are generated using chaotic sequences. The permutation rules are generated by chaotic sequences, then the XOR operation is performed to produce the images of color channels before merging into encrypted RGB images.

In 2019, M. Zarebnia et. al. reported in [93] about the algorithm of fast multiple-image encryption for grayscale images. Each image are divided into four equal non-overlapped sub-images. Sub-images are permuted within an image and then a big image is obtained by combining all images together. Such the sub-blocks are relocated, and then pixels of big image are permuted. Bits in pixels of the permuted big image are XORed with bits of chaotic values, and after that the column values of resultant big image is cyclic shifted. The XOR operation and the cyclic shift are carried out two times to create the diffusion in the encrypted images. However, it is noted that chaotic values for the XOR operation and the

rules of sub-images, and pixels permutation are produced by chaotic maps. It is clear that the diffusion is obtained by the cyclic shift of pixels combining with the XOR operation.

In conclusion, there are three drawbacks in the existing methods of multiple image encryption. First, all of the above-mentioned methods of MIE are designed to work for a single round of encryption. The security level in terms of statistical analysis of ciphertext can not be changeable by changing number of encryption rounds. Second, most of above researches for the MIE use the XOR operation for diffusion. In fact, the XOR operation in [40, 56, 67, 95, 98–100] does not bring the mean of diffusion or an avalanche effect in pixel values does not exist. It is simply considered as a masking process between the scrambled images and chaotic images. However, the truly mean of diffusion property with the domino fashion only exists in [48–50]. Third, in order to resist from the known-plaintext and chosen-plaintext attacks, some of encryption algorithms use hash values from plain images (e.g. SHA-3 in [92], SHA-256 in [48–50, 95, 98–100], or SHA-512 [56]), and those values are used as initial values for chaotic systems or for other processes but those are kept constant throughout encryption. This means that the dynamics of chaotic maps in those algorithms is stationary with fixed values of initial conditions and parameters. The security of chaos-based encryption must be significantly enhanced if the dynamics of chaotic maps is non-stationary.

These drawbacks in the existing algorithms of MIE are overcome in our proposed design which is based on the architecture of permutation-diffusion network using perturbed chaotic maps.

2.2 Perturbed chaotic map

2.2.1 Bit arrangement

In this research, there are two functions operating at the bit level, i.e. bit manipulation and bit arrangement. These operations are to construct models of perturbed chaotic map as reported in [25].

The bit arrangement consists of two activities, i.e., bit extraction and bit permutation. Let us consider matrices of bits A and B with the sizes $I_A \times J_A$ and $I_B \times J_B$, or $A = [a_{ij}]_{1\le i\le I_A,\ 1\le j\le J_A}$ and $B = [b_{ij}]_{1\le i\le I_B,\ 1\le j\le J_B}$, with $a_{ij}, b_{ij} \in \{0, 1\}$. The matrix B is constructed from bits of the matrix A by a rule defined by Y (denoted by $\circ$) as a bit arrangement operator, that is

$$A = Y \circ B. \tag{1}$$

where Y is the rule defining the way that bits of matrix A being extracted for the matrix B. Specifically, the rule of bit extraction is an array of 2-tuples as $Y = [(y_{ij}^{(row)}, y_{ij}^{(col)})]_{1\le i\le I_A,\ 1\le j\le J_A}$, and the value ranges are $y_{ij}^{(row)} \in [1, I_B]$ and $y_{ij}^{(col)} \in [1, J_B]$. A bit of A is taken from that of B as $a_{ij} = b_{y_{ij}^{(row)}, y_{ij}^{(col)}}$. In case that a bit of A is deliberately fixed at a status '0' or '1', then the 2-tuple $(y_{ij}^{(row)}, y_{ij}^{(col)})$ is expressed by B_0 or B_1 for bit '0' or '1', respectively.

In this work, the matrix B is obtained from an array of bit sequences, in which each bit sequence represents for values of a chaotic state variable. The matrix A is used for constructing an array of I_A bit sequences for perturbation. Each bit sequence is collection of bits in the same row of A, i.e. $A_i = ||_{j=1}^{J_A} a_{ij}$, where $||$ denotes for the concatenation of bits.

For example, a positive value is represented in the format of fixed-point number with eighteen bits as a whole and sixteen bits from the least significant position for the fractional part (denoted as $\langle 18, 16\rangle$). Thus, the value of $X = (0.135, 2.634)$ in binary is

$$X_{bin} = \begin{pmatrix} 00.0010001010001111 \\ 10.1010001001001110 \end{pmatrix}. \tag{2}$$

Matrix B is obtained from X_{bin} is

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{3}$$

If the rule of extraction is

$$Y = \begin{bmatrix} (1, 14), (2, 8), (1, 6), (2, 3) \\ (2, 4), (2, 9), (1, 7), (1, 13) \end{bmatrix}, \tag{4}$$

and the bit arrangement as given in (1), then matrix A is

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5}$$

Arrays of bit sequences are achieved by concatenating bits in rows of A is $A_1 = 0001$ and $A_2 = 0100$. Figure 1 illustrates the bit arrangement. In fact, any bit of B can be used multiple times to construct A.

Let us consider the number of distinct rules Y constructed from a certain size of B. Assumed that the number of bits in B chosen to construct Y is $L_B = I'_B * J'_B$ and the number of 2-tuples $(y_{ij}^{(row)}, y_{ij}^{(col)})$ in Y is $L_Y = I_A * J_A$. It is noted that if some of bits of B are used to construct A, so $I'_B \leq I_B$ and $J'_B \leq J_B$. Also, if any bit from B is not repetitively chosen for A, then the number of distinct variants of Y is considered as L_Y-permutations of L_B or

$$N_Y = \frac{L_B!}{(L_B - L_Y)!}. \tag{6}$$

In case that a bit from B is chosen multiple times for constructing A, N_Y is much larger and dependent on how many times a bit of B is allowed being chosen. However, because the number of bits in A is $I_A * J_A$, so the number of distinct bit patterns of A is $2^{I_A * J_A}$. The bit arrangement will be used for bit extraction for perturbation in the PCM.

2.2.2 Bit manipulation

The function of bit manipulation is to lengthen or shorten a bit sequence dependent on the relative length of the input and the output. Note that $|X|$ returns the number of bits or the

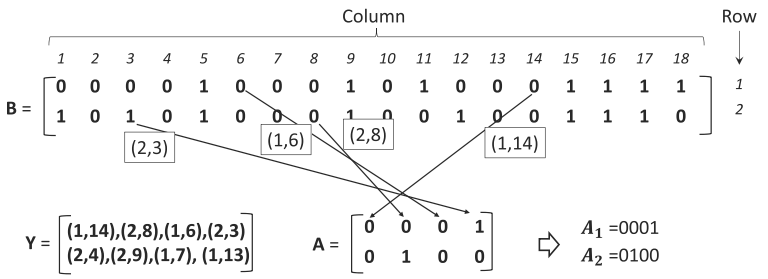


Fig. 1 Example of bit arrangement

length of bit sequence X. Let us consider a bit manipulation of two bit sequences (E_1 and E_2 with the lengths $|E_1|$ and $|E_2|$, respectively) and produce bit sequence E with the length $|E|$. There are two cases of relative lengths, i.e. $|E| < |E_1| + |E_2|$ and $|E| \geq |E_1| + |E_2|$. In this work, two simplest ways to deal with two cases of lengths to get E.

BM-1: For $|E| \geq |E_1| + |E_2|$, the bit sequence E is simply constructed by $E = E_1||E_2$.

BM-2: For the second case $|E| < |E_1| + |E_2|$

Step 1: Find an integer n such that $(n-1) * |E| < |E_1| + |E_2| < n * |E|$.

Step 2: Separate the sequence E into n portions, i.e. T_i $i = \{1...n\}$, such that first $n-1$ portions are with the same length $|E|$ and the last portion T_n is with the length $|T_n| < |E|$.

Step 3: Pad the sequence of bit zeros '0...0' with the length of $|E| - |T_n|$ into the sequence T_n to become $T_n||0...0$.

Step 4: XOR n sequences of bits T_i to have the output sequence E.

The step 4 of BM-2 is shown in Fig. 2. $\triangleright$ and $\triangleleft$ are chosen to denote for the operator of BM-1 and BM-2 in the later equations. In fact, more complicated bitwise operations can be applied to generate the output bit sequence.

2.2.3 Description of perturbed chaotic map

Let us consider a chaotic map realized in a digital platform and its state variables and parameters are perturbed at the bit level as illustrated in Fig. 3. The function of PCM $F(.)$ is expressed by

$$\begin{cases} X_{n+1} &= F(\hat{X}_n, \hat{\Gamma}_n), \\ \hat{X}_n &= X_n \oplus \Delta_{X_n}, \\ \hat{\Gamma}_n &= \Gamma_n \oplus \Delta_{\Gamma_n}. \end{cases} \tag{7}$$

where the vectors are of state variables X_n and control parameters Γ_n; and the perturbation vectors at a certain iteration are Δ_{X_n} and Δ_{Γ_n}. Assumed that the chaotic map has D dimensions and G control parameters, so $X_n = [x_n^{(D)}\ x_n^{(D-1)} ...\ x_n^{(2)}\ x_n^{(1)}]^T$, $\Gamma_n = [\gamma_n^{(G)}\ \gamma_n^{(G-1)} ...\ \gamma_n^{(2)}\ \gamma_n^{(1)}]^T$, $\Delta_{X_n} = [\delta_{x_n^{(D)}}\ \delta_{x_n^{(D-1)}} ...\ \delta_{x_n^{(2)}}\ \delta_{x_n^{(1)}}]^T$ and $\Delta_{\Gamma_n} =$

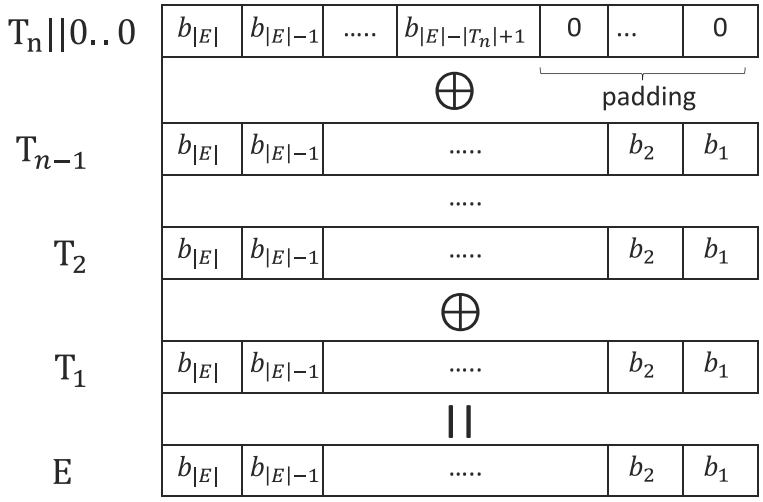


Fig. 2 The bit manipulation

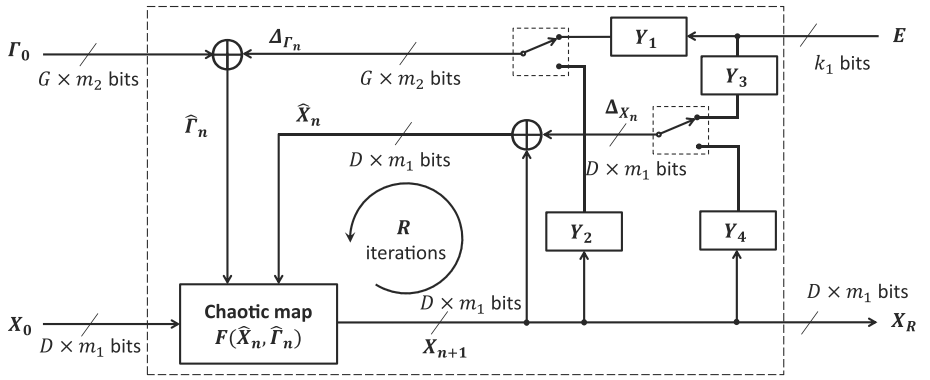


Fig. 3 The perturbed chaotic map (PCM)

$[\delta_{\gamma_n^{(G)}}\ \delta_{\gamma_n^{(G-1)}} \ldots \delta_{\gamma_n^{(2)}}\ \delta_{\gamma_n^{(1)}}]^T$. Amounts of perturbation to the chaotic map can be either from external source E or internal source X_n decided by the switches as

$$\Delta_{\Gamma_n} = \begin{cases} Y_1 \circ E & \text{for } n = 0, \\ Y_2 \circ X_n & \text{for } 1 \le n \le R, \end{cases} \tag{8}$$

and,

$$\Delta_{X_n} = \begin{cases} Y_3 \circ E & \text{for } n = 0, \\ Y_4 \circ X_n & \text{for } 1 \le n \le R. \end{cases} \tag{9}$$

The bit arrangements Y_i are described in Section 2.2.1. Assumed that values of E are represented by k_1 bits, and values of state variables and control parameters are represented by $D \times m_1$ and $G \times m_2$ bits, respectively. In addition, values of state variables and control parameters are represented in the format of either integer or fixed-point number dependent on the definition of original chaotic map. In the case of fixed-point number, m_1 bits of state variables are composed by the most significant bit for the sign, $m_1^{(int)}$ bits for integer portion, and $m_1^{(frac)}$ bits for fractional portion, i.e. $m_1 = m_1^{(int)} + m_1^{(frac)} + 1$. That is similar for the control parameters as $m_2 = m_2^{(int)} + m_2^{(frac)} + 1$. In practical, m_1 and m_2 are adopted for the desired precision to avoid the degradation of chaotic dynamics and for the desired space of secret key.

To ensure that chaotic dynamics is retained without disrupting the model of chaotic map, values of perturbed state variables as well as perturbed control parameters, $\hat{X}_n$ and $\hat{\Gamma}_n$ in (7), must be constrained in its valid ranges as defined by the original chaotic dynamics. Therefore, a number of selective bits at specific positions in X_n and Γ_n may be kept its states fixed, while states of other bits can be flippable by the perturbation. In fact, the state variables and control parameters can be interpreted by arithmetic operations as

$$\begin{aligned} \hat{X}_n &= X_n \pm \Theta_{X_n}, \\ \hat{\Gamma}_n &= \Gamma_n \pm \Theta_{\Gamma_n}, \end{aligned} \tag{10}$$

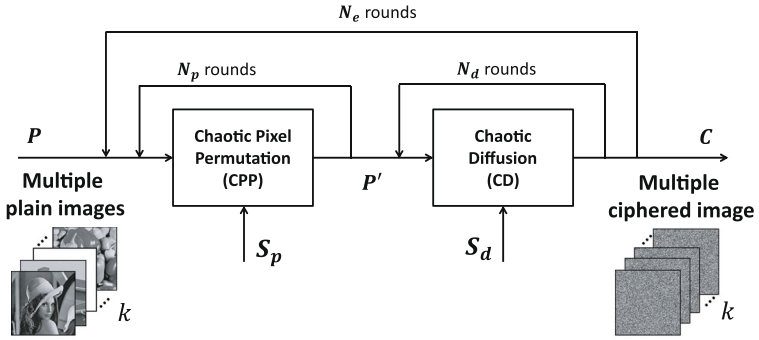
where Θ_{X_n} and Θ_{Γ_n} are seen as amounts of perturbations and those are considered as vectors of instant noise interfering the original orbit of chaotic map, i.e. $\Theta_{X_n} = [\theta_{x_n^{(D)}}\ \theta_{x_n^{(D-1)}} \ldots \theta_{x_n^{(2)}}\ \theta_{x_n^{(1)}}]^T$ and $\Theta_{\Gamma_n} = [\theta_{\gamma_n^{(G)}}\ \theta_{\gamma_n^{(G-1)}} \ldots \theta_{\gamma_n^{(2)}}\ \theta_{\gamma_n^{(1)}}]^T$. The sign "+" or "–" in (10) is dependent on the relative difference between values of X_n and Γ_n before and after

perturbed. Values of $\theta_{x_n^{(i)}}$ and $\theta_{\gamma_n^{(i)}}$ are dependent on the status of the most significant bit being changed before and after perturbation in $x_n^{(i)}$ and $\gamma_n^{(i)}$, respectively.

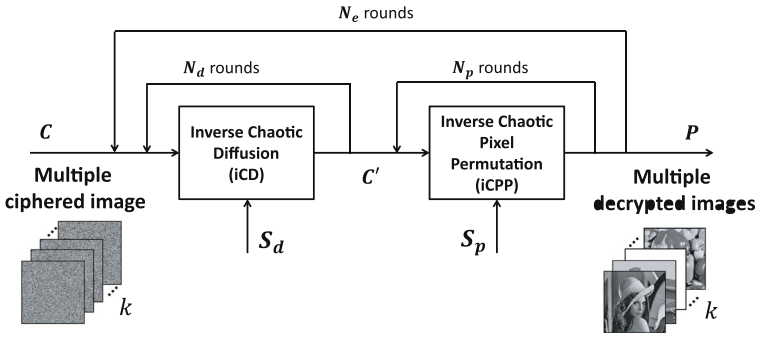
According to the statistical analysis [25], the distribution of bits at low-weight positions in chaotic values is uniform. Therefore, low-weight bits in values of X_n should be chosen to perturb high-weight bits in the feedback, or those should be used for making up amounts of perturbation Δ_{X_n} and Δ_{Γ_n}.

3 Proposed design of multiple image encryption based on perturbed chaos

Figure 4 illustrates the proposed design of MIE for k images. Figure 4a presents the encryption with two stages, i.e. chaotic pixel permutation (CPP) and chaotic diffusion (CD). Respectively, P, P' and C are sets of k plain images, permuted images and ciphered images. The CPP and CD are performed N_p and N_d rounds, respectively. Both the CPP and CD are iterated N_e rounds. The secret key consists of S_p and S_d. The decryption as displayed in Fig. 4b consists of the inverse chaotic pixel permutation (iCPP) and inverse chaotic diffusion (iCD). The order of the permutation and the diffusion is reversed in compared with that in the encryption. Here, the PCM presented in Fig. 3 is employed for the permutation and diffusion.



(a) Encryption



(b) Decryption

Fig. 4 The abstract design of multiple image cryptosystem

3.1 Chaotic pixel permutation for MIE

It is known that the pixel permutation is to exchange values between a pair of pixels within the spatial range of image. Here, the coordinate of destination pixels is produced by the chaotic map in the CPP.

Figure 5 displays the structure of CPP for the MIE, in which the PCM produces the destination coordinates for pixels. There, $\Gamma_0^{(p)}$ and $X_0^{(p)}$ are vectors of initial values. It is assumed that all k plain images are with the same size of $M \times N$; $M = 2^{k_1^{(x)}}$ and $N = 2^{k_1^{(y)}}$.

In the case of multiple image permutation, there are two possible schemes as shown in Fig. 6, i.e. intra-image permutation and inter-image permutation. A pixel of an image can be shuffled with another pixel either of the same image as the scheme of intra-image permutation in Fig. 6a or of another image as the scheme of inter-image permutation in Fig. 6b. Therefore, the single coordinate of k original pixels is represented by bit sequence XY_{orig} as the input of the CPP for perturbation. For simplicity, the bit sequence XY_{orig} is obtained by concatenating bit sequences what represents for row number X and column number Y, i.e. $XY_{orig} = X||Y$ as shown in Fig. 7a.

The bit sequence XY_{orig} with the length $k_1^{(p)} = k_1^{(x)} + k_1^{(y)}$ at the input is converted to the bit sequence E_p with the length $k_1^{(p)}$ by the block Bit Manipulation using the function BM_p. The bit sequence E_p is used for perturbing the PCM. After R_p iterations, chaotic values of state variables of PCM, X_{R_p}, are produced and utilized for figuring out destination coordinates kXY_{des} for k original pixels corresponding to the same original coordinate XY_{orig} of k images. kXY_{des} is achieved by

$$kXY_{des} = Y_p \circ X_{R_p}. \tag{11}$$

where Y_p is the function of Bit arrangement that defined similarly to Y_i in the PCM.

For the scheme of inter-image permutation, the array of bit sequences kXY_{des} representing for the destination coordinates of pixels is given in Fig. 7b. There, the destination coordinate for an original pixel at XY_{orig} of image P_i is the i-th row of kXY_{des}. Each row of kXY_{des} is a bit sequence that comprises of the destination coordinate represented by $k_1^{(x)}$ and $k_1^{(y)}$ bits and the index of destination image encoded by $Log_2 k$ bits. Therefore, the array of bit sequences for the destination coordinate kXY_{des} has $k * (k_1^{(x)} + k_1^{(y)} + Log_2 k)$ bits. However, for the scheme of intra-image permutation, the bit portion for the destination image is not necessary.

It is noted that the CPP is carried out for pixels in the images from left to right and row by row. The structure of inverse chaotic permutation (iCPP) in the decryption is exactly

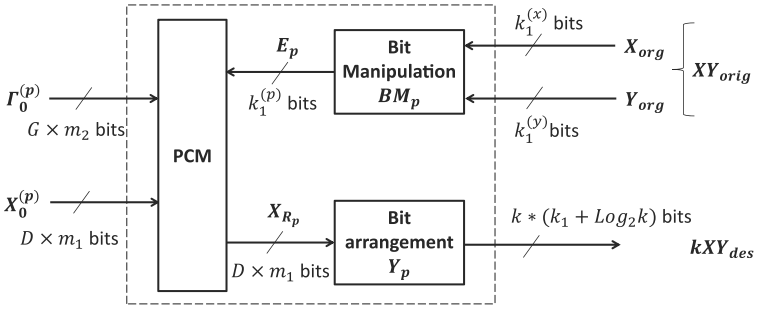
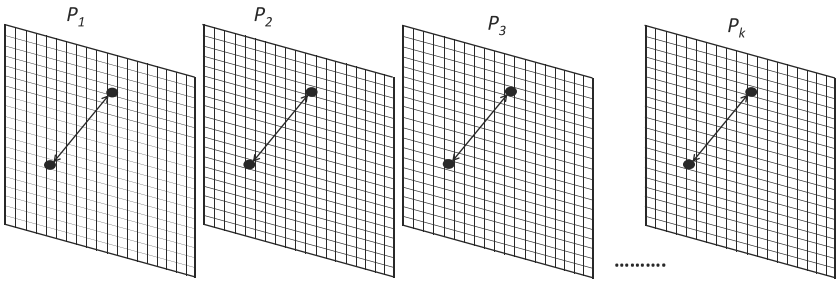
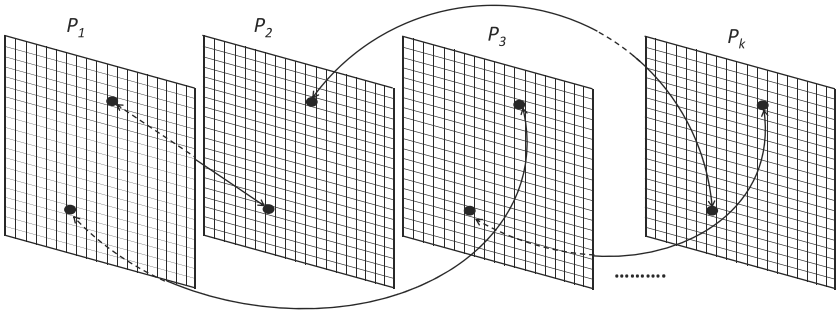


Fig. 5 The proposed structure of chaotic pixel permutation (CPP) for the multiple image encryption

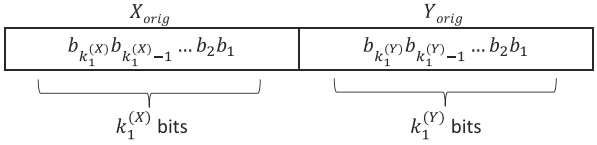


(a) Intra-image pixel permutation.



(b) Inter-image pixel permutation.

Fig. 6 Two options of pixel permutation



(a) Bit sequence of XY_{orig}

P_{orig}	X_{des}	Y_{des}	P_{des}
P_1	$b_{k_1^{(X)}} b_{k_1^{(X)}-1} \ldots b_2 b_1$	$b_{k_1^{(Y)}} b_{k_1^{(Y)}-1} \ldots b_2 b_1$	$b_{\log_2 k} b_{\log_2 k-1} \ldots b_2 b_1$
P_2	...	...	...
..	⋮	⋮	⋮
P_{k-1}	...	...	...
P_k	$b_{k_1^{(X)}} b_{k_1^{(X)}-1} \ldots b_2 b_1$	$b_{k_1^{(Y)}} b_{k_1^{(Y)}-1} \ldots b_2 b_1$	$b_{\log_2 k} b_{\log_2 k-1} \ldots b_2 b_1$
	$k_1^{(Y)}$ bits	$k_1^{(Y)}$ bits	$k_1^{(img)} = Log_2 k$ bits

(b) Bit sequence of kXY_{des}

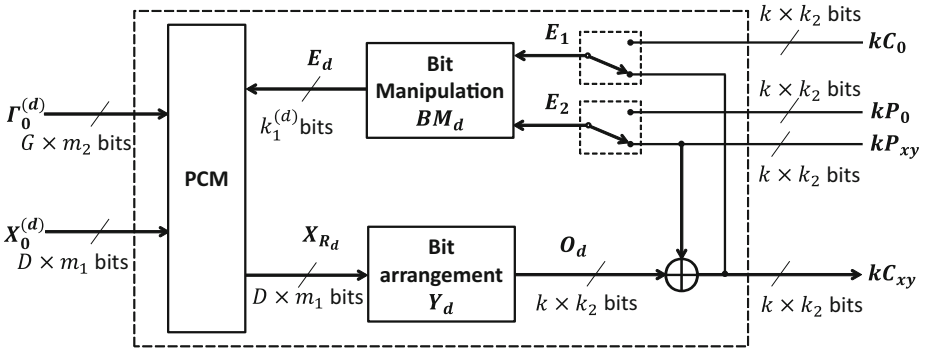
Fig. 7 Array of bit sequences encodes the original and destination pixels in the permutation

the same as that of the CPP, except that the scanning order of pixels is in reverse way in compared with that in the CPP. The block Bit Manipulation with the function BM_p manipulates its input XY_{orig} to become E_p for the perturbation to the PCM. In case that $Log_2(M*N) \geq k_1^{(p)}$, BM_p shortens $Log_2(M*N)$ bits into $k_1^{(p)}$ bits. However, if $Log_2(M*N) < k_1^{(p)}$, the bit lengthening operation can be carried out by Y_1 and Y_2 in the PCM. In any case, the bitwise operations can be optionally used by BM_p.

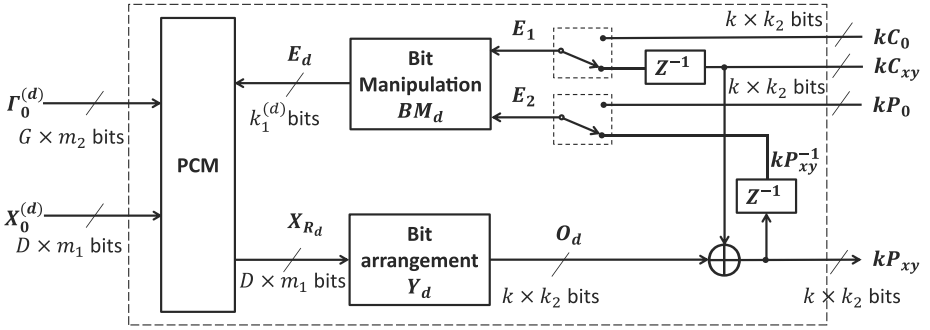
Let us consider the contribution of the permutation to the secret key of cryptosystem. Due that the PCM is perturbed by the coordinate information of pixels, only flippable bits in values of state variables and control parameters are counted for the secret key. The number of bits that contributes to the secret key is $S_p = |\Gamma_n^{(p)}|_{flip} + |X_n^{(p)}|_{flip}$; where $|.|_{flip}$ returns the number of flippable bits.

3.2 Chaotic diffusion for MIE

Figure 8 illustrates the proposed structure of chaotic diffusion (CD) and inverse chaotic diffusion (iCD) using the PCM for the MIE. In both of the CD and iCD, initial values for the PCM is $X_0^{(d)}$ and $\Gamma_0^{(d)}$, and initial values of plain and ciphered pixels are kP_0 and kC_0. However, $2 * k * k_2$ bits of E_1 and E_2 are transformed to $k_1^{(d)}$ by the function BM_d. The value of each pixels is represented by k_2 bits. In the CD and iCD, the scanning order for pixels is from left to right, row by row, and top to bottom.



(a) Chaotic diffusion (CD)



(b) Inverse chaotic diffusion (iCD)

Fig. 8 The proposed structure of (CD) and iCD using the PCM

The CD in Fig. 8a shows that the array of bit sequences kP_{xy} is with the size $k * k_2$ bits representing for values of k pixels from k images. With the perturbation E_d, the PCM iterates R_d times to produce X_{R_d}. The array of bit sequences kC_{xy} that represents for values of ciphered pixels of k images is extracted from X_{R_d}. The equations describe the operation of the CD as follows

$$E_1 = \begin{cases} kC_0 & \text{for } (x, y) = (1, 1); \\ kC_{xy} & \text{for } (x, y) \neq (1, 1), \end{cases} \tag{12}$$

$$E_2 = \begin{cases} kP_0 & \text{for } (x, y) = (1, 1); \\ kP_{xy} & \text{for } (x, y) \neq (1, 1), \end{cases} \tag{13}$$

and

$$\begin{aligned} kC_{xy} &= kP_{xy} \oplus O_d, \\ &= kP_{xy} \oplus (Y_d \circ X_{R_d}). \end{aligned} \tag{14}$$

For the iCD in Fig. 8b, the process to recover plain pixels is almost identical to that of the CD, except that the block Z^{-1} is to delay the data to match with the operation in the CD. The equations for the operation of the iCD are

$$E_1 = \begin{cases} kC_0 & \text{for } (x, y) = (1, 1); \\ kC_{xy}^{-1} & \text{for } (x, y) \neq (1, 1), \end{cases} \tag{15}$$

$$E_2 = \begin{cases} kP_0 & \text{for } (x, y) = (1, 1); \\ kP_{xy}^{-1} & \text{for } (x, y) \neq (1, 1), \end{cases} \tag{16}$$

and

$$\begin{aligned} kP_{xy} &= kC_{xy} \oplus O_d, \\ &= kC_{xy} \oplus (Y_d \circ X_{R_d}). \end{aligned} \tag{17}$$

It is noted that values of plain pixels are utilized not only for calculating the cipher pixels in the CD and for recovering the plain pixels in the iCD, but also perturbing to the PCM. This makes the diffusion double dependent on the plain pixels.

The block Bit Manipulation in the CD is identical to that in the iCD, and its function BM_d is generally to manipulate the inputs E_1 and E_2 with the length $2 * k * k_2$ bits to become $k_1^{(d)}$-bit E_d for the PCM.

The number of bits that the diffusion contributes to the secret key as $S_d = |\Gamma_n^{(d)}|_{flip} + |X_n^{(d)}|_{flip} + |kP_0| + |kC_0|$. Therefore, the total number of bits of the secret key in the proposed design is $S = S_p + S_d$. Next, the specific example will demonstrate the performance of the proposed design.

4 Exemplar simulation

Let us consider the example using the perturbed Standard map for both the permutation and diffusion with the equations as

$$\begin{cases} x_{n+1}^{(1)} &= (\hat{x}_n^{(1)} + \hat{x}_n^{(2)}) \; mod \; 2\pi, \\ x_{n+1}^{(2)} &= (\hat{x}_n^{(2)} + \hat{\gamma}_n^{(1)} sin(\hat{x}_n^{(1)} + \hat{x}_n^{(2)})) \; mod \; 2\pi, \\ \hat{x}_n^{(1)} &= x_n^{(1)} \oplus \Delta x_n^{(1)}, \\ \hat{x}_n^{(2)} &= x_n^{(2)} \oplus \Delta x_n^{(2)}, \\ \hat{\gamma}_n^{(1)} &= \gamma_n^{(1)} \oplus \Delta \gamma_n^{(1)}; \end{cases} \tag{18}$$

where the vectors of state variables $X_n = [x_n^{(2)} \;\; x_n^{(1)}]^T$ and control parameter $\Gamma_n = [\gamma_n^{(1)}]$ are respectively perturbed by $\Delta \hat{X}_n = [\Delta \hat{x}_n^{(2)} \;\; \Delta \hat{x}_n^{(1)}]^T$ and $\Delta \hat{\Gamma}_n = [\Delta \hat{\gamma}_n^{(1)}]$. The PCM is

with D=2 and G=1. The value range of $x_n^{(1)}$ and $x_n^{(2)}$ is [0,2π] as in the original Standard map. Let us choose the value range for the control parameter of the chaotic map as $1.0 \leq \gamma_n^{(1)} < 8.0$. The values of state variables and control parameters are represented in the format of fixed-point number with 32 bits for fractional portions. The chosen bit patterns and the corresponding value ranges are given in Table 1. The bits with the state 'x' are flippable by the perturbation while other ones are fixed with the state '1'.

The simulation is carried out with a set of eight plain images, i.e. Lena, Cameraman, House, Boat, Clock, Black and White. All of the images are with the same size of 256×256 and with the 8-bit grayscale, thus kP_n and kC_n are the array of 8-bit numbers. According to the size of chosen images, the values of parameters in the CPP and CD are adopted as $k_1^{(x)} = 8$, $k_1^{(y)} = 8$ (or $k_1^{(p)} = 16$), $k = 8$ and $k_2 = 8$. In the following text, the notations with the superscripts (p) and (d) are to indicate the state variables and parameters of the PCMs in the permutation and diffusion, respectively.

As shown in Table 1 and Fig. 5, the PCM in the CPP has 102 bits at maximum what can be perturbed by 16 bits, i.e. X and Y, representing for row and column numbers as illustrated in Fig. 7a. Therefore, BM-1 as described in Section 2.2.2 is chosen for BM_p to produce $E_p = X||Y$. Respectively, $Y_1^{(p)}$, $Y_2^{(p)}$, $Y_3^{(p)}$ and $Y_4^{(p)}$ are equivalent Y_1, Y_2, Y_3 and Y_4 in the PCM of the CPP, and those are chosen as shown in Table 2. It is noted that B_0 is denoted for the bit at that position in state variables and control parameters not being perturbed. The CPP gives $S_p = 102$ bits to the secret key.

Similarly, the number of flippable bits in the PCM of the CD is 102 bits, and the number of bits in initial values of kP_0 and kC_0 is 128. Therefore, the number of bits that the diffusion contributes to the secret key is $S_d = 230$ bits. The total number of bits of the secret key is $S = S_p + S_d = 102 + 230 = 332$ bits.

Table 3 shows the rules that bits are extracted for kXY_{des}, kC_{xy} and kP_{xy} as explained in Section 3. Y_p shows that eight sequences of bits kXY_{des} are extracted from X_{R_p} as illustrated in Fig. 7b. Each sequence consists of 16 bits representing for the destination coordinate X_{des} and Y_{des}, and 3 bits indicating the index of destination image P_{des} for the scheme of inter-image permutation as shown in Fig. 6b. Similarly, Y_d displays that eight sequences of bits O_d are obtained from X_{R_d}, and each sequence is of 8 bits representing for a chaotic value used in (14) to produce a diffused pixel.

The initial values of the cipher are chosen for the simulation as shown in Table 4. Due that eight images are encrypted at the same time or $k = 8$, the initial values of eight plain pixels and eight cipher ones for the diffusion are also given.

4.1 Simulation results

The set of test images consists of six natural images (Lena, Cameraman, House, Peppers, Boat and Clock) and two special ones (Black and White) as shown in the first column of Fig. 9. The simulation to verify the effectiveness of the proposed design is carried out in two phases.

In the first phase, the simulation is performed separately for the permutation and diffusion with respect to various number of iterations (R) applied to the PCMs for every pixel and with respect to various number of permutation and diffusion rounds for images (N_p and N_d). From the intensity analysis of permuted and diffused images, suitable values of R, N_p and N_d are adopted for the simulation of encryption.

In the second phase, the encryption with the structure of the permutation and diffusion as shown in Fig. 4a is simulated with various number of encryption rounds (N_e) and the result will be presented.

Table 1 The format of fixed-point number, bit patterns, value ranges and the length of the secret key in the ciphers

Stage	Elements	Format	Bit pattern	Value range	No. bits for the secret key	Total
Permutation	$x_n^{(1)}$	$\langle 35, 32\rangle$	xxx.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	$[0,2\pi]$	35	102
	$x_n^{(2)}$	$\langle 35, 32\rangle$	xxx.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	$[0,2\pi]$	35	
	$\gamma_n^{(1)}$	$\langle 35, 32\rangle$	xx1.1x1xxxxxxxxxxxxxxxxxxxxxxxxxxxxx	[1.625,1.75)		
				[1.875,2.0)		
				[3.875,4.0)		
				[7.875,8.0)	32	
Diffusion	$x_n^{(1)}$	$\langle 35, 32\rangle$	xxx.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	$[0,2\pi]$	35	230
	$x_n^{(2)}$	$\langle 35, 32\rangle$	xxx.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	$[0,2\pi]$	35	
	$\gamma_n^{(1)}$	$\langle 35, 32\rangle$	x1x.1x1xxxxxxxxxxxxxxxxxxxxxxxxxxxxx	[2.625,2.75)		
				[2.875,3.0)		
				[3.875,4.0)		
				[7.875,8.0)	32	
	E_1	$8 \times \langle 8, 0\rangle$	xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx	[0,255]	64	
	E_2	$8 \times \langle 8, 0\rangle$	xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx			
			xxxxxxxx	[0,255]	64	

Table 2 Bit arrangements Y_i in the perturbed Standard map

$Y_1^{(p)}$	$[(1,4)(1,6)B_0B_0(1,12)B_0(1,9)(1,11)(1,3)(1,2)(1,15)(1,14)(1,16)(1,1)(1,5)(1,7)(1,10)(1,8)(1,13)$ $B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0]$
$Y_2^{(p)}$	$[(2,34)(2,31)B_0B_0(1,25)B_0(1,30)(2,23)(1,35)(1,33)(1,29)(2,21)(1,24)(2,19)(1,28)(2,32)(2,27)(1,22)(2,20)$ $(1,18)(2,26)B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0]$
$Y_3^{(p)}$	$\begin{bmatrix} (1,3)(1,1)(1,4)(1,12)(1,16)(1,7)(1,9)(1,15)(1,14)(1,10)(1,13)(1,8)(1,5)(1,6)(1,2)(1,11)B_0B_0B_0 \\ B_0B_0B_0(1,8)(1,7)(1,4)(1,2)(1,6)(1,13)(1,14)(1,12)(1,1)(1,15)(1,16)(1,10)(1,3)(1,9)(1,11)(1,5) \\ B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0 \\ B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0 \end{bmatrix}$
$Y_4^{(p)}$	$\begin{bmatrix} (1,35)(2,32)(1,26)(2,34)(2,20)(2,25)(1,34)(2,34)(2,30)(1,29)(1,24)(2,34)(2,20)(1,31)(1,29)(1,32)(2,22)(1,19) \\ (1,25)(2,31)(1,33)(2,23)(2,27)(1,35)(2,27)(2,23)(2,29)(2,24)(2,31)(2,25)(2,26)(2,30)(1,35)(1,23)(2,34)(1,20) \\ B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0 \\ B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0 \end{bmatrix}$
$Y_1^{(d)}$	$[(1,78)B_0(1,66)B_0(1,25)B_0(1,43)(1,89)(1,16)(1,27)(1,95)(1,44)(1,63)(1,82)(1,98)(1,29)(1,59)(1,68)(1,57)$ $(1,20)(1,17)(1,81)(1,77)(1,61)(1,13)(1,93)(1,54)(1,10)(1,5)(1,14)(1,58)(1,94)(1,97)(1,80)(1,55)]$
$Y_2^{(d)}$	$[(1,35)B_0(2,33)B_0(2,18)B_0(1,27)(1,35)(2,27)(2,33)(1,19)(1,35)(1,34)(1,28)(1,25)(1,24)(2,30)(1,18)(1,24)$ $(2,34)(2,27)(1,26)B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0]$
$Y_3^{(d)}$	$\begin{bmatrix} (1,26)(1,32)(1,67)(1,83)(1,60)(1,71)(1,87)(1,90)(1,3)(1,45)(1,84)(1,75)(1,28)(1,35)(1,62)(1,56)(1,85)(1,64) \\ (1,11)(1,36)(1,40)(1,73)(1,30)(1,4)(1,74)(1,23)(1,19)(1,91)(1,53)(1,46)(1,86)(1,76)(1,12)(1,101)(1,92)(1,8) \\ (1,47)(1,96)(1,39)(1,21)(1,34)(1,72)(1,69)(1,88)(1,38)(1,49)(1,65)(1,15)(1,100)(1,37)(1,42)(1,33)(1,102) \\ (1,7)(1,6)(1,70)(1,48)(1,79)(1,24)(1,51)(1,52)(1,41)(1,2)(1,31)(1,1)(1,9)(1,99)(1,18)(1,22)(1,50) \end{bmatrix}$
$Y_4^{(d)}$	$\begin{bmatrix} (2,21)(1,30)(1,19)(1,18)(2,22)(1,23)(1,29)(2,17)(1,34)(2,32)(2,35)(2,26)(1,28)(2,33)(1,25)(2,24)(1,31)(2,20) \\ (2,31)(2,34)(1,29)(1,23)(1,35)(2,20)(2,25)(1,33)(2,19)(2,21)(1,32)(1,17)(1,24)(2,27)(1,22)(1,28)(1,30)(1,26) \\ (1,27)B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0 \\ (2,18)B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0B_0 \end{bmatrix}$

Table 3 Bit arrangements Y_p and Y_d in the permutation and diffusion in the CPP and CD as illustrated in Figs. 5 and 8

Y_p

$$\begin{bmatrix}
(2,26)(1,14)(2,18)(1,16)(1,13)(1,23)(2,32)(2,27)(1,24)(2,32)(1,15)(2,33)(2,15)(1,24)(2,16)(1,25)\\
(2,13)(1,30)(1,32)(2,34)(2,35)(1,24)(1,19)(2,15)(1,24)(2,26)(2,30)(2,14)(1,28)(1,24)(2,16)(2,34)\\
(1,19)(1,22)(2,32)(2,22)(1,21)(2,21)(1,16)(2,18)(1,14)(2,22)(2,18)(1,19)(2,22)(2,15)(2,24)(1,29)\\
(1,18)(2,31)(2,14)(1,22)(1,13)(2,18)(2,13)(2,17)(1,16)(2,19)(2,17)(2,16)(2,26)(1,33)(1,34)(1,18)\\
(1,24)(1,21)(1,25)(2,19)(2,14)(1,23)(1,16)(2,13)(1,34)(1,22)(1,35)(2,30)(1,13)(2,28)(1,29)(1,27)\\
(2,19)(1,34)(2,33)(2,22)(1,13)(2,28)(1,32)(1,35)(2,14)(2,23)(1,26)(2,28)(2,29)(2,27)(1,29)(1,21)\\
(1,26)(2,15)(1,14)(1,35)(2,19)(2,26)(1,35)(1,17)(1,17)(2,20)(1,34)(1,21)(2,19)(1,16)(2,22)(2,21)\\
(2,16)(1,23)(2,15)(1,27)(2,13)(2,26)(1,31)(1,18)(2,23)(1,26)(2,14)(2,24)(2,27)(1,18)(2,32)(1,35)
\end{bmatrix}$$

$$\begin{bmatrix}
(1,25)(2,17)(1,18)\\
(2,20)(1,15)(2,30)\\
(2,20)(2,27)(1,31)\\
(1,13)(2,22)(2,29)\\
(2,33)(1,18)(2,15)\\
(1,34)(2,14)(1,26)\\
(1,24)(2,35)(2,21)\\
(2,14)(2,17)(1,22)
\end{bmatrix}$$

Y_d

$$\begin{bmatrix}
(1,26)(1,33)(2,22)(1,33)(2,32)(2,29)(1,18)(1,24)\\
(1,34)(2,30)(1,24)(2,29)(2,22)(1,28)(1,35)(2,33)\\
(2,23)(2,30)(1,24)(1,35)(2,19)(1,22)(2,32)(1,19)\\
(1,27)(1,32)(2,31)(1,20)(2,25)(1,24)(1,26)(2,28)\\
(1,20)(2,34)(2,29)(1,20)(1,28)(2,34)(2,32)(1,30)\\
(1,34)(1,21)(2,27)(1,29)(1,34)(1,29)(2,25)(1,31)\\
(2,32)(1,28)(2,19)(2,34)(1,18)(1,27)(2,20)(1,24)\\
(1,18)(2,30)(2,29)(1,31)(2,35)(1,24)(2,22)(1,33)
\end{bmatrix}$$

Table 4 Chosen initial values of state variables and control parameters

State variables	The permutation	The diffusion
$x_0^{(1)}$	0.1234567890	0.9876543210
$x_0^{(2)}$	0.0987654321	0.0123456789
$\gamma_0^{(1)}$	1.9753186420	2.8642097531
kP_0		[64 154 37 73 17 56 72 68]
kC_0		[123 11 27 88 33 211 97 63]

4.1.1 Simulation result of permutation

A pixel of an image is shuffled with another pixel of either itself or another image in the scheme of inter-image permutation as illustrated in Fig. 6b. The permutation does not change the value of pixels regardless to number of iterations R_p as well as that of permutation rounds N_p. Therefore, the intensity of each permuted image approaches to the mean intensity of test images (called a saturation value) after a number of permutation rounds. Here, the saturation value of intensity is 137.175 for the chosen set of test images.

Firstly, in order to choose a suitable value for iteration number R_p applied to the PCM of the CPP for each pixel, the permutation is simulated with varying number of R_p while fixed $N_p = 1$. Figure 9 displays the permuted images for $R_p = 1..4$ in each row for each image. Intuitively, the permuted images with $R_p = 1$ in the second column retain the intensities of original ones. Even the intensities among permuted images are unbalanced, the objects in the original images can not be recognized. For $R_p = 2..4$, the permuted images look random and their intensities get balanced. Table 5 and Fig. 10 show the change of intensity in the permuted images with respect to number of iterations $R_p = 1..9$. It is clear that, starting at $R_p = 2$, the intensities of permuted images approach to and fluctuate around the saturation value, 137.175. Number of iterations applied to the PCM should be chosen as $R_p \geq 3$ to guarantee the randomness and balance in the intensities of permuted images. Therefore, $R_p = 4$ is adopted for all the later simulations.

Then, the permutation is simulated with various number of permutation rounds while iteration number is fixed at $R_p = 4$. Figure 11 illustrates the permuted images with varying number of permutation rounds $N_p = 2...5$ in rows. Visually, the structural elements in the original images are replaced by the noise-like pattern. Table 6 and Fig. 12 show the intensity of permuted images with respect to N_p. It is clear that the intensities of permuted images are diminished to 137.175, the mean intensity of original images. The deviation of the intensities of permuted images is reduced significantly for $N_p \geq 3$, thus $N_p = 4$ is chosen for the later simulations of encryption.

It is noted that the most significant change in the intensities of the original images is observed with two special images (Black and White).

4.1.2 Simulation result of diffusion

Similar to the simulation of permutation, the diffusion is simulated with various number of iterations (R_d) applied to the PCM of the CD and with that of diffusion rounds N_d. Figure 13 illustrates the diffused images with various number of iterations $R_d = 1...4$ and fixed number of diffusion round $N_d = 1$. It is clear that the noise-like pattern is observed in the diffused images even at $R_d = 1$. The intensities of diffused images for $R_d = 1..9$ are

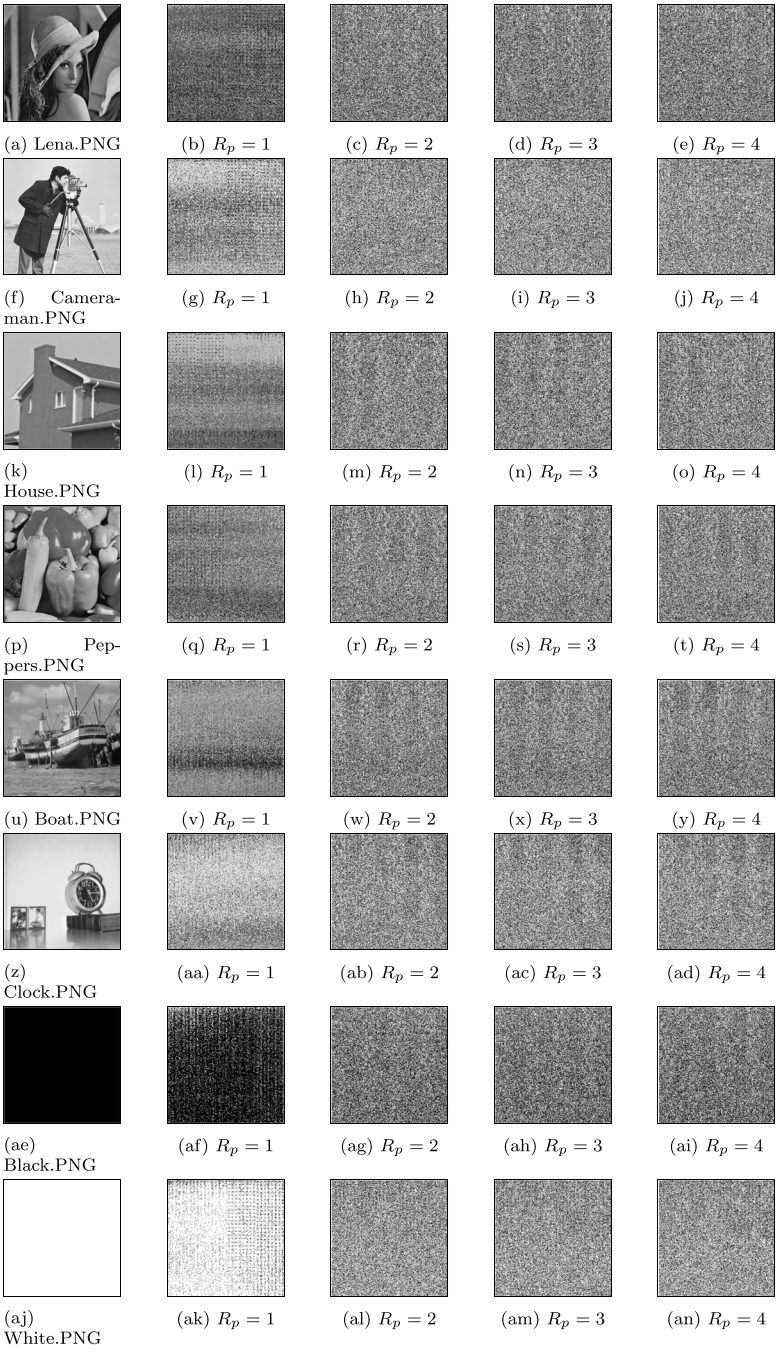


Fig. 9 Permuted images with various number of iterations $R_p = 1...4$ with a single permutation round at $N_p = 1$

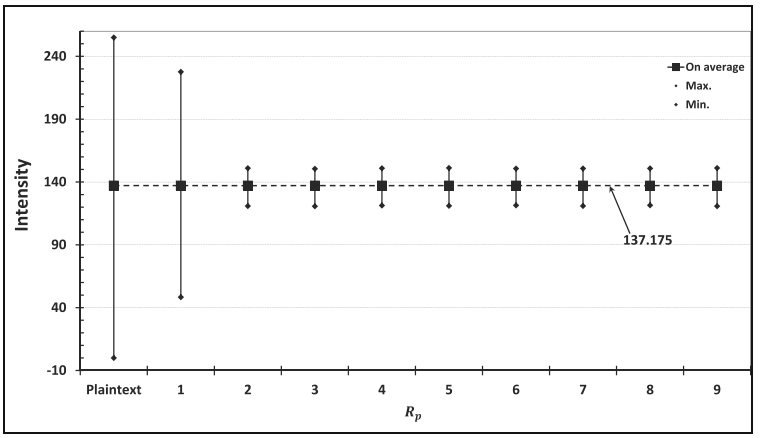


Fig. 10 Intensity of original and permuted images with various number of iterations, $R_p = 1..9$ and fixed $N_p = 1$

shown in Table 7 and Fig. 14. The intensities of diffused images fluctuate around 127.5 with small deviation for $R_d \geq 1$. The value of 127.5 is the ideal one for 8-bit grayscale images, or the central value of the range [0,255]. The number of iterations should be larger than 2, and $R_d = 4$ is chosen for all the later diffusion simulations.

Furthermore, the diffusion is simulated with various number of diffusion rounds N_d and fixed number of iterations at $R_d = 4$. Figure 15 shows the diffused images with various number of diffusion rounds $N_d = 1...4$. It looks all diffused images random and independent from number of diffusion rounds, N_d. More specifically, Table 8 and Fig. 16 show that the intensities of diffused images rapidly approach and fluctuate around ideal value (127.5). The fluctuation is low and obtained even at $N_d = 1$, thus the encryption should adopt $N_d \geq 2$. $N_d = 4$ is chosen for the simulations of encryption later.

4.1.3 Simulation result of encryption

According to the simulation results for the permutation and diffusion given above, the encryption is simulated with the parameters chosen at fixed $R_p = R_d = 4$ for the PCMs in the CPP and CD, and $N_p = N_d = 4$ for the permutation and diffusion, while number of encryption rounds is varying as $N_e = 1..5$.

Figure 17 displays the simulation result of encryption for $N_e = 1..4$, where original images and corresponding ciphered ones are presented in rows. Intuitively, the encrypted images have good randomness and balanced intensities.

4.2 Statistical analyses

Let us quantify the randomness of the encrypted images by means of histogram, information entropy and correlation of two adjacent pixels. The encryption performs on multiple images at the same time, so the mean values of the measures are considered for the effectiveness of the encryption.

Table 5 Intensity of original and permuted images with various number of iterations R_p

	Intensity of									
	Original	Permuted image with								
		$R_p = 1$	$R_p = 2$	$R_p = 3$	$R_p = 4$	$R_p = 5$	$R_p = 6$	$R_p = 7$	$R_p = 8$	$R_p = 9$
Lena	98.677	101.268	130.429	130.402	129.608	129.571	129.962	130.324	130.167	130.205
Cameraman	166.942	155.871	144.979	145.089	145.197	144.595	144.801	145.082	144.777	144.968
	137.985	135.221	136.974	136.762	137.231	136.858	136.943	136.890	136.927	136.909
Peppers	123.104	123.282	134.755	135.320	134.768	135.037	135.215	135.306	135.113	135.142
Boat	129.712	131.736	134.301	134.417	134.346	134.432	134.505	133.786	133.956	133.977
Clock	185.980	173.962	144.074	144.176	144.011	144.653	143.844	144.354	144.138	144.318
Black	0.000	48.407	120.899	120.701	121.265	121.059	121.395	120.909	121.468	120.735
White	255.000	227.653	150.989	150.533	150.975	151.195	150.735	150.749	150.854	151.147
On average	137.175									

4.2.1 Histogram analysis

The histogram of an image provides the distribution of pixel values. Let us consider the statistical analysis of histogram by means of χ^2 for a 8-bit grayscale image as

$$\chi^2 = \sum_{i=0}^{255} \frac{(O_i - E_i)^2}{E_i}, \tag{19}$$

where the expected occurrence frequency E_i for the image with the size of $M \times N$ is $\frac{M*N}{256}$; observed occurrence frequency O_i is the number of pixels with the value i. The significance of histogram is considered if it is conformed a uniform distribution by means of a hypothesis test. Here, the hypothesis test is accepted if $\chi^2 \leq \chi^2_\alpha(255)$; α is the significance level. It is chosen as $\alpha = 0.05$, so $\chi^2_{0.05}(255) = 293.247$. This means that the histogram is considered as a uniform distribution if $\chi^2 \leq 293.247$.

Table 9 shows the χ^2-test results for the encrypted images with various number of encryption rounds. Remarkably, italicized numbers indicate that χ^2 tests are not passed. For $N_e \geq 3$, all encrypted images pass the χ^2 test with the values less than 293.247. In other words, the distribution of pixel values of encrypted images is uniform for $N_e \geq 3$.

4.2.2 Information entropy

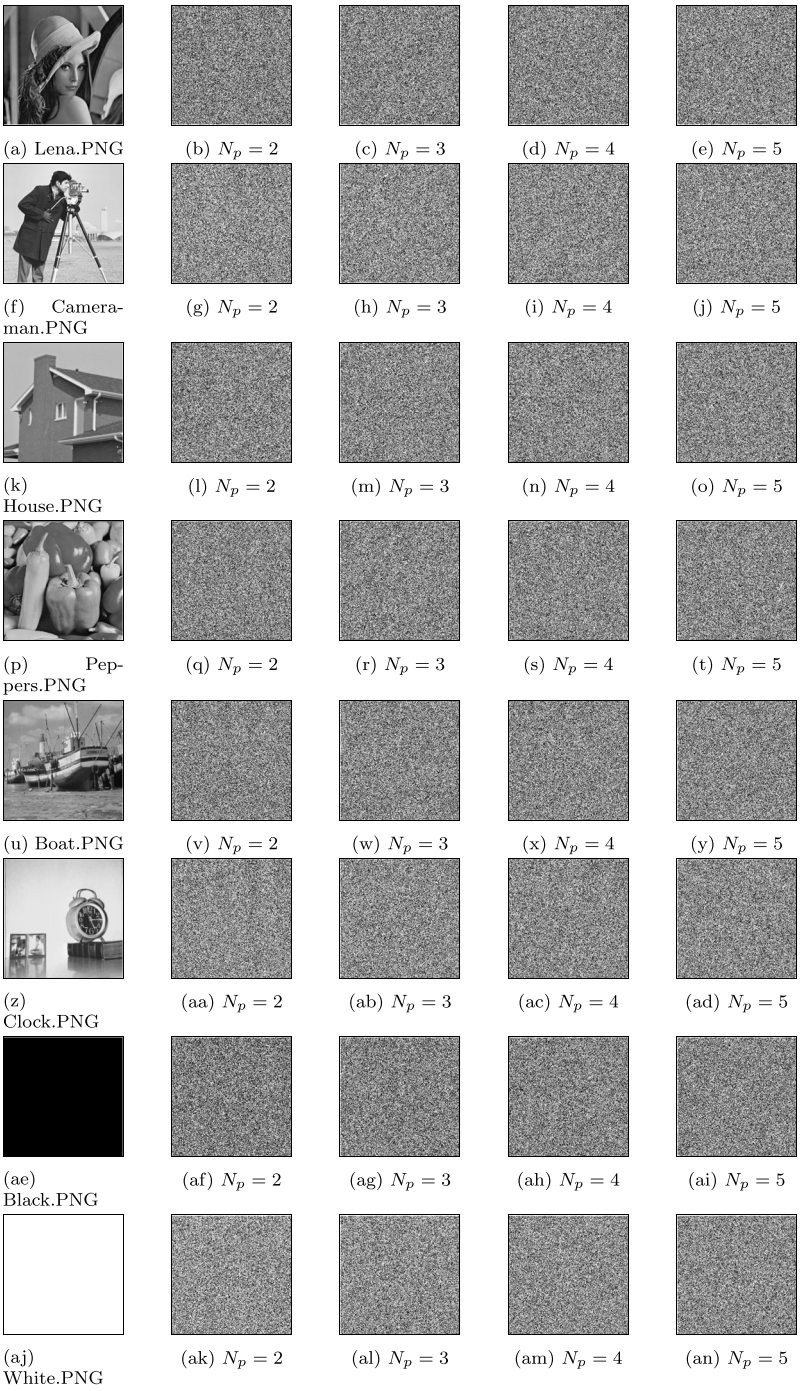
Information entropy of an image indicates the probability of values of pixels v_i, $p(v_i)$, for a 8-bit grayscale image and it is computed by

$$IE = \sum_{i=0}^{255} p(v_i) log_2 \frac{1}{p(v_i)} \quad \text{(bits)}. \tag{20}$$

It is expected that an encrypted image has IE as close to the ideal value (8 bits for a 8-bit grayscale image) as possible. Table 10 displays the information entropy of plaintext images and that of corresponding encrypted ones with various number of encryption rounds. The information entropy of individual encrypted images and its averages are greater than 7.99 for any number of encryption rounds. Figure 18 presents the entropy approaching to 7.9972

Table 6 Intensity of original and permuted images

	Intensity					
	Original	Permuted image with				
		$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$	$N_p = 5$
Lena	98.677	129.608	132.090	134.249	135.131	135.677
Cameraman	166.942	145.197	142.859	141.069	139.850	139.249
House	137.985	137.231	136.873	137.061	137.127	136.843
Peppers	123.104	134.768	135.681	136.168	136.672	136.733
Boat	129.712	134.346	135.028	135.874	136.021	136.310
Clock	185.980	144.011	141.249	139.631	138.460	138.609
Black	0.000	121.265	127.585	131.416	133.490	134.649
White	255.000	150.975	146.036	141.932	140.649	139.331
On average	137.175					



(a) Lena.PNG (b) $N_p = 2$ (c) $N_p = 3$ (d) $N_p = 4$ (e) $N_p = 5$

(f) Cameraman.PNG (g) $N_p = 2$ (h) $N_p = 3$ (i) $N_p = 4$ (j) $N_p = 5$

(k) House.PNG (l) $N_p = 2$ (m) $N_p = 3$ (n) $N_p = 4$ (o) $N_p = 5$

(p) Peppers.PNG (q) $N_p = 2$ (r) $N_p = 3$ (s) $N_p = 4$ (t) $N_p = 5$

(u) Boat.PNG (v) $N_p = 2$ (w) $N_p = 3$ (x) $N_p = 4$ (y) $N_p = 5$

(z) Clock.PNG (aa) $N_p = 2$ (ab) $N_p = 3$ (ac) $N_p = 4$ (ad) $N_p = 5$

(ae) Black.PNG (af) $N_p = 2$ (ag) $N_p = 3$ (ah) $N_p = 4$ (ai) $N_p = 5$

(aj) White.PNG (ak) $N_p = 2$ (al) $N_p = 3$ (am) $N_p = 4$ (an) $N_p = 5$

Fig. 11 Permuted images with various number of permutation rounds, $N_p = 2...5$ and fixed $R_p = 4$

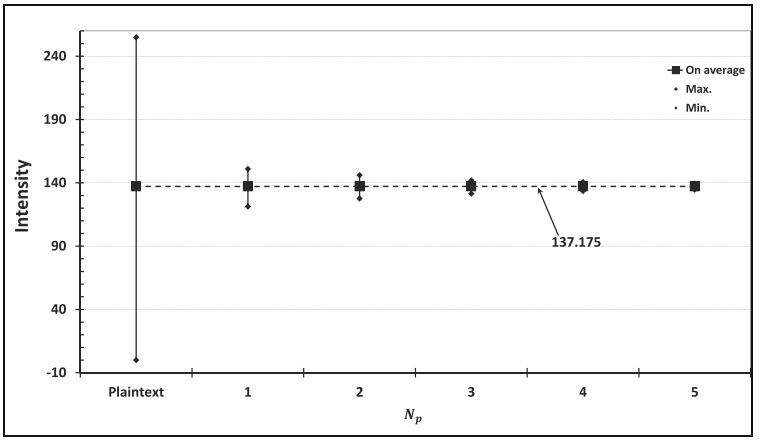


Fig. 12 Intensity of original and permuted images with various number of permutation rounds $N_p = 1...5$ and fixed $R_p = 4$

as the increase of N_e. This means that pixel values of the encrypted images are almost random for $N_e \geq 2$.

4.2.3 Correlation of two adjacent pixels

It is well-known that the equation for the Pearson correlation coefficient of two sequences X and Y is

$$\rho_{X,Y} = \frac{\sum_{i=1}^{N_{pair}} (x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\left(\sum_{i=1}^{N_{pair}} (x_i - \overline{X})^2\right)\left(\sum_{i=1}^{N_{pair}} (y_i - \overline{Y})^2\right)}}, \tag{21}$$

where x_i and y_i are values of elements in the sequences X and Y, respectively; It is assumed that X and Y are the same length, N_{pair}; $\overline{X}$ and $\overline{Y}$ are the means of X and Y, respectively. The lower correlation coefficient is, the higher independence of a sequence from the other is. In the case of 2D image, the correlation coefficients of adjacent pixel pairs are measured to quantify the visual structure of images. Specifically, the pairs of adjacent pixels are chosen in three directions, i.e. horizontal (H), vertical (V) and diagonal (D). Here, x_i and y_i are values of adjacent pixels in a specified direction, and the number of pairs N_{pair} is adopted by all possible pairs over the image.

Table 11 shows the correlation coefficients of the original images and corresponding encrypted ones. The correlation coefficients of encrypted images are very small in compared with those of original images. As seen in Table 12, the average correlation coefficients over all images in each direction are also small. It means that the micro visual structures in the original images are almost completely removed and replaced by the random pattern to become the encrypted images, even after a single round of encryption.

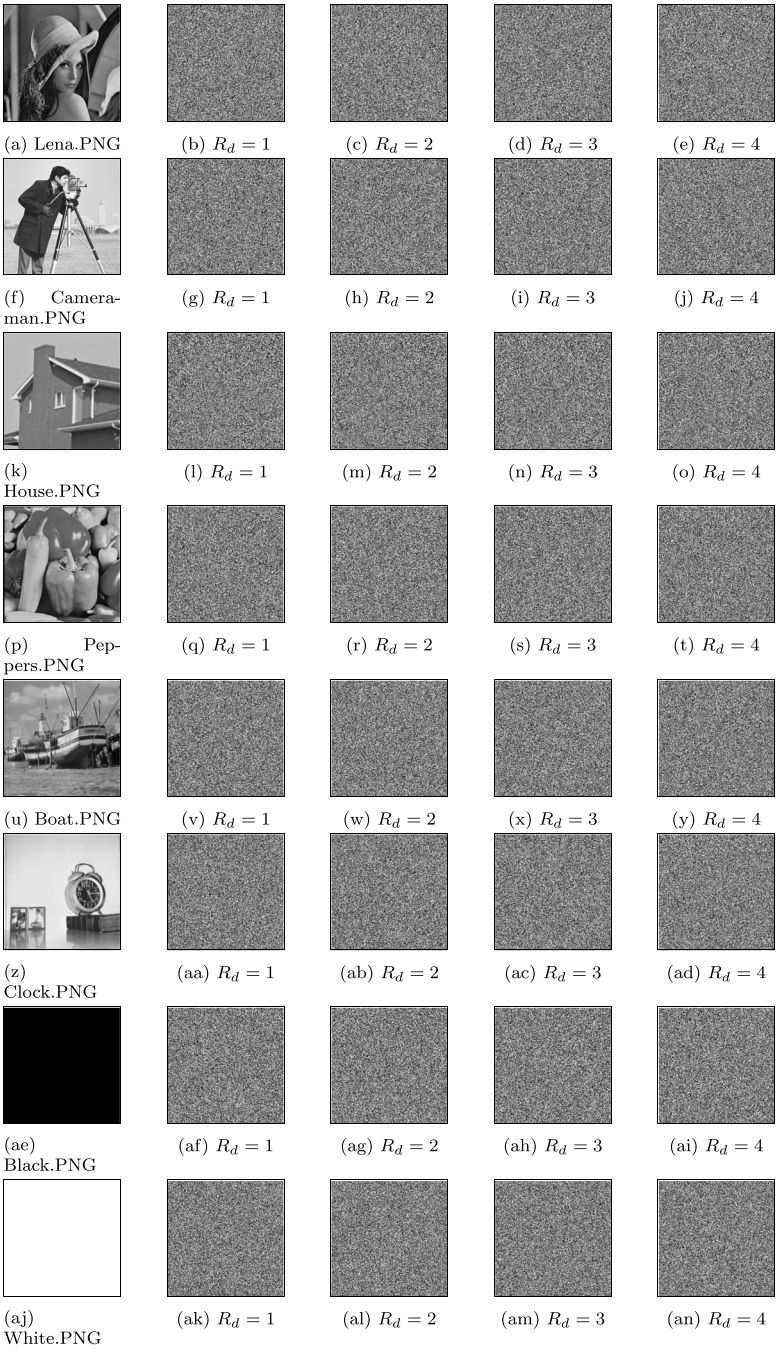


Fig. 13 Diffused images with various number of iterations, $R_d = 1...4$, with a single diffusion round at $N_d = 1$

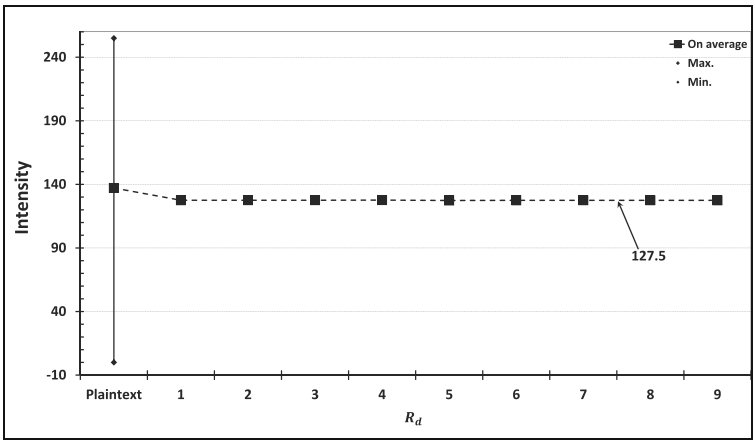


Fig. 14 Intensity of original and diffused images with various number of iterations $R_d = 1...9$ and fixed $N_d = 1$

4.3 Security analyses

4.3.1 Space of secret key

It is obvious that the secret key of the cipher is constructed by the number of flippable bits in the CPP and CD, and those of initial values of kP_0 and kC_0. Table 1 displays the number of bits from each parameter contributing to the secret key, i.e. 102 bits by the permutation and 230 bits by the diffusion. Therefore, the total number of bits of the secret key is 332, or the space of secret key is 2^{332}. That is large enough to resist the type of brute force attack on the nowadays computer.

4.3.2 Sensitivity of secret key

The sensitivity of secret key is considered by the difference in two versions of encrypted images which are produced by the cryptosystem using two value sets of secret key, i.e. the original one and modified one. To evaluate the sensitivity of secret key on a parameter, the value of a single interested parameter of the secret key is slightly modified to have a modified value set of secret key. The original and modified value sets of secret key are used for encrypting the same set of test images. The difference in two sets of encrypted images given by the tolerance in the secret key is estimated by the ciphertext difference rate (Cdr), the number of pixels change rate ($NPCR$), and the unified averaged changed intensity ($UACI$).

The ciphertext difference rate (Cdr) quantifies the sensitivity of secret key. It is calculated by

$$Cdr = \frac{Diff(C, C_1) + Diff(C, C_2)}{2M * N} 100\%, \tag{22}$$

where M and N are the number of rows and columns of pixels; Respectively, C, C_1 and C_2 are three ciphertext images what are produced by the encryption with the use of three value sets of secret key K, $K + \Delta K$ and $K - \Delta K$. The difference between corresponding pairs of images is measured by the difference function $Diff(A, B)$, which returns the number of

Table 7 Intensity of original and diffused images with various number of iterations R_d

	Intensity of									
	Original	Diffused image with								
		$R_d = 1$	$R_d = 2$	$R_d = 3$	$R_d = 4$	$R_d = 5$	$R_d = 6$	$R_d = 7$	$R_d = 8$	$R_d = 9$
Lena	98.677	127.707	127.683	127.441	127.902	127.430	127.279	127.389	127.204	127.294
Cameraman	166.942	127.420	127.522	127.477	127.167	127.862	127.403	126.904	127.292	127.636
House	137.985	127.989	127.427	127.821	127.486	127.104	127.775	127.078	127.538	127.690
Peppers	123.104	127.843	127.814	127.635	127.436	127.159	127.561	127.578	127.402	127.590
Boat	129.712	127.563	127.914	127.688	127.486	127.304	127.412	128.030	127.886	127.578
Clock	185.980	127.321	127.161	127.521	128.015	127.354	127.329	127.874	127.378	127.380
Black	0.000	127.290	127.483	127.636	127.904	127.279	127.175	127.616	127.517	127.633
White	255.000	127.101	127.361	127.200	127.706	127.276	127.677	127.477	127.643	126.850
On average	137.175	127.529	127.546	127.552	127.638	127.346	127.451	127.493	127.482	127.456

pixels in the image A with values different from those in B. It is expressed by

$$Diff(A, B) = \sum_{x=1}^{M} \sum_{y=1}^{N} Difp(A(x, y), B(x, y)), \tag{23}$$

where $Difp(.)$ is considered by a pair of pixels at the position (x, y) as

$$Difp(A(x, y), B(x, y)) = \begin{cases} 1, & \text{for } A(x, y) \neq B(x, y), \\ 0, & \text{for } A(x, y) = B(x, y). \end{cases} \tag{24}$$

In this work, the original value set of secret key is as given in Table 4. The amounts of tolerance ΔK to evaluate the sensitivity for interested parameters of secret key are adopted as small as possible. Here, those are chosen as equal to the precision of value representation for parameters presented in Table 13. It is noted that the names of interested parameters are showed in the subscript and those belonging to the permutation and diffusion are indicated in the superscript. To evaluate the sensitivity of a specific parameter, the simulation is carried out with a modified value of only single interested parameter while the value of other parameters is unchanged as it is in the original value set of secret key. Therefore, there are eight modified value sets of the secret key corresponding to eight parameters as given in Table 14. It is noted that the tolerances kP_0 and kC_0 are only applied to the last elements, i.e. $kP_0(8)$ and $kC_0(8)$. The plain images are encrypted using the original value and eight modified value sets of secret key for estimation of the sensitivity.

Table 15 displays Cdr for the sensitivity on parameters of secret key. For $N_e \geq 2$, a slight difference in the value of secret key makes, on average, the value of more than 99.5% pixels changed in encrypted images. This means that the encryption is very high sensitivity to every parameter of the secret key.

Besides, the sensitivity of secret key can be analysed by using the $NPCR$ and $UACI$ between a pair of encrypted images C and C_1 using the original value of secret key K and its modified version $K + \Delta K$. The equations for $NPCR$ and $UACI$ are as

$$NPCR = \frac{Diff(C, C_1)}{M * N} 100\%, \tag{25}$$

Table 8 Intensity of original and diffused images

	Intensity					
	Original	Diffused image with				
		$N_d = 1$	$N_d = 2$	$N_d = 3$	$N_d = 4$	$N_d = 5$
Lena	98.677	127.902	127.358	127.002	127.461	126.887
Cameraman	166.942	127.167	127.357	127.561	127.232	127.748
House	137.985	127.486	127.448	127.447	127.533	127.412
Peppers	123.104	127.436	128.002	126.794	127.467	127.543
Boat	129.712	127.486	127.618	127.673	127.690	127.860
Clock	185.980	128.015	127.428	127.043	127.383	127.708
Black	0	127.904	127.333	127.278	127.581	127.237
White	255	127.706	127.918	127.749	127.472	127.287
On average	142.675	127.638	127.588	127.319	127.477	127.460

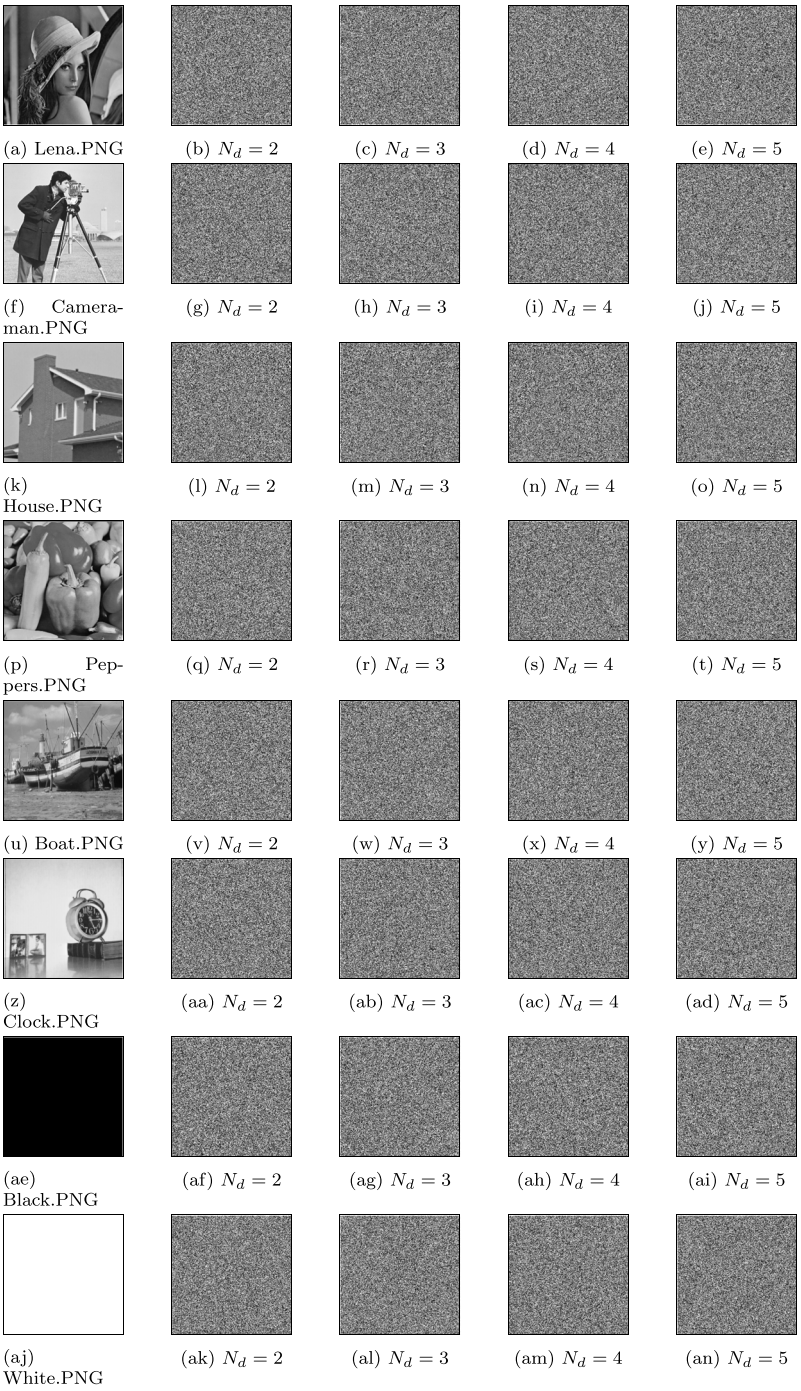


Fig. 15 Diffused images with various number of diffusion rounds, $N_d = 2...5$ and fixed $R_d = 4$

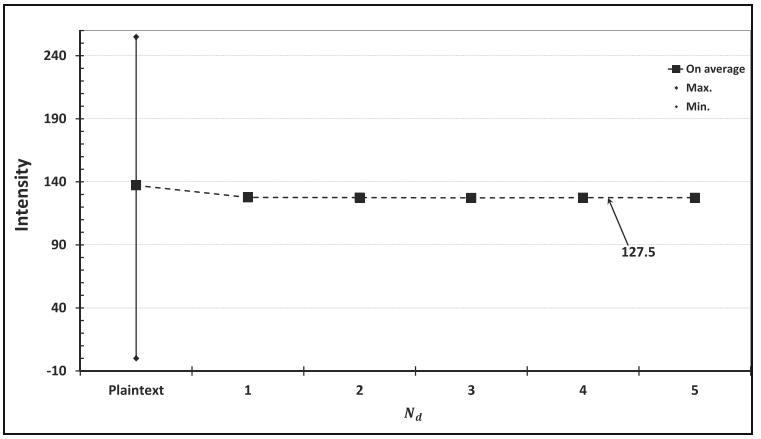


Fig. 16 Intensity of original and diffused images with various number of diffusion rounds $N_d = 1...5$ and fixed $R_d = 4$

and

$$UACI = \frac{1}{M * N}\left[\sum_{x,y}\frac{|C(x,y) - C_1(x,y)|}{255}\right]100\%, \tag{26}$$

where $C_(x, y)$ and $C_1(x, y)$ are pixels at location (x, y) of encrypted images C and C_1, respectively; the difference function $Diff(.)$ is as shown in (23).

However, the question is that how high the sensitivity of secret key is enough that a cryptosystem can resist from differential attacks. The statistical hypothesis for the $NPCR$ and $UACI$ tests is employed to answer this question. The critical values of tests for $NPCR$ and $UACI$ were estimated at a significance level α as presented in [84]. For 8-bit grayscale images with the size of 256×256, the critical value at $\alpha = 0.05$ for $NPCR$ is $NPCR^*_{0.05} = 99.569\%$ and that for $UACI$ is in the range of $UACI^{*-}_{0.05} = 33.282\%$ and $UACI^{*+}_{0.05} = 33.644\%$. The $NPCR$ and $UACI$ scores what are calculated from the

Table 9 Histogram analysis

χ^2 Test ($\chi^2_{0.05}(255) = 293.247$)						
Image	Plaintext	Encryption round (N_e)				
		1	2	3	4	5
Lena	30577.7	*10579.453*	*517.094*	292.859	245.680	272.906
Cameraman	161271.9	262.094	*293.289*	259.734	251.633	283.320
House	299789.2	*11844.953*	*494.414*	238.570	250.570	200.359
Peppers	36777.5	281.234	245.539	229.086	237.078	258.672
Boat	100674.9	263.625	178.438	283.094	256.906	270.914
Clock	282061.6	263.039	274.719	255.297	239.781	241.984
Black	16711680	*2584.398*	*313.969*	258.453	258.352	249.633
White	16711680	229.148	222.086	282.133	276.711	255.813
On average	4291814.1	*3288.493*	*317.443*	262.403	252.089	254.200

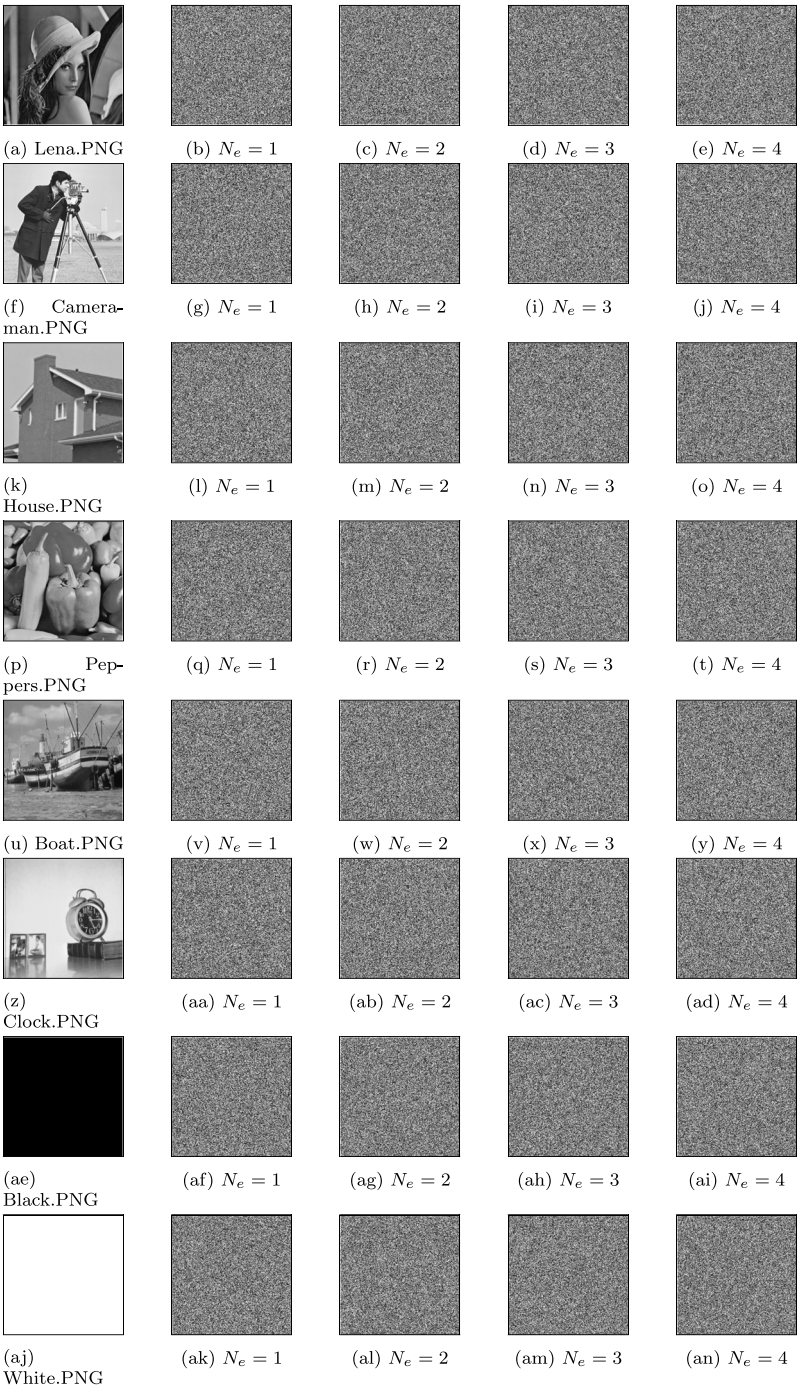


Fig. 17 Encrypted images with a number of encryption rounds N_e and fixed $N_p = N_d = 4$

Table 10 Information entropy (IE)

Information Entropy						
Image	Plaintext	Encryption round (N_e)				
		1	2	3	4	5
Lena	7.5327	7.8957	7.9944	7.9968	7.9973	7.9970
Cameraman	6.9046	7.9971	7.9968	7.9971	7.9972	7.9969
House	6.4971	7.8834	7.9946	7.9974	7.9972	7.9978
Peppers	7.5330	7.9969	7.9973	7.9975	7.9974	7.9971
Boat	7.5691	7.9971	7.9980	7.9969	7.9972	7.9970
Clock	6.7057	7.9971	7.9970	7.9972	7.9973	7.9973
Black	0	7.9714	7.9965	7.9972	7.9972	7.9973
White	0	7.9975	7.9975	7.9969	7.9970	7.9972
On average	5.2960	7.9670	7.9965	7.9971	7.9972	7.9972

simulation are sufficient to resist from differential attacks if $NPCR \geq NPCR^*_{0.05}$ and $UACI \in [UACI^{*-}_{0.05}, UACI^{*+}_{0.05}]$

The simulation results for $NPCR$ and $UACI$ are shown in Tables 16 and 17, respectively. For $N_e \geq 4$ all the individual values of $NPCR$ and $UACI$ pass the tests of statistical hypothesis. In other words, values of $NPCR$ are larger than the critical one $NPCR^*_{0.05}$ and those of $UACI$ are within the range of $[UACI^{*-}_{0.05}, UACI^{*+}_{0.05}]$, respectively. In other words, the tests for the sensitivity of secret key are passed for every parameters of secret key. Moreover, all the average values of $NPCR$ and $UACI$ pass the tests for $N_e \geq 3$. In consequence, a small change in the secret key makes large change in the ciphertexts, and it can resist from differential attacks based on the analysis of secret key for $N_e \geq 4$.

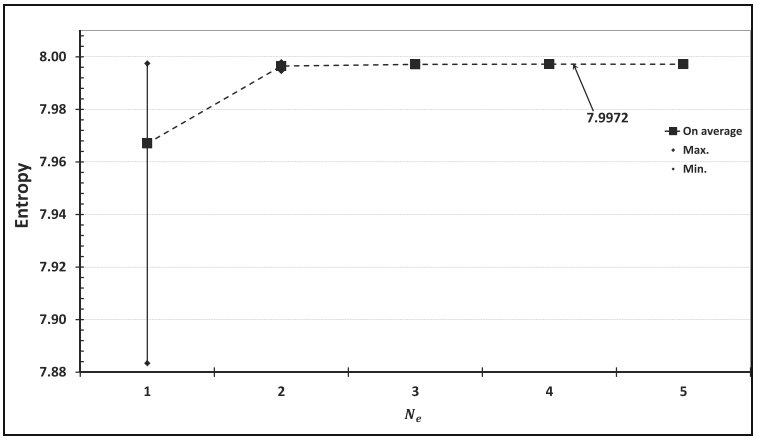


Fig. 18 Entropy of encrypted images with various number of encryption rounds $N_e = 1...5$ and fixed $R_p = R_d = 4$, $N_p = N_d = 4$

Table 11 Correlation coefficients of original images and their encrypted ones

Encryp. round	Correlation coefficient $\rho_{X,Y}$					
(N_e)	H	V	D	H	V	D
	Lena			Cameraman		
Plaintext	0.93998	0.96934	0.91793	0.91957	0.95494	0.89619
1	0.00004	-0.00219	-0.00391	-0.00173	-0.00228	-0.00104
2	0.00262	0.00471	0.00182	-0.00598	-0.00016	-0.00240
3	0.00246	-0.00367	0.00412	0.00306	-0.00050	0.00235
4	-0.00351	0.00115	0.00239	-0.00204	0.00571	0.00229
5	0.00307	0.00675	0.00038	0.00990	-0.00034	0.00240
	House			Peppers		
Plaintext	0.93998	0.96934	0.91793	0.91957	0.95494	0.89619
1	-0.00334	-0.00034	0.00264	0.00186	-0.00091	-0.00098
2	0.00635	0.00112	0.00460	-0.00475	0.00555	-0.00308
3	0.00400	-0.00309	0.00025	0.00366	-0.00679	-0.00353
4	-0.00054	-0.00305	0.00305	0.00291	0.00553	0.00159
5	0.00530	-0.00016	0.00303	0.00040	0.00140	-0.00526
	Boat			Clock		
Plaintext	0.93998	0.96934	0.91793	0.91957	0.95494	0.89619
1	0.00253	-0.00213	-0.00066	-0.00366	-0.00111	-0.00463
2	-0.00126	0.01044	-0.00280	-0.00492	-0.00747	0.00017
3	-0.00157	-0.00005	0.00022	-0.00117	0.00391	-0.00463
4	-0.00240	-0.00072	-0.00464	-0.00806	0.00496	0.00297
5	-0.00133	-0.00022	-0.00168	0.00299	0.00096	0.00504
	Black			White		
Plaintext	0.93998	0.96934	0.91793	0.91957	0.95494	0.89619
1	0.00273	-0.00116	-0.00494	0.00445	-0.00117	-0.00444
2	-0.00369	0.00326	-0.00385	0.00127	0.00247	0.00540
3	0.00402	0.00336	-0.00001	0.00330	-0.00198	-0.00566
4	-0.00398	-0.00565	-0.00343	-0.00510	0.00315	-0.00797
5	-0.00132	0.00047	-0.00494	0.00276	0.00537	0.00272

4.3.3 Analysis of differential attacks

Differential attack is based on analysis a pair of ciphered images produced by encryption of a pair of plain images for the purpose to learn about the secret key used in the encryption. It is noted that a pair of plain images are used in which one of them is slightly different from the other. In fact, the analysis of differential attack is to measure the plaintext sensitivity, and it is characterized by the $NPCR$ and $UACI$ of a pair of ciphered images.

For the multiple grayscale-image encryption, the set of eight original images are encrypted using the secret key as given in Table 4 to achieve the set of eight encrypted

Table 12 Average correlation coefficients $\bar{\rho}_{X,Y}$ of original and encrypted images

Encryption round (N_e)	On average		
	H	V	D
Plaintext	NaN	NaN	NaN
1	0.00036	-0.00141	-0.00225
2	-0.00130	0.00249	-0.00002
3	0.00222	-0.00110	-0.00086
4	-0.00284	0.00138	-0.00047
5	0.00272	0.00178	0.00021

images, each encrypted image is denoted by C. Then, one of original images is modified slightly and that together with the other original images are encrypted to produce another set of eight ciphered images, each encrypted image is named C_1. The analysis of $NPCR$ and $UACI$ as given in Eqs. (25)-(26) is carried on every pair of ciphered images C and C_1. To analyse the sensitivity of each plaintext, eight sets of modified images are encrypted individually using the same value of secret key. $NPCR$ and $UACI$ are calculated for every pair of ciphertexts and their averages are obtained for each of sets of modified images. Similar to the analysis of the sensitivity of secret key, $NPCR$ and $UACI$ are compared with the critical value and the critical range to evaluate the randomness tests for image encryption [84].

Normally, a modified image is obtained by choosing a random pixel from the plain image and slight change applied to adopted pixel value. In this example, the last pixel of image at the position (255,255) is chosen deliberately to minimize the affect of pixel selection to the result encrypted images due that the diffusion is carried out in the forward direction. The value of chosen pixel is added 1 to if it is less than 255, or subtracting 1 if it is equal to 255. With this modification, the sensitivity of plaintext is lowest in the first round of encryption, and it is increased with larger number of encryption rounds.

Tables 18 and 19 show the result of $NPCR$ and $UACI$ measured for the sensitivity of plaintexts with various number of encryption rounds. The individual values of $NPCR$ and

Table 13 The value of ΔK for Cdr, $NPCR$ and $UACI$

Stage	ΔK	Value of tolerance
Permutation	$\Delta K^{(p)}_{x_0^{(1)}}$	2^{-32}
	$\Delta K^{(p)}_{x_0^{(2)}}$	2^{-32}
	$\Delta K^{(p)}_{\gamma_0^{(1)}}$	2^{-32}
Diffusion	$\Delta K^{(d)}_{x_0^{(1)}}$	2^{-32}
	$\Delta K^{(d)}_{x_0^{(2)}}$	2^{-32}
	$\Delta K^{(d)}_{\gamma_0^{(1)}}$	2^{-32}
	ΔK_{kC_0}	1
	ΔK_{kP_0}	1

Table 14 The original value of secret key and eight modified versions

Value of secret key	Elements of secret key							
	The permutation			The diffusion				
K	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K+\Delta K^{(p)}_{x_0^{(1)}}$	$x_0^{(1)}+\Delta K^{(p)}_{x_0^{(1)}}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K-\Delta K^{(p)}_{x_0^{(1)}}$	$x_0^{(1)}-\Delta K^{(p)}_{x_0^{(1)}}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K+\Delta K^{(p)}_{x_0^{(2)}}$	$x_0^{(1)}$	$x_0^{(2)}+\Delta K^{(p)}_{x_0^{(2)}}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K-\Delta K^{(p)}_{x_0^{(2)}}$	$x_0^{(1)}$	$x_0^{(2)}-\Delta K^{(p)}_{x_0^{(2)}}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K+\Delta K^{(p)}_{\gamma_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}+\Delta K^{(p)}_{\gamma_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K-\Delta K^{(p)}_{\gamma_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}-\Delta K^{(p)}_{\gamma_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K+\Delta K^{(d)}_{x_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}+\Delta K^{(d)}_{x_0^{(1)}}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K-\Delta K^{(d)}_{x_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}-\Delta K^{(d)}_{x_0^{(1)}}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K+\Delta K^{(d)}_{x_0^{(2)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}+\Delta K^{(d)}_{x_0^{(2)}}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K-\Delta K^{(d)}_{x_0^{(2)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}-\Delta K^{(d)}_{x_0^{(2)}}$	$\gamma_0^{(1)}$	kP_0	kC_0
$K+\Delta K^{(d)}_{\gamma_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}+\Delta K^{(d)}_{\gamma_0^{(1)}}$	kP_0	kC_0
$K-\Delta K^{(d)}_{\gamma_0^{(1)}}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}-\Delta K^{(d)}_{\gamma_0^{(1)}}$	kP_0	kC_0
$K+\Delta K_{kP_0}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$kP_0+\Delta K_{kP_0}$	kC_0
$K-\Delta K_{kP_0}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$kP_0-\Delta K_{kP_0}$	kC_0
$K+\Delta K_{kC_0}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	$kC_0+\Delta K_{kC_0}$
$K-\Delta K_{kC_0}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	$x_0^{(1)}$	$x_0^{(2)}$	$\gamma_0^{(1)}$	kP_0	$kC_0-\Delta K_{kC_0}$

Table 15 Sensitivity of the secret key by means of Cdr with various encryption rounds N_e

N_e	Image	Cdr (%) on							
		The permutation			The diffusion				
		$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	kC_0	kP_0
1	Lena	99.571	99.535	98.412	98.460	99.578	98.469	98.462	98.425
	Cameraman	99.626	99.588	99.593	99.612	99.635	99.604	99.619	99.603
	House	99.557	99.524	98.425	98.508	99.513	98.405	98.469	98.454
	Peppers	99.634	99.652	99.594	99.590	99.619	99.585	99.623	99.599
	Boat	99.611	99.635	99.594	99.626	99.632	99.616	99.621	99.614
	Clock	99.621	99.586	99.590	99.608	99.622	99.596	99.628	99.635
	Black	99.615	99.604	99.218	99.220	99.622	99.181	99.231	99.228
	White	99.629	99.621	99.603	99.596	99.609	99.619	99.613	99.608
	On average	99.608	99.593	99.254	99.278	99.604	99.259	99.283	99.271
2	Lena	99.604	99.631	99.421	99.399	99.619	99.391	99.420	99.406
	Cameraman	99.575	99.620	99.593	99.582	99.581	99.597	99.586	99.630
	House	99.626	99.598	99.476	99.427	99.592	99.467	99.480	99.467
	Peppers	99.628	99.625	99.631	99.622	99.594	99.619	99.587	99.605
	Boat	99.596	99.610	99.591	99.606	99.641	99.609	99.589	99.593
	Clock	99.612	99.601	99.625	99.614	99.613	99.592	99.625	99.614
	Black	99.632	99.590	99.525	99.519	99.590	99.544	99.560	99.553
	White	99.607	99.652	99.601	99.622	99.588	99.609	99.594	99.609
	On average	99.610	99.616	99.558	99.549	99.602	99.553	99.555	99.559
3	Lena	99.646	99.616	99.580	99.568	99.593	99.610	99.605	99.567
	Cameraman	99.584	99.615	99.590	99.609	99.583	99.581	99.629	99.606
	House	99.626	99.622	99.623	99.566	99.643	99.591	99.577	99.568
	Peppers	99.637	99.622	99.615	99.616	99.624	99.597	99.609	99.617
	Boat	99.620	99.618	99.596	99.591	99.639	99.590	99.615	99.596
	Clock	99.602	99.622	99.625	99.631	99.580	99.616	99.614	99.628
	Black	99.615	99.596	99.621	99.645	99.628	99.604	99.579	99.601
	White	99.597	99.610	99.606	99.585	99.592	99.599	99.612	99.629
	On average	99.616	99.615	99.607	99.601	99.610	99.599	99.605	99.601
4	Lena	99.637	99.619	99.602	99.626	99.592	99.594	99.605	99.619
	Cameraman	99.591	99.616	99.613	99.593	99.638	99.583	99.625	99.593
	House	99.609	99.615	99.619	99.601	99.591	99.625	99.644	99.588
	Peppers	99.612	99.590	99.622	99.620	99.615	99.596	99.599	99.592
	Boat	99.601	99.612	99.596	99.635	99.591	99.635	99.624	99.582
	Clock	99.618	99.603	99.603	99.612	99.614	99.620	99.580	99.588
	Black	99.617	99.596	99.573	99.592	99.611	99.571	99.599	99.624
	White	99.596	99.604	99.606	99.570	99.625	99.631	99.607	99.618
	On average	99.610	99.607	99.604	99.606	99.610	99.607	99.610	99.601
5	Lena	99.603	99.602	99.587	99.619	99.597	99.600	99.589	99.596
	Cameraman	99.615	99.614	99.625	99.629	99.603	99.602	99.590	99.606
	House	99.579	99.614	99.606	99.584	99.627	99.594	99.612	99.596
	Peppers	99.599	99.601	99.600	99.617	99.605	99.629	99.640	99.596
	Boat	99.608	99.596	99.610	99.644	99.594	99.608	99.635	99.609
	Clock	99.614	99.616	99.662	99.614	99.635	99.629	99.593	99.609
	Black	99.571	99.603	99.592	99.603	99.602	99.585	99.594	99.578
	White	99.606	99.625	99.593	99.607	99.616	99.618	99.617	99.618
	On average	99.599	99.609	99.610	99.615	99.610	99.608	99.609	99.601

Table 16 Sensitivity of the secret key by means of $NPCR$ with various encryption rounds N_e

N_e	Image	$NPCR$ (%) test on ($NPCR^*_{0.05} = 99.569\%$)							
		The parameters of permutation			The parameters of diffusion				
		$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	kC_0	kP_0
1	Lena	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Not-Pass	Not-Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Not-Pass	Not-Pass
2	Lena	Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Not-Pass	Not-Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Not-Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Not-Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Not-Pass	Not-Pass
3	Lena	Pass	Pass	Pass	Not-Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass
	House	Pass	Pass	Pass	Not-Pass	Pass	Pass	Not-Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 16 (continued)

N_e	Image	$NPCR$ (%) test on ($NPCR^*_{0.05} = 99.569\%$)							
		The parameters of permutation			The parameters of diffusion				
		$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	kC_0	kP_0
	Clock	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Not-Pass	Not-Pass	Not-Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
4	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
5	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 17 Sensitivity of the secret key by means of $UACI$ with various encryption rounds N_e

N_e	Image	$UACI$ (%) test on ($UACI^{*-}_{0.05} = 33.282\%$ and $UACI^{*+}_{0.05} = 33.644\%$							
		The parameters of permutation			The parameters of diffusion				
		$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	kC_0	kP_0
1	Lena	Not-Pass	Not-Pass	Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass
	House	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Not-Pass	Pass	Pass	Not-Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Not-Pass	Not-Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Not-Pass	Pass	Not-Pass	Pass	Pass	Pass	Pass
	On average	Not-Pass	Not-Pass	Pass	Pass	Pass	Pass	Pass	Pass
2	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Not-Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
3	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Not-Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass

Table 17 (continued)

N_e	Image	$UACI$ (%) test on ($UACI^{*-}_{0.05} = 33.282\%$ and $UACI^{*+}_{0.05} = 33.644\%$							
		The parameters of permutation			The parameters of diffusion				
		$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	$x_0^{(2)}$	$x_0^{(1)}$	$\delta_0^{(1)}$	kC_0	kP_0
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Not-Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
4	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
5	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pas
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

$UACI$ pass the tests of statistical hypothesis for $N_e \geq 4$. The average values of $NPCR$ pass the tests for $N_e \geq 4$ while those of $UACI$ meet the critical range for $N_e \geq 2$. It means that a small change in the plaintext results large change in the ciphertexts, or the encryption is highly sensitive to the plaintext for few number of encryption rounds. It is clear that a number of individual values of $NPCR$ and $UACI$ are increased significantly from the first to the second round of encryption. That is due to the last pixel of plain images being modified. In fact, with the scanning order of row by row and left to right, the diffusion will makes the value of more pixels changed if the modified pixel is permuted with the one at lower row number. However, for the second round of encryption and beyond ($N_e \geq 2$), the sensitivities on the plaintexts are equal for every plaintext.

4.4 Comparison with existing methods

In this section, the simulation result of the example employing the proposed design is compared with very recent methods regardless to their structures, plaintext images and the size of images. The simulation results in our example are obtained at $N_e \geq 4$ and $N_p = N_e = 4$ for the comparison. In addition, the simulation results of existing methods are re-interpreted by calculating the average for the context of multiple images, except for those in qualitatively presentation. Table 20 presents the measures for the statistical and the security analyses in which V/D and N/A stand for visual demonstration and not available. The visual demonstration means that the result is considered as a qualitative measure rather than quantitative one. H, V, and D are denoted for the horizontal, vertical, and diagonal directions of pixels in the calculation of correlation of an image. Here, the range is used for either different encryption round in our example or different encryption algorithms in the compared works.

The histogram of encrypted images is measured and its distribution is analyzed by χ^2-test for the homogeneous. All the tests of encrypted images in this example are passed for $N_e \geq 3$ as given in Table 9. It indicates that the gray level of all the encrypted images are with the uniform distribution. Whereas, all the referral works only qualitatively demonstrate by plotting. The information entropy of the encrypted images in this example is comparable with that given in most of referral works. Specifically, it is better than that obtained in [97] and [101]. The correlation of adjacent pixels is computed and presented in most of reports, except for [97], and the result of this example is as good as those in the existing methods in all directions.

In fact, as one of advantage of chaos-based image encryption is that the space of secret key is large, and it can be easily adjusted for the desire of applications. It shows that the space of secret key in the example is as large as that in the referral works, and it is large enough to resist the brute force attack running on the nowadays computer. The sensitivity of secret key is usually considered by means of Cdr, $NPCR$, and $UACI$ for two encrypted images what are produced by using two slightly different values of secret key. $NCPR$ and $UACI$ pass the test of statistical hypothesis with the significance level $\alpha = 0.05$ for $N_e \geq 4$. Also, the range of Cdr in this example are overlapped with that in other referral works. It means that the encryption in this example has as good sensitivity of secret key as that in other works. Similarly, the result of sensitivity of plaintext in this example shows that the encryption is also comparable with those in other referral works.

In summary, it is clear that the exemplar encryption using the proposed design is as good as those in the existing works.

Table 18 Sensitivity of the plaintext by means of $NPCR$ with various encryption rounds N_e

N_e	Image	($NPCR^*_{0.05} = 99.569\%$) $NPCR$ (%) test for the sensitivity of							
		Lena	Cameraman	House	Peppers	Boat	Clock	Black	White
1	Lena	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Cameraman	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	House	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Peppers	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Boat	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Clock	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Black	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	White	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	On average	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
2	Lena	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	White	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass
	On average	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
3	Lena	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Pass	Pass	Not-Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass
	House	Not-Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 18 (continued)

N_e	Image	$(NPCR^*_{0.05} = 99.569\%)$ $NPCR$ (%) test for the sensitivity of							
		Lena	Cameraman	House	Peppers	Boat	Clock	Black	White
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Not-Pass	Pass	Pass	Not-Pass	Not-Pass	Not-Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Not-Pass	Pass	Not-Pass	Not-Pass	Pass	Pass	Pass	Not-Pass
4	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
5	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 19 Sensitivity of the plaintext by means of $UACI$ with various encryption rounds N_e at fixed $N_p = N_d = 4$

N_e	Image	($UACI^{*-}_{0.05} = 33.282\%$ and $UACI^{*+}_{0.05} = 33.644\%$) $UACI$ (%) test for the sensitivity of							
		Lena	Cameraman	House	Peppers	Boat	Clock	Black	White
1	Lena	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Cameraman	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	House	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Peppers	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Boat	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Clock	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	Black	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	White	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
	On average	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass	Not-Pass
2	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Not-Pass	Pass	Not-Pass	Not-Pass	Pass	Not-Pass	Pass	Not-Pass
	Peppers	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Not-Pass	Not-Pass	Not-Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Not-Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
3	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Not-Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 19 (continued)

N_e	Image	$(UACI^{*-}_{0.05} = 33.282\%$ and $UACI^{*+}_{0.05} = 33.644\%)$ $UACI$ (%) test for the sensitivity of							
		Lena	Cameraman	House	Peppers	Boat	Clock	Black	White
4	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
5	Lena	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Cameraman	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	House	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Peppers	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Boat	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Clock	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	Black	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	White	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	On average	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Table 20 Comparison between the result of example at $N_e \geq 4$, $N_p = N_d = 4$ with other existing methods

Method	Statistical analysis	Security analysis				
	Quantitative histogram analysis	Information entropy	Correlation of adjacent pixels	Space of secret key	Sensitivity of secret key	Sensitivity of plaintext (or differential attack)
Proposed method	all completely pass the χ^2-test	≥ 7.996	H : $[-0.00806, 0.00989]$ V : $[-0.00565, 0.006746]$ D : $[-0.00797, 0.00504]$	2^{332}	Cdr : $[99.570, 99.662]$ $NPCR$: Pass $UACI$: Pass	$NPCR$: Pass $UACI$: Pass
[5]	V/D	7.999	$H = -0.00265$ $V = 0.00704$ $D = 0.00323$	2^{512}	V/D	$NPCR$: Pass $UACI$: Pass
[22]	V/D	7.998	$H = 0.00719$ $V = -0.00020$ $D = 0.00056$	$3.2768 * 10^{73}$ ($\approx 2^{245}$)	N/A	$NPCR$: Pass $UACI$: Pass
[40]	V/D	7.999	$H = 0.00187$ $V = 0.00030$ $D = 0.00031$	10^{165} ($\approx 2^{548}$)	Cdr : $[99.591, 99.648]$ $NPCR$: N/A $UACI$: N/A	$NPCR$: Pass $UACI$: Pass
[48]	V/D	7.999	$H = -0.00058$ $V = -0.00029$ $D = -0.00031$	$3.4 * 10^{128}$ ($\approx 2^{427}$)	$NPCR$: Pass $UACI$: Pass	$NPCR$: Pass $UACI$: Pass
[49]	V/D	7.997	$H = -0.00124$ $V = -0.00151$ $D = 0.0005$	$3.4 * 10^{128}$ ($\approx 2^{427}$)	$NPCR$: Pass $UACI$: Pass	$NPCR$: Pass $UACI$: Pass
[14]	V/D	7.996	$H = 0.00133$ $V = 0.00253$ $D = 0.00026$	2^{256}	V/D	$NPCR$: Pass $UACI$: Pass

Table 20 (continued)

Method	Statistical analysis	Security analysis				
	Quantitative histogram analysis	Information entropy	Correlation of adjacent pixels	Space of secret key	Sensitivity of secret key	Sensitivity of plaintext (or differential attack)
[56]	V/D	7.999	$H = -0.0036$ $V = 0.0016$ $D = 0.0058$	2^{512}	V/D $UACI$: Pass	$NPCR$: Pass
[67]	V/D	7.999	$H = -0.02775$ $V = 0.03057$ $D = 0.00385$	$7.37 * 10^{134} (\approx 2^{448})$	V/D	N/A
[93]	V/D	7.999	$H = -0.00364$ $V = 0.00262$ $D = 0.00124$	$10^{195} (\approx 2^{647})$	V/D	$NPCR$: Pass $UACI$: Pass
[95]	V/D	7.999	$H = 0.00092$ $V = -0.00102$ $D = 0.00052$	$10^{135} * (8k)!$ $(\approx 2^{448} * (8k)!)$	N/A $UACI$: Pass	$NPCR$: Pass
[97]	V/D	7.810	N/A	10^{56} $(\approx 2^{186})$	N/A	N/A
[98]	V/D	7.999	$H = 0.00030$ $V = 0.00263$ $D = 0.00412$	10^{60} $(\approx 2^{199})$	$Cdr = 99.778$ $NPCR$: N/A $UACI$: N/A	$NPCR$: Pass $UACI$: Pass
[99]	V/D	7.999	$H = -0.00089$ $V = 0.00189$ $D = 0.00071$	10^{160} $(\approx 2^{531})$	N/A	$NPCR$: Pass $UACI$: Pass
[100]	V/D	7.999	$H = -0.00092$ $V = -0.0005$ $D = 0.00035$	10^{56} $(\approx 2^{186})$	V/D	$NPCR$: Pass $UACI$: Pass
[101]	V/D	7.994	$H = 0.00137$ $V = 0.00117$ $D = -0.00370$	2^{187}	V/D	N/A

5 Discussions and conclusions

The proposed model has some advantages in compared with the existing methods. First, the architecture of the model is based on the permutation-diffusion network. Therefore, the statistical properties of ciphered images and the level of security can be adjusted by either number of permutation, diffusion, and encryption rounds (N_p, N_d, and N_e), or number of iterations applying to chaotic maps in the permutation and diffusion (R_p and R_d). For a specific configuration using the proposed model, the procedure for choosing parameters is as follows. For the permutation, the number of iteration R_p for the PCM in the permutation is chosen at what the statistical properties of output images are saturated while $N_p = 1$. Then, at a chosen value of R_p, the simulation for various number of permutation rounds N_p is performed and the value of N_p is chosen at what the statistical properties of permuted images are with small fluctuation. The same process is carried out to choose the value of R_d and N_d for the diffusion. With chosen value of R_p, R_d, N_p, and N_d, the encryption is simulated to choose number of encryption rounds, N_e, such that the criteria to choose N_e is based on the statistical properties of final ciphered images. The above example has demonstrated these processes, and the simulation results show that, with the perturbed Standard maps chosen for both the permutation and diffusion, the value of encryption are chosen as $R_p = R_d = 4$, $N_p = N_d = 4$, and $N_e \geq 4$.

Second, the PCMs in the proposed design are perturbed on both state variables and control parameters, so its dynamics is non-stationary. It is dependent on the coordinate of pixels in the permutation and on the value of pixels in the diffusion as shown in Figs. 5 and 8, respectively. The proposed design possesses the property of authentication. Therefore, the attack is successful only if the secret key of S bits is fully known. In the example, the numbers of chaotic orbits of PCMs in the permutation and diffusion are 2^{102} and 2^{230}, respectively. The dynamics of PCMs are hopping among such huge numbers of orbits during encryption.

Third, with the fixed-point representation for values of state variables and control parameters, the space of secret key can be larger by increasing the number of bits in the fractional portion. However, the loss is that computation time is longer. Moreover, the value ranges of control parameters as well as state variables are dependent on the patterns of bits chosen to represent fractional numbers, and those can be segmented like control parameters given in Table 1 of the example. In that case, values of parameters are hopping on such the ranges.

Forth, it is clear from Fig. 8 that the proposed design of MIE is with the strong dependence on the content of images by means of the perturbation on the control parameters and state variables during the diffusion. Therefore, it makes the chosen-plaintext and known-plaintext attacks hardly successful in applying to the proposed design.

In fact, there are some extents to the proposed design. First, the proposed design can encrypt k images with k unequal to the power of two. Also, the images can be with different sizes and even with the size of images being unequal to the power of two. In the case of $k \neq 2^i$ (i is an integer), the number of bits in the third portion in Fig. 7b must be $k_1^{(img)} = ceil(log_2 k)$ (the function $ceil(.)$ rounds a number up to the nearest integer). The index of the destination image for the pixel permutation is calculated by using the function $mod(k)$ applying to the value for the index represented by the corresponding sequence of bits in kXY_{des}. For the case of $M \neq 2^j$ or $N \neq 2^t$ (j and t are integers), the coordinate of destination pixels can be identified in the same way.

Second, the proposed design can be used for the RGB images in one of two ways. In one way, each color channel of a RGB image can be dealt as a grayscale image in the encryption.

In another way, each pixel of a RGB image is considered as normal, in which its number of bits are equal to sum of bits from all channels.

Third, there are two options of pixel permutation, i.e. intra-image and inter-image ones. Similarity, for the diffusion, it is carried out on the pixels only within individual images in the example. However, the diffusion in the proposed design can be developed for inter-image one by additional consideration of the destination image for a pixel as the same way in the pixel permutation. In the proposed model, if the encryption using the intra-image permutation together with intra-image diffusion, the encryption performs for each image independently and all images are encrypted in parallel.

In conclusion, the proposed design allows to encrypt more than one image at the same time by thoroughly exploiting a large number of bits generated by chaotic maps. Intuitively, chaotic maps are used more efficiently in compared with single image encryption. In addition, the security of the proposed design is enhanced by using the perturbed chaotic map and high sensitivity on the content of images. Also, it can be adjusted by the use of permutation-diffusion architecture.

Acknowledgements This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2018.06.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Abdelfatah RI (2020) A new fast double-chaotic based image encryption scheme. Multimed Tools Appl 79(1):1241–1259
2. Abdulla AA, Sellahewa H, Jassim SA (2019) Improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images. Multimed Tools Appl 78(13):17799–17823
3. Alvarez G, Amigó JM, Arroyo D, Li S (2011) Lessons learnt from the cryptanalysis of chaos-based ciphers. Springer, Berlin, pp 257–295
4. Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. International Journal of Bifurcation and Chaos 16(08):2129–2151
5. Banik A, Shamsi Z, Laiphrakpam DS (2019) An encryption scheme for securing multiple medical images. Journal of Information Security and Applications 49:102398
6. Bhatnagar G, Wu QMJ (2012) Selective image encryption based on pixels of interest and singular value decomposition. Digital Signal Processing 22(4):648–663
7. Bilal M, Imtiaz S, Abdul W, Ghouzali S, Asif S (2014) Chaos based zero-steganography algorithm. Multimed Tools Appl 72(2):1073–1092
8. Cao LC, Luo YL, Qiu SH, Liu JX (2015) A perturbation method to the Tent map based on Lyapunov exponent and its application. Chin Phys B 24(10):100501
9. Chai X, Gan Z, Zhang M (2017) A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion. Multimed Tools Appl 76:15561–15585
10. Xin Chen J, Liang Zhu Z, Fu C, Yu H (2013) An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism. Opt Express 21(23):27873–27890
11. Chen S, Shu R (2011) Block permutation cipher in chaos with Feistel structure for wireless sensor networks. In: Jin D., Lin S. (eds) Advances in computer science, intelligent system and environment. Springer Berlin Heidelberg, Berlin, pp 391-396
12. Cheng P, Yang H, Wei P (2015) A fast image encryption algorithm based on chaotic map and lookup table. Nonlinear Dynnamics 79:2121–2131
13. Cheng S, Wang L, Ao N, Han Q (2020) A selective video encryption scheme based on coding characteristics. Symmetry 12(3):332

14. Deepak M, Ashwin V, Amutha R (2014) A new multistage multiple image encryption using a combination of chaotic block cipher and iterative fractional fourier transform. In: 2014 First International Conference on Networks Soft Computing (ICNSC2014), pp 360–364
15. Dhivya R, Padmapriya V, Sundararaman R, Rayappan J, Amirtharajan R (2018) Chaos assisted variable bit steganography in transform domain. Electron Lett 54(2):1332–1334
16. El Assad S, Noura H (2014) Generator of chaotic sequences and corresponding generating system. US Patent 8,781,116
17. Fang D, Sun S (2018) A new scheme for image steganography based on hyperchaotic map and DNA sequence. J Inf Hiding Multim Signal Process 9:392–399
18. Fouda JAE, Effa JY, Sabat SL, Ali M (2014) A fast chaotic block cipher for image encryption. Commun Nonlinear Sci Numer Simul 19(3):578–588
19. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. International Journal of Bifurcation and Chaos 08(06):1259–1284
20. Fridrich J, Lisoněk P, Soukal D (2007) On steganographic embedding efficiency. In: Camenisch JL, Collberg CS, Johnson NF, Sallee P (eds) Information hiding. Springer, Berlin, pp 282–296
21. Fu C, jie Chen J, Zou H, hong Meng W, feng Zhan Y, wen Yu Y (2012) A chaos-based digital image encryption scheme with an improved diffusion strategy. Opt Express 20(3):2363–2378
22. ul Haq T, Shah T (2020) Algebra-chaos amalgam and DNA transform based multiple digital image encryption. Journal of Information Security and Applications 102592:54
23. Hilborn R (2001) Chaos And Nonlinear dynamics: an introduction or scientists and engineers 2nd Edition. Oxford University Press, USA
24. Hoang TM (2021) Perturbed chaotic map with varying number of iterations and application in image encryption. In: 2020 IEEE Eighth International conference on communications and electronics (ICCE), pp 413–418
25. Hoang TM, Assad SE (2020) Novel models of image permutation and diffusion based on perturbed digital chaos. Entropy 22(5):548
26. Huang ZJ, Cheng S, Gong LH, Zhou NR (2020) Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform. Opt Lasers Eng 124:105821
27. Kanso A (2012) Steganographic algorithm based on a chaotic map. Commun Nonlinear Sci Numer Simul 17(8):3287–3302
28. Kar N, Mandal K, Bhattacharya B (2018) Improved chaos-based video steganography using DNA alphabets, vol 4. SI: CI & Smart Grid Cyber Security
29. Katz J, Lindell Y (2014) Introduction to modern cryptography, second Edition, 2nd edn. Chapman and hall/CRC, London
30. Kennedy M, Riccardo R, Setti G (2000) Chaotic electronics in telecommunications, 1st edn. CRC, Boca Raton
31. Khalind O, Aziz B (2015) LSB steganography with improved embedding efficiency and undetectability. Comp Sci Info Tech 5(1):89–105
32. Khan J, Ahmad J (2019) Chaos based efficient selective image encryption. Multidim Syst Sign Process 30:943–961
33. Kocarev L, Lian S (2011) Chaos-based Cryptography. Springer, Berlin
34. Li X, Meng X, Wang Y, Yang X, Yin Y, Peng X, He W, Dong G, Chen H (2017) Secret shared multiple-image encryption based on row scanning compressive ghost imaging and phase retrieval in the fresnel domain. Opt Lasers Eng 96:7–16
35. Liu L, Lin J, Miao S, Liu B (2017) A double perturbation method for reducing dynamical degradation of the digital Baker map. International Journal of Bifurcation and Chaos 27(07):1750103
36. Liu L, Liu B, Hu H, Miao S (2018) Reducing the dynamical degradation by bi-coupling digital chaotic maps. International Journal of Bifurcation and Chaos 28(05):1850059
37. Liu L, Miao S (2017) Delay-introducing method to improve the dynamical degradation of a digital chaotic map. Inf Sci 396:1–13
38. Liu X, Song Y, Jiang GP (2019) Hierarchical bit-level image encryption based on chaotic map and Feistel network. Int J Bifurcation Chaos 29(02):1950016
39. Liu Y, Luo Y, Song S, Cao L, Liu J, Harkin J (2017) Counteracting dynamical degradation of digital chaotic Chebyshev map via perturbation. International Journal of Bifurcation and Chaos 27(03):1750033
40. Malik DS, Shah T (2020) Color multiple image encryption scheme based on 3D-chaotic maps. Math Comput Simul 178:646–666
41. Masuda N, Jakimoski G, Aihara K, Kocarev L (2006) Chaotic block ciphers: from theory to practical algorithms. IEEE Transactions on Circuits and Systems I: Regular Papers 53(6):1341–1352
42. Matthews R (1989) On the derivation of a "chaotic" encryption algorithm. Cryptologia 13(1):29–42

43. Mondal B (2020) A secure steganographic scheme based on chaotic map and DNA computing. In: Sharma DK, Balas VE, Son LH, Sharma R, Cengiz K (eds) Micro-electronics and telecommunication engineering. Springer, Singapore, pp 545–554
44. Mousavi M, Sadeghiyan B (2021) A new image encryption scheme with Feistel like structure using chaotic s-box and rubik cube based p-box. Multimedia Tools and Applications 80(9):13157–13177
45. Mukherjee S, Sanyal G (2018) A chaos based image steganographic system. Multimed Tools Appl 77(21):27851–27876
46. Norouzi B, Mirzakuchaki S (2017) An image encryption algorithm based on DNA sequence operations and cellular neural network. Multimedia Tools and Applications 76(11):13681–13701
47. Patel S, Bharath K, Rajesh P, Kumar M (2020) Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique. Multimed Tools Appl 79(43):31739–31757
48. Patro KAK, Acharya B (2018) Secure multi-level permutation operation based multiple colour image encryption. Journal of Information Security and Applications 40:111–133
49. Patro KAK, Acharya B (2020) A novel multi-dimensional multiple image encryption technique. Multimed Tools Appl 79(19):12959–12994
50. Patro KAK, Soni A, Netam PK, Acharya B (2020) Multiple grayscale image encryption using cross-coupled chaotic maps. Journal of Information Security and Applications 52:102470
51. Peng J, Lei L, Han Q, Jia R (2014) A chaos-based block cipher with Feistel structure. In: 2014 IEEE 13Th International conference on cognitive informatics and cognitive computing, pp. 343–348
52. Peng J, Zhang D (2009) Image encryption and chaotic cellular neural network. In: Machine learning in cyber trust: security, privacy, and reliability. Springer, Boston, pp 183-213
53. Rehman AU, Khan JS, Ahmad J, Hwang SO (2016) A new image encryption scheme based on dynamic S-Boxes and chaotic maps. 3D Research 7(1):1–8
54. Sheela S, Sathyanarayana SV (2022) Generation of chaotic random binary sequences for cryptographic applications. Concurrency and Computation: Practice and Experience 34:e6497
55. Saeed M (2013) A new technique based on chaotic steganography and encryption text in DCT domain for color image. Journal of Engineering Science and Technology 8:508–520
56. Sahasrabuddhe A, Laiphrakpam DS (2021) Multiple images encryption based on 3D scrambling and hyper-chaotic system. Inf Sci 550:252–267
57. Saidi M, Hermassi H, Rhouma R, Belghith S (2017) A new adaptive image steganography scheme based on DCT and chaotic map. Multimed Tools Appl 76(11):13493–13510
58. Sayed WS, Tolba MF, Radwan AG, Abd-El-Hafiz SK, Soliman AM (2018) Security and efficiency of Feistel networks versus discrete chaos for lightweight speech encryption. In: 2018 30Th international conference on microelectronics (ICM), pp 92–95
59. Shannon CE (1949) Communication theory of secrecy systems. Bell System Technical Journal 28(4):656–715
60. Sharif A, Mollaeefar M, Nazari M (2017) A novel method for digital image steganography based on a new three-dimensional chaotic map. Multimed Tools Appl 76(6):7849–7867
61. Shu-Bo L, Jing S, Zheng-Quan X, Jin-Shuo L (2009) Digital chaotic sequence generator based on coupled chaotic systems. Chin Phys B 18(12):5219–5227
62. Shuai CHEN, ZHONG XX (2007) Confidential communication through chaos encryption in wireless sensor network. Journal of China University of Mining and Technology 17(2):258–261
63. Situ G, Zhang J (2006) Position multiplexing for multiple-image encryption. J Opt A Pure Appl Opt 8(5):391–397
64. Som S, Kotal A, Mitra A, Palit S, Chaudhuri BB (2014) A chaos based partial image encryption scheme. In: 2014 2Nd international conference on business and information management (ICBIM), pp 58–63
65. Stavroulakis P (2006) Chaos applications in telecommunications. CRC, Boca Raton
66. Tang G, Liao X (2005) A method for designing dynamical s-boxes based on discretized chaotic map. Chaos, Solitons Fractals 23(5):1901–1909
67. Tang Z, Song J, Zhang X, Sun R (2016) Multiple-image encryption with bit-plane decomposition and chaotic maps. Opt Lasers Eng 80:1–11
68. Tao S, Ruli W, Yixun Y (1998) Perturbance-based algorithm to expand cycle length of chaotic key stream. Electron Lett 34(1):873–874
69. Thenmozhi S, Chandrasekaran M (2013) A novel technique for image steganography using nonlinear chaotic map. In: 2013 7Th International conference on intelligent systems and control (ISCO), pp 307–311
70. Tong XJ, Zhang M, Wang Z, Liu Y, Xu H, Ma J (2015) A fast encryption algorithm of color image based on four-dimensional chaotic system. J Vis Commun Image Represent 33:219–234
71. Valandar MY, Ayubi P, Barani MJ (2017) A new transform domain steganography based on modified logistic chaotic map for color images. Journal of Information Security and Applications 34:142–151

72. Valandar MY, Barani MJ, Ayubi P, Aghazadeh M (2019) An integer wavelet transform image steganography method based on 3D sine chaotic map. Multimed Tools Appl 78(8):9971–9989
73. Wai M, Tam FCM, Lau CKT (2007) Digital communications with chaos: multiple access techniques and performance. Elsevier, Amsterdam
74. Wang J, Ding Q (2018) Dynamic rounds chaotic block cipher based on keyword abstract extraction. Entropy 20(9):693
75. Wang L, Cheng H (2019) Pseudo-random number generator based on logistic chaotic system. 21(10):960
76. Wang Q, Zhang Q, Wei X (2010) Image encryption algorithm based on DNA biological properties and chaotic systems. In: 2010 IEEE Fifth international conference on bio-inspired computing: theories and applications (BIC-TA), pp. 132–136
77. Wang X, Lin S, Li Y (2021) Bit-level image encryption algorithm based on BP neural network and gray code. Multimed Tools Appl 80(8):11655–11670
78. Wang X, Zhao D (2011) Multiple-image encryption based on nonlinear amplitude-truncation and phase-truncation in fourier domain. Opt Commun 284(1):148–152
79. Wang Y, Quan C, Tay C (2014) Nonlinear multiple-image encryption based on mixture retrieval algorithm in fresnel domain. Opt Commun 330:91–98
80. Wang Y, Wong KW, Liao X, Chen G (2011) A new chaos-based fast image encryption algorithm. Appl Soft Comput 11(1):514–522
81. Weng H, Zhang C, Chen P, Chen R, Xu J, Liao Y, Liang Z, Shen D, Zhou L, Ke J (2021) A quantum chaotic image cryptosystem and its application in IoT secure communication. IEEE Access 9:20481–20492
82. Wong KW, Kwok BSH, Law WS (2008) A fast image encryption scheme based on chaotic Standard map. Phys Lett A 372(15):2645–2652
83. Wu X, Wang D, Kurths J, Kan H (2016) A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system. Inf Sci 349-350:137–153
84. Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. In: Cyber journals: multidisciplinary journals in science and technology, journal of selected areas in telecommunications (JSAT)
85. Xiang H, Liu L (2020) An improved digital Logistic map and its application in image encryption. Multimed Tools Appl 79(41):30329–30355
86. Xiang T, Wong KW, Liao X (2007) Selective image encryption using a spatiotemporal chaotic system. Chaos 17:023115
87. Yang F, Mou J, Sun K (2020) Lossless image compression-encryption algorithm based on BP neural network and chaotic system. Multimed Tools Appl 79:19963
88. Yang H, Wong KW, Liao X, Zhang W, Wei P (2010) A fast image encryption and authentication scheme based on chaotic maps. Commun Nonlinear Sci Numer Simul 15(11):3507–3517
89. Yao W, Wu F, Zhang X, Zheng Z, Wang Z, Wang W, Qiu W (2016) A fast color image encryption algorithm using 4-pixel Feistel structure. PLOS ONE 11(11):e0165937
90. Yao W, Zhang X, Zheng Z, Qiu W (2015) A colour image encryption algorithm using 4-pixel Feistel structure and multiple chaotic systems. Nonlinear Dyn 81(1):151–168
91. Yavuz E (2019) A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme. Optics & Laser Technology 114:224–239
92. Ye G, Huang X (2016) A secure image encryption algorithm based on chaotic maps and SHA-3. Security and Communication Networks 9(13):2015–2023. https://doi.org/10.1002/sec.1458
93. Zarebnia M, Pakmanesh H, Parvaz R (2019) A fast multiple-image encryption algorithm based on hybrid chaotic systems for gray scale images. Optik 179:761–773
94. Zhang J, Huo D (2019) Image encryption algorithm based on quantum chaotic map and DNA coding. Multimedia Tools and Applications 78(11):15605–15621
95. Zhang L, Zhang X (2020) Multiple-image encryption algorithm based on bit planes and chaos. Multimedia Tools and Applications 79(29):20753–20771
96. Zhang S, Liu L (2021) A novel image encryption algorithm based on SPWLCM and DNA coding. Math Comput Simul 190:723–744
97. Zhang X, Wang X (2017) Multiple-image encryption algorithm based on mixed image element and chaos. Computers & Electrical Engineering 62:401–413
98. Zhang X, Wang X (2017) Multiple-image encryption algorithm based on mixed image element and permutation. Opt Lasers Eng 92:6–16
99. Zhang X, Wang X (2018) Multiple-image encryption algorithm based on the 3D permutation model and chaotic system. Symmetry 10:660

100. Zhang X, Wang X (2019) Multiple-image encryption algorithm based on DNA encoding and chaotic system. Multimedia Tools and Applications 78(6):7841–7869
101. Zhou N, Yan X, Liang H (2018) Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system. Quantum Inf Process 17:338
102. Zhou NR, Huang LX, Gong LH, Zeng QW (2020) Novel quantum image compression and encryption algorithm based on DQWT and 3D hyper-chaotic henon map. Quantum Inf Process 19(9):284
103. Zhu H, Tong X, Wang Z, Ma J (2020) A novel method of dynamic s-box design based on combined chaotic map and fitness function. Multimedia Tools and Applications 79(17):12329–12347
104. Zhu S, Wang G, Zhu C (2019) A secure and fast image encryption scheme based on double chaotic s-boxes. Entropy 21:790