Signal Processing 144 (2018) 134–144

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Medical image encryption using high-speed scrambling and pixel adaptive diffusion

Zhongyun Hua[a], Shuang Yi[b], Yicong Zhou[b,*]

[a] School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China
[b] Department of Computer and Information Science, University of Macau, Macau 999078, China

ARTICLE INFO

Article history:
Received 4 March 2017
Revised 16 September 2017
Accepted 3 October 2017
Available online 4 October 2017

Keywords:
Cryptosystem
Image encryption
Medical image
High-speed scrambling
Pixel adaptive diffusion

ABSTRACT

This paper presents a new encryption scheme of protecting medical images. It has high efficiency and shows robustness of defending some impulse noise and data loss. First, some random data are inserted into surroundings of the image. Then, two rounds of high-speed scrambling and pixel adaptive diffusion are performed to randomly shuffle neighboring pixels and spread these inserted random data over the entire image. The proposed encryption scheme can be directly applied to medical images with any representation format. We provide two kinds of operations to implement the pixel adaptive diffusion: bitwise XOR and modulo arithmetic. The former has high efficiency in hardware platforms while the latter can achieve fast speed in software platforms. Simulations and evaluations show that both encryption schemes using bitwise XOR and modulo arithmetic have high security levels, can achieve much faster speeds, and can better adapt to impulse noise and data loss interference than several typical and state-of-the-art encryption schemes.

1. Introduction

1.1. Background

In modern hospitals, digital medical images play more and more important roles in diagnosing and treating diseases and thus they attracts increasing attentions [1,2]. Generally speaking, these medical images may involve a lot of privacy of patients and some of them are very confidential and sensitive. Disastrous accidents may occur if these private images are stolen, viewed or used by unauthorized accesses. For example, a hacker or a malicious database administrator may use the unauthorized images for their personal benefits, such as medical marketing and fraudulent insurance claims, which may cause life-threatening risks. Therefore, protecting medical images is quite important.

Up to now, many technologies have been developed to protect all kinds of images such as medical images. Among these technologies, encryption is the most intuitionistic and effective way that transforms images into unrecognized ones [3–8]. Only with the correct key, one can recover the original image [9–13]. Recently, many image encryption schemes were proposed that can be used to protect medical images with high security level [14–17]. Here are some examples. In [18], the authors have presented an encryption scheme using quaternion to protect DICOM image, where DICOM is a developed standard to facilitate the secure and reliable communication of medical images among different medical imaging equipment [19]. In [20], Zhang et al. designed a novel image encryption scheme using rotation matrix bit-level permutation and block diffusion. In [21], Zhou et al. presented a new security scheme that can simultaneously encrypt and compress images using hyper-chaotic system and 2D compressive sensing. In [22], Zhang et al. proposed a medical image encryption and compression scheme using compressive sensing and pixel permutation approach. It can also simultaneously encrypt and compress the medial images.

Although all kinds of encryption schemes were developed to protect medical images, to the best of our knowledge, some of them have weaknesses in different aspects. First, with the improvement of computer computation ability and development of cryptanalysis theory, some developed encryption schemes have the risks of being broken [23–25]. For example, as the chaos theory has the properties of unpredictability and ergodicity [26], they are widely used to develop image encryption schemes to protect medical images. But due to the performance limitation of the used chaotic systems, some chaos-based cryptosystems have been proved of owning low security levels [27–29].

Secondly, to contain more detail information for a better clinical diagnosis, a large number of bits are usually used to repre-

* Corresponding author.
E-mail addresses: huazhongyun@hit.edu.cn (Z. Hua), yicongzhou@umac.mo (Y. Zhou).

https://doi.org/10.1016/j.sigpro.2017.10.004
0165-1684/

Table 1
The common representation formats of digital medical image.

Formats	Header	Extension	Data Types
ANALYZE	348 byte binary stream	.img or .hdr	Unsigned integer (8-bit), signed integer (16-, 32-bit), float (32-,64-bit), complex (64-bit)
NIFTI	348 or 352 byte binary stream	.nll	Signed and unsigned integer (from 8- to 64-bit), float (from 32- to 128-bit), complex (from 64- to 256-bit)
MINC	Extensible binary stream	.mnc	Signed and unsigned integer (from 8- to 32-bit), float (32-, 64-bit), complex (32-, 64-bit)
DICOM	Variable length binary stream	.dcm	Signed and unsigned integer, (8-, 16-bit; 32-bit only allowed for radiotherapy dose)

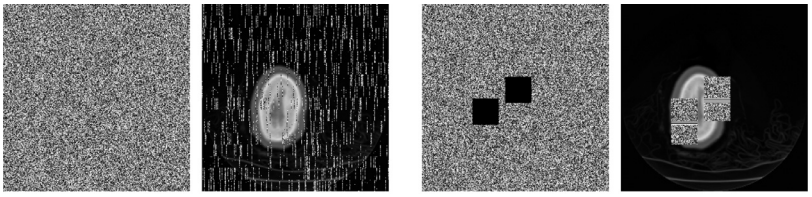

Fig. 1. The demonstration of AES-CFB with noise and data loss. (a) Cipher-image with salt-and-pepper noise; (b) decrypted result of (a); (c) cipher-image with data loss; (d) decrypted result of (c).

sent a pixel in medical image, while we commonly use 8 bits to represent a pixel of gray-scale image and 24 bits to represent a pixel of color image. Table 1 shows some common formats of digital medical image [30]. It shows that a pixel of medical image may be represented by at most 256 bits. One the other hand, to clearly reflect the actual situations of organs and tissues, hundreds, even thousands of two-dimensional images with high resolution may be collected in a diagnosis. Therefore, high efficiency is strongly required in protecting medical images. The existing encryption schemes can be divided into two classes: fixed bit length encryption schemes [18,31–33] and variable bit length encryption schemes [34–36]. The former encrypts images in fixed bit length and can not be directly applied to images with different representation formats. The latter can encrypt images in pixel level. However, this kind of existing encryption schemes usually has complicated structure, which may greatly increase the computation time.

Besides, blurring or data loss may happen when information is stored in physical devices or corrupted by malicious accesses. For some existing encryption schemes, if blurring or data loss happens in the cipher-images encrypted by these schemes, the corresponding decrypted images may loss a large amount of information and result in a low visualized quality. Fig. 1 demonstrates the decrypted results, whose associated cipher-images encrypted by AES with Cipher Feedback mode (AES-CFB) are blurred or loss some data. From the figure, we can observe that when the cipher-image is blurred by salt-and-pepper noise, its decrypted result also has noise, when the cipher-image losses some data, the information of the associated area in decrypted result is totally lost.

1.2. Contributions

To address the problems of existing encryption schemes of protecting medical images, we propose an image encryption scheme. It adopts the well-known substitution-permutation network. First, random values are inserted into surroundings of the image. Then, two rounds of high-speed scrambling and pixel adaptive diffusion are performed to randomly shuffle pixel positions and spread the inserted values over entire image. The proposed encryption scheme can directly encrypt medical image with any representation format. The more bits to present every pixel, the faster speed can be achieved. To implement the pixel adaptive diffusion operation in the proposed encryption scheme, we provide two operations: bit-wise XOR (BX) and modulo arithmetic (MA). They have high efficiency in hardware platforms and software platforms, respectively. The main contributions and novelties of this work can be summarized as follows: (1) We propose an encryption scheme to protect medical images with high efficiency and robustness. It has a high security level as it can encrypt an identical image into different cipher-images, even when using the same secret key; (2) We provide two implementations, which have high efficiency in hardware platforms and software platforms, respectively; (3) Simulation results show that both implementations can encrypt medical images with different representation formats into noise-like images; (4) Evaluations and comparisons show that the proposed encryption scheme has high security level, faster speeds, and better robustness of defending impulse noise and data loss than several typical encryption schemes.

The rest of this paper is organized as follows. Section 2 describes the medical image encryption scheme; Section 3 provide the simulation results of the proposed encryption scheme and discusses its properties; Section 4 evaluates the performance of the proposed encryption scheme and compares it with several other classical schemes; Section 5 concludes the paper.

2. Medical image encryption scheme

This section describes the encryption scheme. Its structure is shown in Fig. 2, in which the secret key $\mathbf{K}$ has length of 256 bits. The key distribution is to decompose the secret key to obtain subkeys for the scrambling and diffusion. The scrambling and diffusion operations are to randomly shuffle pixel positions and change pixel values, respectively. Random data insertion is to add random values to surroundings of the image. This can ensure that the encrypted result of each execution of our encryption scheme is unique and different, even when using a same secure key to encrypt an identical image several times. Benefited from this property, two rounds of encryption processes can guarantee a high security level. Users can also use three or more rounds to further enhance the security level. Taking into account the efficiency, our proposed encryption scheme uses two rounds. The encryption is represented as $\mathbf{C} = \mathrm{Enc}(\mathbf{P}, \mathbf{K})$ and the decryption is denoted as $\mathbf{P} = \mathrm{Dec}(\mathbf{C}, \mathbf{K})$.

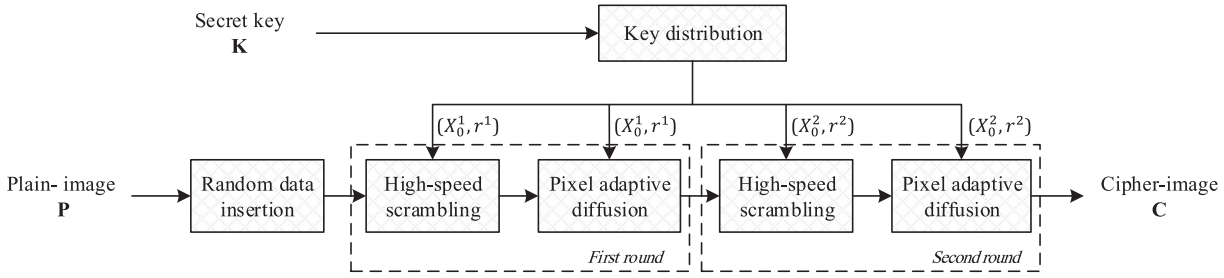

Fig. 2. Structure of the proposed medical image encryption scheme.

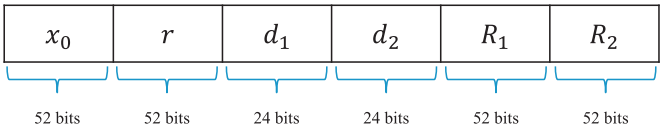

Fig. 3. Structure of the secret key.

2.1. Key distribution

The secure key **K** is used to generate pseudo-random numbers for the high-speed scrambling and pixel adaptive diffusion. The Logistic-Sine system (LSS) proposed in [37] is used to generate pseudo-random numbers and is denoted as LSS-PRNG. It is defined as

$$X_{n+1} = (rX_n(1 - X_n) + (4 - r)\sin(\pi X_n)/4) \mod 1, \quad (1)$$

where the control parameter $r \in [0, 4]$ and X_n is the iteration variable and $X_n \in (0, 1)$. Given initial state (X_0, r), a determined pseudo-random sequence $\{X_i | i = 1, 2, \cdots\}$ can be generated. The secure key **K** is used to generate the initial states for the two rounds of encryption and its structure is shown in Fig. 3. The variables x_0, r, R_1 and R_2 are float numbers that can be converted from a 52-bit stream by

$$\mathrm{FN} = \sum_{i=1}^{52} \mathrm{Bin}_i \times 2^{-i}.$$

Two integers d_1 and d_2 can be obtained from a 24-bit stream by

$$\mathrm{IN} = \sum_{i=1}^{24} \mathrm{Bin}_i \times 2^{i-1}.$$

Then, the initial states for the two encryption rounds can be obtained as follows,

$$\begin{cases} X_0^i = d_i \times (x_0 + R_i) \mod 1, \\ r^i = d_i \times (r + R_i) \mod 4, \end{cases} \quad (2)$$

where $i = \{1, 2\}$. Using the initial states (X_0^1, r^1) and (X_0^2, r^2), the LSS-PRNG can generate pseudo-random sequences for the two rounds of high-speed scrambling and pixel adaptive diffusion.

2.2. Random data insertion

In our proposed encryption scheme, we insert some random data to the surroundings of the image. These data are randomly generated and can be regarded as noise. Thus, in each execution, these inserted data are different and even the system designer and users can't know their values. Suppose the plain-image **P** is of size $H \times W$, two random vectors, **R** of size $2 \times W$ and **O** of size $(H + 2) \times 2$, are randomly generated. Their values are represented by the same data format as the pixels of **P**. The two rows of **R** are inserted into the top and bottom of **P**, and the two columns of **O** are inserted into the left and right of **P**. Fig. 4 demonstrates the operation of random data insertion. In the whole encryption process, random data insertion is performed only one time. These inserted data are random and different in each encryption, and they can affect the whole pixels after the high-speed scrambling and pixel adaptive diffusion. This makes each encrypted result different, even when using the same secret key to encrypt an identical image. Because the scrambling and diffusion operations are reversible, using the correct key, the decryption process can completely recover the original image without these inserted data.

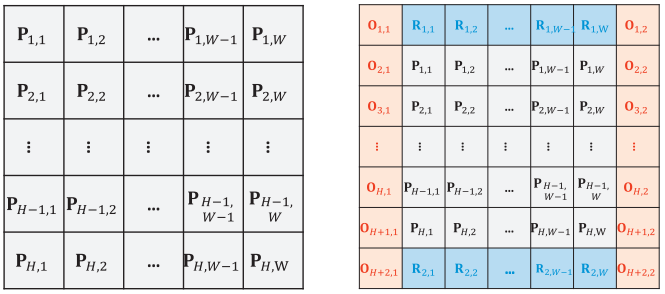

Fig. 4. Demonstration of random data insertion: (a) plain-image **P**; (b) operation result.

2.3. High-speed scrambling

High-speed scrambling is to fast shuffle pixel positions within the image. It changes row and column positions of pixels simultaneously, and thus can efficiently reduce the strong correlation between neighboring pixels. First, a matrix **S** is generated by the LSS-PRNG with initial state (X_0^i, r^i) ($i = 1$ in the first round and $i = 2$ in the second round). Then, the pixels can be totally shuffled with **S**. Algorithm 1 presents the pseudocode of the overall high-speed scrambling.

2.3.1. Generation of scrambling matrix S

We suppose the image to be encrypted is of size $M \times N$. Firstly, generate two random vectors, **A** of length M and **B** of length N, using the LSS-PRNG with the initial state (X_0^i, r^i) ($i = 1$ in the first round and $i = 2$ in the second round) derived from the secret key. Secondly, sort **A** and **B**, respectively, and obtain two index vectors, **I** and **J**. Thirdly, initial a matrix of size $M \times N$, and assign each row of the matrix as **J**. Finally, shift each row of the matrix using each element of **I** and obtain **S**. Fig. 5 shows an example of generating **S** with $M = 4$, $N = 4$.

2.3.2. Image scrambling using S

Using the values in each column of **S**, the algorithm shifts the image pixels from different rows and columns simultaneously. The scrambling operation can be demonstrated using a bijection g. Suppose (r, c) denotes position of pixel in **P** and (m, n) represents that

Algorithm 1 High-speed scrambling.

Input: Image $\mathbf{P}$ of size $M \times N$ and initial state (X_0^i, r^i).
Output: $\mathbf{T}$.

$\mathbf{R} = \text{LSS-PRNG}(X_0^i, r^i)$, where $\mathbf{R} \in \mathbb{Z}^{1\times(M+N)}$;
$\mathbf{A} = \mathbf{R}_{1:M}$, $\mathbf{B} = \mathbf{R}_{(M+1):(M+N)}$;
Sort $\mathbf{A}$ and get sorted $\mathbf{A}'$, where $\mathbf{A}' = \mathbf{A}_\mathbf{I}$;
Sort $\mathbf{B}$ and get sorted $\mathbf{B}'$, where $\mathbf{B}' = \mathbf{B}_\mathbf{J}$;
Set $\mathbf{S} \in \mathbb{N}^{M\times N}$, $\mathbf{T} \in \mathbb{N}^{M\times N}$;
for $i = 1$ to M **do**
  for $j = 1$ to N **do**
    $m = ((j - \mathbf{I}(i) - 1) \mod N) + 1$;
    $\mathbf{S}_{i,m} = \mathbf{J}_j$;
  end for
end for
for $j = 1$ to N **do**
  for $i = 1$ to M **do**
    $r = i$, $c = \mathbf{S}_{i,j}$;
    $m = ((r - \mathbf{S}_{1,j} - 1) \mod M) + 1$, $n = \mathbf{S}_{m,j}$;
    $\mathbf{T}_{m,n} = \mathbf{P}_{r,c}$;
  end for
end for

in scrambled result $\mathbf{T}$, the bijection g from (r, c) to (m, n) is described as

$$g_{(r,c)\longrightarrow(m,n)} : \begin{cases} m = ((r - \mathbf{S}_{1,j} - 1) \mod M) + 1, \\ n = \mathbf{S}_{m,j}, \end{cases} \tag{3}$$

where j satisfies $\mathbf{S}_{r,j} = c$. The inverse scrambling in the decryption process is to perform the inverse bijection g' from (m, n) to (r, c), which is defined as

$$g'_{(m,n)\longrightarrow(r,c)} : \begin{cases} r = ((m + \mathbf{S}_{1,j} - 1) \mod M) + 1, \\ c = \mathbf{S}_{r,j}, \end{cases} \tag{4}$$

where j satisfies $\mathbf{S}_{m,j} = n$. The detail procedure can be described as follows:

- *Step 1*: Initialize column index $j = 1$.
- *Step 2*: Find the pixels of $\mathbf{P}$ with positions $\{(1, \mathbf{S}_{1,j}), (2, \mathbf{S}_{2,j}), \cdots, (M, \mathbf{S}_{M,j})\}$;
- *Step 3*: Connect these pixels into a circle and shift them $\mathbf{S}_{1,j}$ cells to the upper direction.
- *Step 4*: Repeat *Step 2* through *Step 3* $M - 1$ times for $j = 2 \sim N$.

Fig. 6 gives a numerical example for the image of size 4×4. Fig. 6(b) shows the pixels in original image $\mathbf{P}$ and their distributions in its scrambled result $\mathbf{T}$, and Fig. 6(c) depicts the one-to-one position mapping between pixels of $\mathbf{P}$ and those of $\mathbf{T}$. The detailed operation can be described as follows:

- As the 1st column of $\mathbf{S}$ is $\{4, 1, 2, 3\}^T$, find the pixels of $\mathbf{P}$ in gray with positions $\{(1, 4), (2, 1), (3, 2), (4, 3)\}$, and shift them $\mathbf{S}_{1,1} = 4$ cells to the upper direction. Then, $\mathbf{T}_{1,4} = \mathbf{P}_{1,4}$, $\mathbf{T}_{2,1} = \mathbf{P}_{2,1}$, $\mathbf{T}_{3,2} = \mathbf{P}_{3,2}$ and $\mathbf{T}_{4,3} = \mathbf{P}_{4,3}$;
- As the 2nd column of $\mathbf{S}$ is $\{2, 4, 3, 1\}^T$, find the pixels of $\mathbf{P}$ in cyan with positions $\{(1, 2), (2, 4), (3, 3), (4, 1)\}$, and shift them $\mathbf{S}_{1,2} = 2$ cells to the upper direction. Then, $\mathbf{T}_{1,2} = \mathbf{P}_{3,3}$, $\mathbf{T}_{2,4} = \mathbf{P}_{4,1}$, $\mathbf{T}_{3,3} = \mathbf{P}_{1,2}$ and $\mathbf{T}_{4,1} = \mathbf{P}_{2,4}$;
- As the 3-rd column of $\mathbf{S}$ is $\{3, 2, 1, 4\}^T$, find the pixels of $\mathbf{P}$ in khaki with positions $\{(1, 3), (2, 2), (3, 1), (4, 4)\}$, and shift them $\mathbf{S}_{1,3} = 3$ cells to the upper direction. Then, $\mathbf{T}_{1,3} = \mathbf{P}_{4,4}$, $\mathbf{T}_{2,2} = \mathbf{P}_{1,3}$, $\mathbf{T}_{3,1} = \mathbf{P}_{2,2}$ and $\mathbf{T}_{4,4} = \mathbf{P}_{3,1}$;
- As the 4th column of $\mathbf{S}$ is $\{1, 3, 4, 2\}^T$, find the pixels of $\mathbf{P}$ in white with positions $\{(1, 1), (2, 3), (3, 4), (4, 2)\}$, and shift them $\mathbf{S}_{1,4} = 1$ cell to the upper direction. Then, $\mathbf{T}_{1,1} = \mathbf{P}_{2,3}$, $\mathbf{T}_{2,3} = \mathbf{P}_{3,4}$, $\mathbf{T}_{3,4} = \mathbf{P}_{4,2}$ and $\mathbf{T}_{4,2} = \mathbf{P}_{1,1}$.

2.4. Pixel adaptive diffusion

Pixel adaptive diffusion is to spread little change of plain-image over all the pixels in cipher-image. It is performed using the previous pixel and randomly generated value to change the current pixel. The operation order is shown as Fig. 7.

To adapt to the software and hardware environments, respectively, we provide two implementation operations: bitwise XOR (BX) and modulo arithmetic (MA). The former has a high computation efficiency in hardware environment while the latter can achieve a fast speed in software environment. The proposed encryption scheme using BX is called MIE-BX and that using MA is termed as MIE-MA. Pixel adaptive diffusion using BX is defined as

$$\mathbf{C}_{i,j} = \begin{cases} \mathbf{T}_{i,j} \oplus \mathbf{T}_{M,N} \oplus \mathbf{Q}_{i,j}, & \text{if } i = 1, j = 1, \\ \mathbf{T}_{i,j} \oplus \mathbf{C}_{M,j-1} \oplus \mathbf{Q}_{i,j}, & \text{if } i = 1, j \neq 1, \\ \mathbf{T}_{i,j} \oplus \mathbf{C}_{i-1,j} \oplus \mathbf{Q}_{i,j}, & \text{if } i \neq 1, \end{cases} \tag{5}$$

where $\oplus$ denotes the BX operation. $\mathbf{Q}$ is a random matrix generated by LSS-PRNG with the initial state (X_0^i, r^i) ($i = 1$ in the first round and $i = 2$ in the second round). It has the same size and its elements are represented as the same data format as the pixels in $\mathbf{T}$. Pixel adaptive diffusion using MA is defined as

$$\mathbf{C}_{i,j} = \begin{cases} (\mathbf{T}_{i,j} + \mathbf{T}_{M,N} + \mathbf{Q}_{i,j}) \mod F, & \text{if } i = 1, j = 1, \\ (\mathbf{T}_{i,j} + \mathbf{C}_{M,j-1} + \mathbf{Q}_{i,j}) \mod F, & \text{if } i = 1, j \neq 1, \\ (\mathbf{T}_{i,j} + \mathbf{C}_{i-1,j} + \mathbf{Q}_{i,j}) \mod F, & \text{if } i \neq 1, \end{cases} \tag{6}$$

where F denotes the number of intensity levels, e.g. $F = 256$ if a pixel is represented by 8 bits.

In the decryption process, the inverse operation of Eq. (5) is defined as

$$\mathbf{T}_{i,j} = \begin{cases} \mathbf{C}_{i,j} \oplus \mathbf{T}_{M,N} \oplus \mathbf{Q}_{i,j}, & \text{if } i = 1, j = 1, \\ \mathbf{C}_{i,j} \oplus \mathbf{C}_{M,j-1} \oplus \mathbf{Q}_{i,j}, & \text{if } i = 1, j \neq 1, \\ \mathbf{C}_{i,j} \oplus \mathbf{C}_{i-1,j} \oplus \mathbf{Q}_{i,j}, & \text{if } i \neq 1, \end{cases} \tag{7}$$

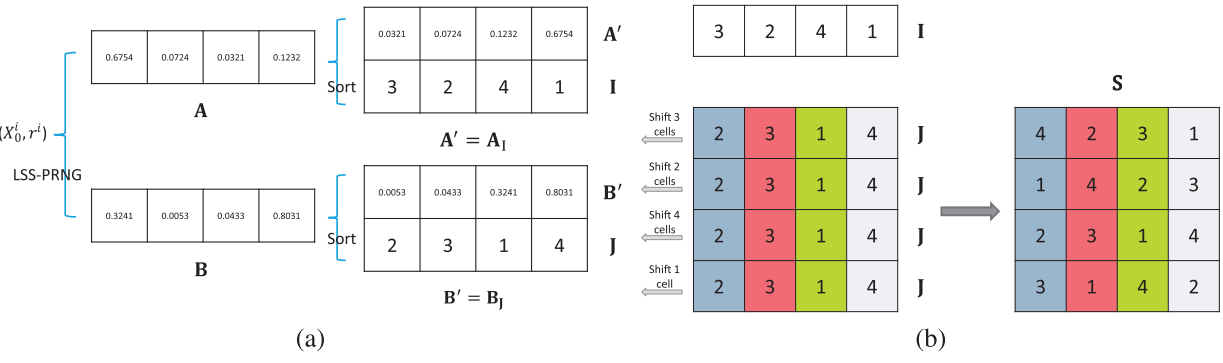

Fig. 5. An example of generating scrambling box $\mathbf{S}$: (a) producing two index vectors $\mathbf{I}$ and $\mathbf{J}$ using sub-key; (b) generating $\mathbf{S}$ using $\mathbf{I}$ and $\mathbf{J}$.

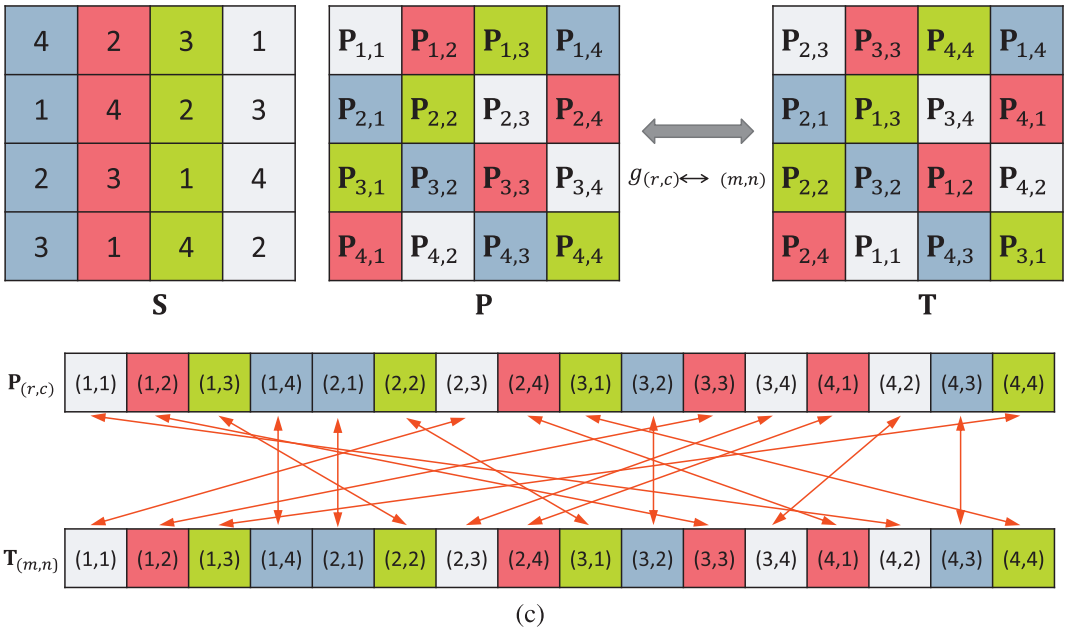

Fig. 6. A number example of high-speed scrambling: (a) scrambling box **S**; (b) pixels in original image **P** and their distribution in scrambled result **T**; (c) one-to-one position mapping between pixels in **P** and **T**.

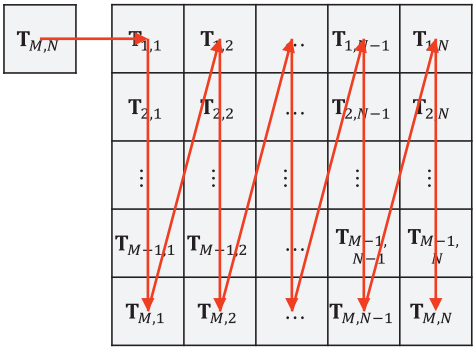

Fig. 7. Pixel adaptive diffusion order.

and the inverse operation of Eq. (6) is defined by

$$\mathbf{T}_{i,j} = \begin{cases} (\mathbf{C}_{i,j} - \mathbf{T}_{M,N} - \mathbf{Q}_{i,j}) \mod F, & \text{if } i = 1, j = 1, \\ (\mathbf{C}_{i,j} - \mathbf{C}_{M,j-1} - \mathbf{Q}_{i,j}) \mod F, & \text{if } i = 1, j \neq 1, \\ (\mathbf{C}_{i,j} - \mathbf{C}_{i-1,j} - \mathbf{Q}_{i,j}) \mod F, & \text{if } i \neq 1. \end{cases} \tag{8}$$

The operation order in decryption process is opposite to that in encryption process. Theoretically, MIE-BX can encrypt images of any size. But in practical applications, due to the storage limitation of hardware implementation, when the image size is too large and the required storage exceeds the storage of hardware implementation, MIE-BX can't directly encrypt the image. In this case, MIE-BX first divides the image into image blocks with smaller size and then encrypts each of these blocks one by one.

3. Simulation results and discussions

This section provides the simulation results of both MIE-BX and MIE-MA in the MATLAB software and discusses their properties.

3.1. Simulation results

Both MIE-BX and MIE-MA can be directly applied to images with any representation format. Here, we use medical images with different representation formats to do the test. Figs. 8 and 9 show the encryption procedures of MIE-BX and MIE-MA for 8-bit, 16-bit and 24-bit medical images, respectively. For the 16-bit medical images, we linearly transform their pixel values into range [0, 255]. For the histograms of 24-bit medical images, we first divide a 24-bit medical image into three 8-bit images and then calculate the histograms of these 8-bit images. One can observe from the figures that all the plain-images have the histograms with many patterns, but their corresponding cipher-images encrypted by MIE-BX and MIE-MA have uniformly distributed pixels. Moreover, we have calculated information entropies of these plain-images and cipher-images and their results are listed in Table 2. The information entropy is a most widely used test to measure the randomness of data sequence and can be defined as

$$H = -\sum_{i=1}^{L} (p(i) \log_2 p(i)), \tag{9}$$

where L is the number of possible values, $p(i)$ is the probability of ith possible value. When all the possible values have the same probabilities, the data sequence can achieve the maximum value and $H_{\max} = \log_2 L$. Our experiments use 256 bins for all the 8-bit, 16-bit and 24-bits images, and thus $H_{\max} = \log_2 256 = 8$. An image with a bigger information entropy means that its pixels are more uniform. From Table 2, one can observe that all the cipher-images encrypted by both MIE-BX and MIE-MA can achieve information entropies that close to the maximum, which means that they have high randomness.

3.2. Discussions

In MIE-BX and MIE-MA, high-speed scrambling can quickly separate neighboring pixels, random values are inserted into surroundings of the image and these values are spread over entire im-

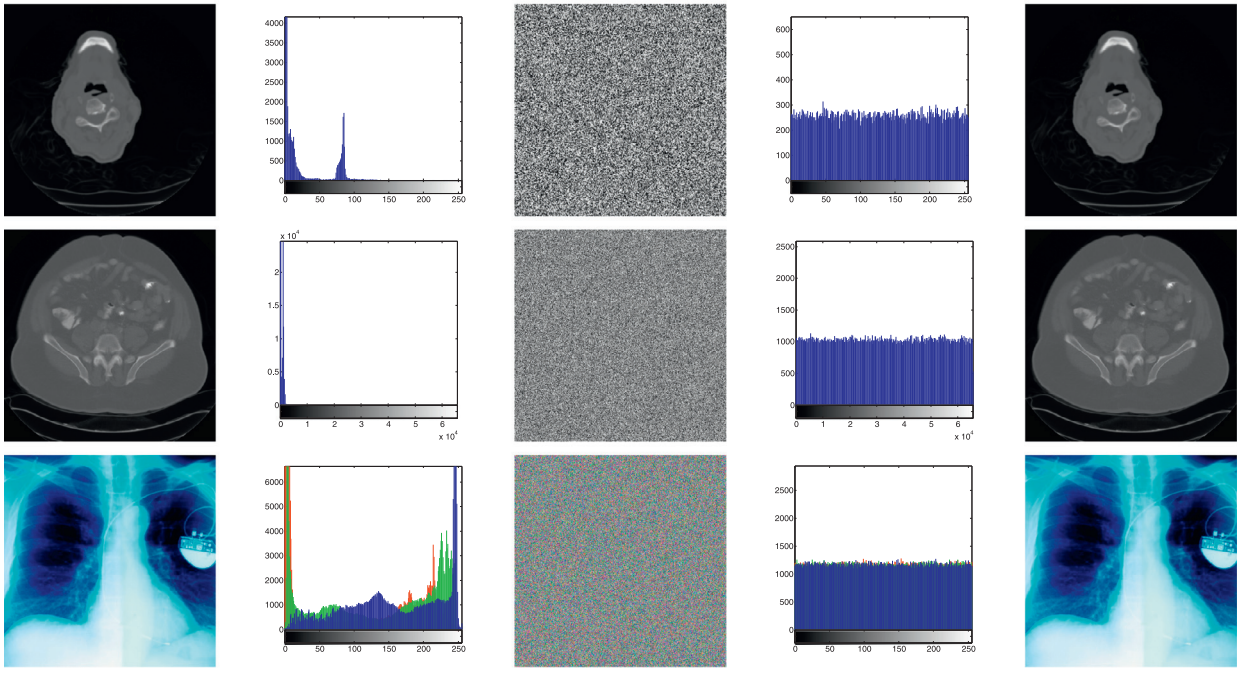

Fig. 8. Simulation results of MIE-BX. The images from top to bottom are 8-bit, 16-bit and 24-bit images. (a) Plain-images; (b) histograms of (a); (c) cipher-images; (d) histograms of (c); (d) decrypted images.

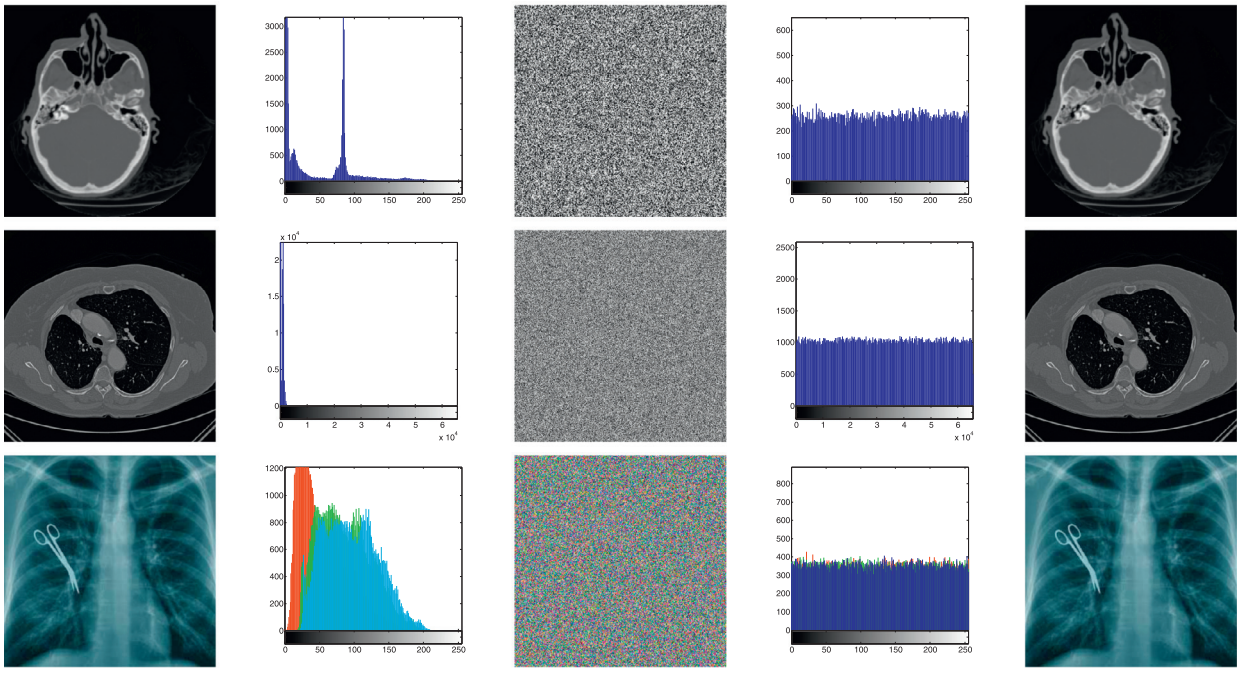

Fig. 9. Simulation results of MIE-MA. The images from top to bottom are 8-bit, 16-bit and 24-bit images. (a) Plain-images; (b) histograms of (a); (c) cipher-images; (d) histograms of (c); (d) decrypted images.

age by the subsequent diffusion operation. By doing these, MIE-BX and MIE-MA can achieve many advantages:

- As MIE-BX and MIE-MA strictly follow the principles of confusion and diffusion, and it can encrypt an identical image into different cipher-images using a same secret key, they have strong ability of resisting well-known attacks, such as the chosen-plaintext attack, and differential attack. This will be experimentally verified in Section 4.1.
- As they have simple structures and can be directly applied to images with any representation format, they can achieve high computation efficiency;
- They have high robustness, which means that when blurring or data loss happens to cipher-images, the algorithms can still recover the original image with a high visual quality.

Table 2
The information entropies of different plain-images and their cipher-images encrypted by MIE-BX and MIE-MA. For all the 8-bit, 16-bit and 24-bits images, 256 bins are used.

	MIE-BX			MIE-MA		
	8-bit	16-bit	24-bit	8-bit	16-bit	24-bit
Plain-images	4.2051	5.6775	6.3962	5.2431	5.7915	6.8723
Cipher-images	7.9977	7.9994	7.9981	7.9969	7.9994	7.9965

4. Performance evaluations

This section evaluates the performance of MIE-BX and MIE-MA from three aspects: security, efficiency and ability of defending noise and data loss. Several classical and art-of-the-state encryption schemes are selected as comparison methods: AES-CFB, AES with Cipher Block Chaining mode (AES-CBC), AES with Electronic Codebook mode (AES-ECB), HZPC [13], DPR [18], BW [38], TMW [39], WZNS [32] and ZHPC [34].

4.1. Security

Here, we measure key security and randomness of cipher-image.

4.1.1. Key security

First, the key space should be appropriate. The key space of MIE-BX and MIE-MA is 2^{256}, which has a proper size and high ability of defending brute-force attack [40]. On the other hand, the key should be extremely sensitive. If the secret key is not sensitive enough, which means that some little different secret keys can also correctly reconstruct the original image, the secret key may degenerate and the actual key space may less than the theoretical one [40–42].

Here, we use the number of bit change rate (NBCR) [43] to test key sensitivity. The NBCR of two images $\mathbf{B}_1$ and $\mathbf{B}_2$ can be defined as

$$\mathtt{NBCR}(\mathbf{B}_1, \mathbf{B}_2) = \frac{\mathtt{Ham}(\mathbf{B}_1, \mathbf{B}_2)}{Len}, \tag{10}$$

where $\mathtt{Ham}(\mathbf{B}_1, \mathbf{B}_2)$ denotes the Hamming distance of $\mathbf{B}_1$ and $\mathbf{B}_2$, and *Len* is the total number of bits of $\mathbf{B}_1$ or $\mathbf{B}_2$. If the obtained NBCR approaches to 50%, $\mathbf{B}_1$ and $\mathbf{B}_2$ are totally different images with no correlations.

To test the key sensitivity of MIE-BX and MIE-MA in both the encryption and decryption processes, we first generate a random secret key,

$$\mathbf{K}_1 = 5295BA0009046C9825358AE378554C4258C223090F5D14 F06E2993D9145A0C59.$$

For each of MIE-BX and MIE-MA, our experiments are designed as follows: 1) change one bit of $\mathbf{K}_1$ to obtain $\mathbf{K}_2$; 2) encrypt a plain-image $\mathbf{P}$ using $\mathbf{K}_1$ and $\mathbf{K}_2$ to generate two cipher-images $\mathbf{C}_1$ and $\mathbf{C}_2$, and then calculate their NBCR; 3) decrypt a same cipher-image $\mathbf{C}_1$ using $\mathbf{K}_1$ and $\mathbf{K}_2$ to generate two decrypted images $\mathbf{D}_1$ and $\mathbf{D}_2$, and then calculate their NBCR. Fig. 10 shows the key sensitivity analysis results for each of 256 bits. From the results we can observe that when changing any one of 256 bits in a secret key, the obtained two cipher-images in encryption process and two decrypted results in decryption process are totally different. This means that the secret keys of MIE-BX and MIE-MA are extremely sensitive.

4.1.2. Ability of resisting chosen-plaintext attack

The chosen-plaintext attack is a kind of cryptanalysis model, where the attackers can choose the plaintexts to encrypt and obtain their corresponding ciphertexts. By analyzing the plaintexts and their corresponding ciphertexts, the attackers try to deduce some secret information, and use these obtained information to recover the original images. Many researchers have studied that some cryptosystems can be successfully attacked by chosen-plaintext attack [44,45].

An image encryption algorithm with high security level should have the ability to resist chosen-plaintext attack. In our proposed MIE-BX and MIE-MA, random noise data are inserted into the surroundings and these data will affect all the pixels after the high-speed scrambling and pixel adaptive diffusion. As a result, when encrypting an identical image many time using a same secret key, the obtained cipher-images are totally different, as the inserted data in each encryption are different.

To experimentally demonstrate this property, we take 20 medical images from the SPIE-AAPM Lung CT Challenge 'CT-Training-BE001' dataset as examples. For each image, we use the same secure key to encrypt them twice, and calculate the number of bit change rate (NBCR) (defined in Eq. (10)) of the two obtained cipher-images. Table 3 lists the test results. One can see that all the NBCRs close to 50%, which demonstrates that encrypting an identical image twice using the same secure key, the obtained two cipher-images are totally different. When each cipher-image is unique and different, attackers can not repeat the encryption process and have difficult to obtain the relations between plaintext and ciphertexts, and thus the chosen-plaintext attack on MIE-MA and MIE-BX are noneffective. Note that an encryption algorithm with strong ability of defending chosen-plaintext attack can also defend known-plaintext attack, MIE-MA and MIE-BX can also resist known-plaintext attack.

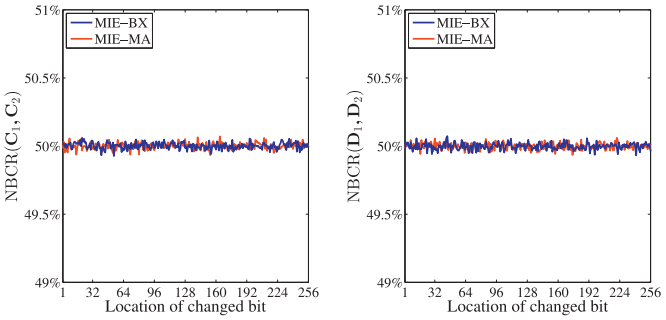

Fig. 10. Key sensitivity analysis: (a) NBCR between $\mathbf{C}_1$ and $\mathbf{C}_2$, where $\mathbf{C}_1$ and $\mathbf{C}_2$ are two cipher-images encrypted from a same plain-image and two secret keys with one bit difference; (b) NBCR between $\mathbf{D}_1$ and $\mathbf{D}_2$, where $\mathbf{D}_1$ and $\mathbf{D}_2$ are two decrypted results from a same cipher-image and two secret keys with one bit deference.

Table 3
The NBCRs of two cipher-images encrypted from an identical image using the same secret key.

File name	MIE-BX	MIE-MA	File name	MIE-BX	MIE-MA
000000.dcm	49.98%	49.97%	000010.dcm	49.98%	50.05%
000001.dcm	49.95%	49.99%	000011.dcm	50.03%	50.01%
000002.dcm	50.01%	49.99%	000012.dcm	50.00%	49.99%
000003.dcm	49.97%	49.99%	000013.dcm	50.00%	49.99%
000004.dcm	50.03%	50.02%	000014.dcm	50.00%	50.02%
000005.dcm	50.03%	49.98%	000015.dcm	50.04%	49.98%
000006.dcm	50.00%	49.99%	000016.dcm	49.99%	49.98%
000007.dcm	50.02%	50.03%	000017.dcm	50.00%	49.99%
000008.dcm	49.97%	50.00%	000018.dcm	50.00%	49.98%
000009.dcm	50.00%	50.03%	000019.dcm	49.99%	49.98%

Table 4
The NPCR and UACI results of the MIE-BX and MIE-MA.

File name	MIE-BX		MIE-MA	
	NPCR	UACI	NPCR	UACI
000000.dcm	99.9974%	33.2716%	99.9996%	33.3182%
000001.dcm	99.9989%	33.3266%	99.9996%	33.3862%
000002.dcm	99.9996%	33.3451%	99.9992%	33.3426%
000003.dcm	99.9996%	33.2651%	99.9977%	33.3624%
000004.dcm	99.9977%	33.3053%	99.9974%	33.3273%
000005.dcm	99.9977%	33.2878%	99.9981%	33.3321%
000006.dcm	99.9970%	33.3107%	99.9977%	33.3230%
000007.dcm	99.9974%	33.3172%	99.9974%	33.3264%
000008.dcm	99.9989%	33.3729%	99.9996%	33.3077%
000009.dcm	99.9981%	33.3500%	99.9989%	33.2860%
000010.dcm	99.9985%	33.3030%	99.9977%	33.3911%
000011.dcm	99.9977%	33.3358%	99.9989%	33.3648%
000012.dcm	99.9985%	33.3652%	99.9977%	33.3526%
000013.dcm	99.9981%	33.3725%	99.9989%	33.3572%
000014.dcm	99.9996%	33.3996%	99.9977%	33.2857%
000015.dcm	99.9974%	33.3768%	99.9989%	33.3165%
000016.dcm	99.9989%	33.3486%	99.9985%	33.3887%
000017.dcm	99.9989%	33.3296%	99.9977%	33.3881%
000018.dcm	99.9974%	33.3026%	99.9989%	33.3260%
000019.dcm	99.9977%	33.3558%	99.9974%	33.3559%

4.1.3. Ability of resisting differential attack

The differential attack traces how the difference in plaintexts can affect the ciphertexts. For an image encryption algorithm, its ability of resisting differential attack can be quantitatively tested by the number of pixel change rate (NPCR) and uniform average change intensity (UACI). NPCR measures the number of different pixels of two images while UACI collects the different values of pixels of two images. Suppose $\mathbf{C}_1$ and $\mathbf{C}_2$ are two cipher-images encrypted from two plain-images with only one bit difference, NPCR and UACI are defined as

$$\mathtt{NPCR}(\mathbf{C}_1, \mathbf{C}_2) = \sum_{i,j} \frac{\mathbf{A}(i,j)}{G} \times 100\%,$$

and

$$\mathtt{UACI}(\mathbf{C}_1, \mathbf{C}_2) = \sum_{i,j} \frac{|\mathbf{C}_1(i,j) - \mathbf{C}_2(i,j)|}{(L-1) \times G} \times 100\%,$$

where G denotes the total number of pixel in each cipher-image, L indicates the number of allowed pixel value, and $\mathbf{A}$ represents the difference between $\mathbf{C}_1$ and $\mathbf{C}_2$, which is defined as

$$\mathbf{A}(i,j) = \begin{cases} 0, & \text{if } \mathbf{C}_1(i,j) = \mathbf{C}_2(i,j), \\ 1, & \text{if } \mathbf{C}_1(i,j) \neq \mathbf{C}_2(i,j). \end{cases}$$

Recently, the NPCR and UACI scores for an image encryption algorithm with a good ability of resisting the differential attack were provided in [46]. The expected NPCR and UACI values of two independent random images are given by

$$\mathrm{NPCR}_{expected} = \left(1 - \frac{1}{2^{\log_2^L}}\right) \times 100\%, \tag{11}$$

and

$$\mathrm{UACI}_{expected} = \frac{1}{L^2}\left(\frac{\sum_{i=1}^{(L-1)} i(i+1)}{L-1}\right) \times 100\%, \tag{12}$$

respectively. It is obvious that the expected NPCR and UACI values correspond to the gray level of an image.

In our experiment, we use 20 medical images (filenames are from “000000.dcm” to “000019.dcm”) selected from the SPIE-AAPM Lung CT Challenge ‘CT-Training-BE001’ dataset to do the experiment. Table 4 shows the NPCR and UACI results of these images for both MIE-MA and MIE-BX. As every pixel of these images is represented by 16 bits, the expected NPCR and UACI are 99.9985% and 33.3338% according to Eqs. (11) and (12). As can be seen from the table, both MIE-MA and MIE-BX can obtain NPCRs and UACIs that are quite close to the expected values. Thus, MIE-MA and MIE-BX have strong ability of resisting differential attack.

Table 5
Average encryption time (E.T.) and decryption time (D.T.) of different encryption schemes for images with different sizes.

Image size	$256 \times 256 \times 16$		$512 \times 512 \times 16$		$1024 \times 1024 \times 16$	
	E.T. (s)	D.T. (s)	E.T. (s)	E.T. (s)	D.T. (s)	E.T. (s)
AES-CFB	8.9678	8.9870	35.6805	35.6197	143.0931	142.8006
AES-CBC	6.8299	6.9057	27.2153	27.5655	110.4678	110.4279
AES-ECB	6.7954	6.8138	27.0125	27.1624	109.3918	108.4706
HZPC	0.1112	0.1065	0.4973	0.4794	2.1126	2.0364
DPR	–	–	0.9662	0.9620	3.7570	3.7580
BW	0.1753	0.1838	1.3842	1.4671	11.0031	11.4628
TMW	0.0750	0.1110	0.4425	0.7155	6.4339	9.0132
WZNS	0.4474	0.4216	1.7938	1.7028	7.4772	7.0774
ZHPC	0.3672	0.3262	2.7225	2.7804	27.6921	27.7257
MIE-BX	0.1166	0.1175	0.4687	0.4646	1.9413	1.9662
MIE-MA	0.0248	0.0247	0.1053	0.1043	0.5098	0.5207

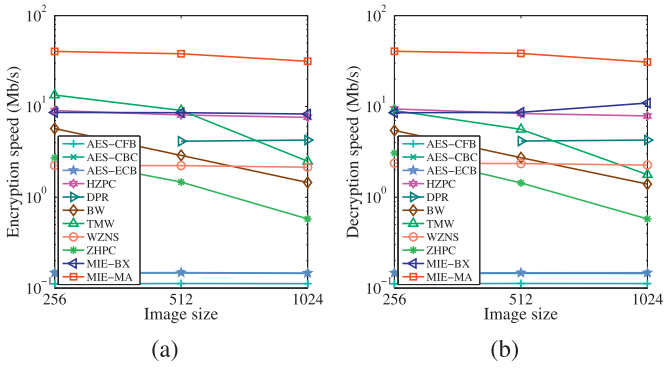

Fig. 11. Average speeds of different encryption schemes against image size: (a) encryption speed; (b) decryption speed.

4.2. Efficiency

We tested the efficiency of different encryption schemes. The processer of used computer is Intel(R) Core(TM) i5-3320M CPU @ 2.6 GHz. Two groups of experiments were designed and the first group tests 16-bit medical images with different sizes. Table 5 lists the average encryption time and decryption time of 20 medical images. Fig. 11 shows the average encryption and decryption speeds of different encryption schemes for image with different sizes. MIE-MA can achieve much faster speed than other schemes. This is due to the two reasons: 1) our proposed encryption scheme encrypts image in pixel level, while some others encrypt image in bit level; 2) benefited from that random data insertion can ensure the encrypted result of each execution is unique and different, two rounds of substation-permutation in our proposed scheme can guarantee a high security level, while others perform multiple rounds.

The second group of experiments tests images with different representation formats by fixing image size as 512×512. For different encryption schemes, Table 6 shows their average time of encrypting and decrypting twenty 8-bit, 16-bit, 24-bit and 32-bit images, and Fig. 12 shows their average encryption and decryption speeds. Because MIE-BX and MIE-MA encrypts images in pixel level. Their encryption time only corresponds to the image size, and is irrelevant to how many bits to present a pixel. For an image with fixed size, when using more bits to present every pixel, the image data increases, but the encryption time remains almost the same, namely the speed increases. This can be observed from

Table 6
Average encryption time (E.T.) and decryption time (D.T.) of different encryption schemes for image with different bit depths.

Image size	$512 \times 512 \times 8$		$512 \times 512 \times 16$		$512 \times 512 \times 24$		$512 \times 512 \times 32$	
	E.T. (s)	D.T. (s)	E.T. (s)	E.T. (s)	D.T. (s)	E.T. (s)	D.T. (s)	E.T. (s)
AES-CFB	17.8718	17.9117	35.6805	35.6197	53.7655	53.7368	71.8799	71.8256
AES-CBC	13.6021	13.7416	27.2153	27.5655	40.8740	41.2740	54.6284	55.1508
AES-ECB	13.5105	13.5414	27.0125	27.1624	40.6132	40.6698	54.2302	55.2340
HZPC	0.5017	0.4858	0.4973	0.4794	0.4776	0.4711	0.4666	0.4480
BW	1.3635	1.4659	1.3824	1.4671	1.3558	1.4708	1.3651	1.4645
TMW	0.4159	0.6802	0.4425	0.7155	0.4286	0.6925	0.4219	0.7218
WZNS	0.9286	0.8769	1.7938	1.7028	2.7496	2.6230	3.7149	3.5479
ZHPC	2.8759	2.8760	2.7225	2.7804	2.8775	2.8979	2.8781	2.8604
MIE-BX	0.4654	0.4697	0.4687	0.4646	0.4804	0.4842	0.4640	0.4685
MIE-MA	0.1065	0.1057	0.1053	0.1043	0.1070	0.1081	0.1041	0.1050

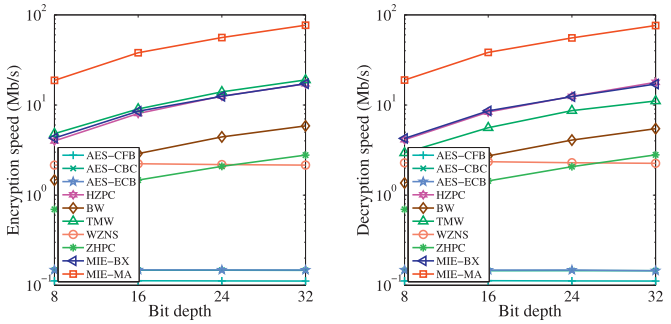

Fig. 12. Average speeds of different encryption schemes against bit depth: (a) encryption speed; (b) decryption speed.

Table 6 and Fig. 12. Besides, we can observe that, among all the encryption schemes, MIE-MA can achieve the fastest speed.

4.3. Robustness of defending noise and data loss

When a cipher-image is blurred or losses some data, a reliable encryption scheme should still recover the original image without missing too much significant information.

As the pixel adaptive diffusion operation in encryption and decryption processes are asymmetric, MIE-BX and MIE-MA have strong robustness to defend impulse noise and data loss. Fig. 13 demonstrates the asymmetric structure of pixel adaptive diffusion operation in MIE-BX and MIE-MA. In the encryption process, some random values are inserted into surroundings of the image and after the subsequent diffusion, these random values can affect all the pixels. However, in the decryption process, the change of one pixel in cipher-image can affect only two pixels when doing inverse diffusion. By this principle, when a portion of pixels in the cipher-image are changed, the decryption process can still recover the original image with a high visual quality.

Fig. 14 demonstrates the situations that data loss happens in cipher-images for different encryption schemes. The first and third rows show cipher-images with about 2% data loss in central of the image and the second and fourth rows show the corresponding decrypted results. As can be observed, the decrypted image in AES-CBC losses all the image information of the associated area, the decrypted images in WZNS and BW loss all the image information, the decrypted image in TMW have pixel changes in the whole image, and decrypted images in HZPC, ZHPC, MIE-BX and MIE-MA are globally visualized with some noise. However, it is obvious that MIE-BX and MIE-MA can achieve decrypted results with much less noise than other schemes.

To quantitatively measure the reliability of defending noise and data loss, the peak signal-to-noise ratio (PSNR) introduced in [47] is used. It can measure the difference between original image **P** and decrypted image **D**. Suppose the image is of size $M \times N$, PSRN is defined as

$$\mathrm{PSNR} = 10 \times \log_{10}\left\{\frac{(2^E-1)^2}{\mathrm{MSE}}\right\}, \tag{13}$$

where E is the bit depth of image and the mean square error (MSE) is defined as

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(\mathbf{P}_{i,j}-\mathbf{D}_{i,j})^2. \tag{14}$$

Higher PSNR means less difference between **P** and **D**. If **P** and **D** are the same, their PSNR is infinity.

The first group of experiments tests the reliability of defending impulse noise. For each encryption scheme, we add different percentages of salt-and-pepper noise to cipher-images and subsequently decrypt these cipher-images, and then calculate PSNRs between the original image and these decrypted results. Table 7 shows the PSRN results of different encryption schemes. MIE-BX and MIE-MA can obtain higher PSNRs than the rest encryption schemes.

The second group of experiments tests the reliability of defending data loss. For each encryption scheme with a given data loss percentage, the experiment is performed as follows: 1) encrypt a plain-image into cipher-image; 2) rearrange the two-dimensional (2D) data of cipher-image into one-dimensional (1D), since data are transmitted as bit streams in transmission channels; 3) set the front part of data with the given percentage to be 0; 4) rearrange the 1D data back to 2D and decrypt it; 5) calculate PSNR between the original image and decrypted result. Table 8 shows PSRN results of different encryption schemes with different percentages of data loss in the cipher-images. As the three AES work modes encrypt data bit by bit, they can achieve the highest PSNRs. Except for them, MIE-BX and MIE-MA can also achieve high PSNRs, which means that they have good ability of defending data loss.

5. Conclusion

This paper introduced a high-efficiency and robust encryption scheme to protect medical images. It is composed of three components: random data insertion, high-speed scrambling and pixel adaptive diffusion. The random data insertion is to add some random value to surroundings of the image. The high-speed scrambling is to randomly shuffle neighboring pixels. The pixel adaptive diffusion is to spread these inserted values to entire pixels. As the proposed encryption scheme can be directly applied to images with any representation format, it can achieve faster speed when

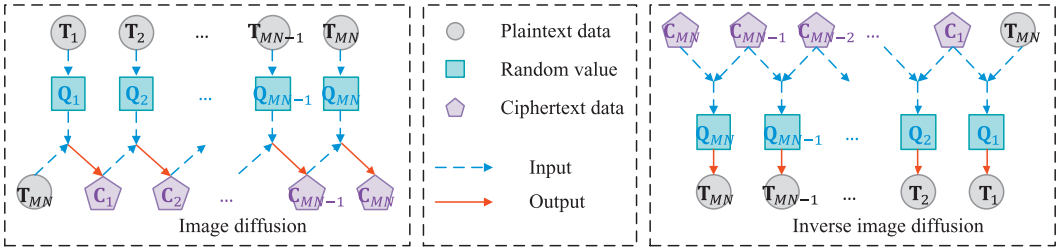

Fig. 13. Asymmetric structure of pixel adaptive diffusion in MIE-BX and MIE-MA. The left and right diagrams demonstrate the operation of pixel adaptive diffusion in Eq. (5) or Eq. (6) and the operation of inverse diffusion in Eq. (7) or Eq. (8), respectively.

Table 7
PSNR results of different encryption schemes with different percentages of salt-and-pepper noise.

Noise percentages (%)	0.005	0.01	0.05	0.1	0.5	1	5
AES-CFB	38.2000	33.7359	28.1414	24.7400	17.7389	15.0117	9.3495
AES-CBC	35.4874	33.7518	27.4401	24.7846	17.7263	14.9289	9.3944
AES-ECB	40.2702	34.2813	27.9288	24.3786	18.0616	15.2176	9.4920
HZPC	40.0133	33.0153	27.3373	25.3750	18.2612	15.2922	9.5086
BW	8.9365	8.9365	8.9365	8.9365	8.9365	8.9365	8.9365
TMW	9.4414	9.4416	9.4409	9.4420	9.4426	9.4472	9.4533
WZNS	27.5721	24.4354	16.3985	13.7587	8.4428	7.3506	7.0013
ZHPC	39.2570	37.9052	29.9551	27.8602	20.5037	17.6369	11.2990
MIE-BX	45.2897	41.5032	34.4531	31.1051	24.0857	21.1961	14.3645
MIE-MA	44.6574	41.3979	34.1717	30.9284	24.0263	21.1761	14.3195

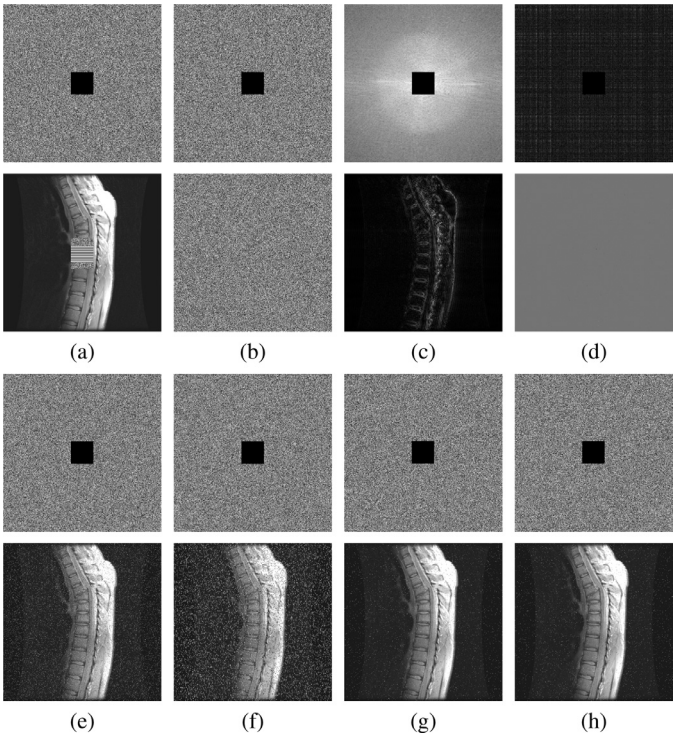

Fig. 14. The first and third rows are the cipher-images with about 2% data loss and the second and fourth rows are the corresponding decrypted results, where the encryption schemes are (a) AES-CBC, (b) WZNS, (c) TMW, (d) BW, (e) HZPC, (f) ZHPC, (g) MIE-BX, and (h) MIE-MA.

every pixel of the image is represented by more bits. Using different operations to implement the pixel adaptive diffusion, we provided two versions for the proposed encryption scheme: MIE-BX and MIE-MA. They have high efficiency in hardware and software platforms, respectively. Users have flexibility to select a proper implementation according to the actual environment. Simulation results and performance evaluations were provided. Compared with some classical encryption schemes, MIE-BX and MIE-MA have high security levels, and can achieve much higher efficiency and better robustness of defending data loss and impulse noises. As our proposed encryption scheme has high efficiency, we will investigate its applications in more multimedia files such as medicine videos.

Table 8
PSNR results of different encryption schemes with different percentages of data loss.

Data loss percentage (%)	0.01	0.05	0.1	0.5	1	5
AES-CFB	42.9027	38.9279	36.4463	29.8818	26.8918	19.9400
AES-CBC	44.2391	37.7542	35.0361	28.2328	25.2278	18.2531
AES-ECB	45.2118	37.8735	35.1196	28.2642	25.2442	18.2529
HZPC	37.7218	31.0985	28.0790	22.5456	20.3096	14.1595
BW	8.9364	8.9364	8.9364	8.9364	8.9364	8.9365
TMW	20.7533	20.7676	20.7779	20.7873	20.7740	20.9193
WZNS	23.3740	17.1888	15.0784	10.5159	10.1254	10.0783
ZHPC	37.0756	29.9666	27.3330	20.5016	17.6193	11.3568
MIE-BX	45.6263	36.8437	34.0194	26.8247	23.9041	17.0768
MIE-MA	44.8217	37.1681	33.9738	26.9081	24.0189	17.1581

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61701137, in part by the Shenzhen Science and Technology Innovation Council under Grant JCYJ20170307150704051, in part by the Macau Science and Technology Development Fund under Grant FDCT/016/2015/A1, and in part by the Research Committee at University of Macau under Grants MYRG2014-00003-FST and MYRG2016-00123-FST.

References

[1] S. Zinger, D. Ruijters, L. Do, P.H.N. de With, View interpolation for medical images on autostereoscopic displays, IEEE Trans. Circuits Syst. Video Technol. 22 (1) (2012) 128–137.
[2] C. Lacoste, J.H. Lim, J.P. Chevallet, D.T.H. Le, Medical-image retrieval based on knowledge-assisted text and image indexing, IEEE Trans. Circuits Syst. Video Technol. 17 (7) (2007) 889–900.
[3] X. Chai, Y. Chen, L. Broyde, A novel chaos-based image encryption algorithm using DNA sequence operations, Opt. Lasers Eng. 88 (2017) 197–213.
[4] Y.-Q. Zhang, X.-Y. Wang, A new image encryption algorithm based on non-adjacent coupled map lattices, Appl. Soft Comput. 26 (2015) 10–20.

[5] P. Ping, F. Xu, Z.-J. Wang, Image encryption based on non-affine and balanced cellular automata, Signal Process. 105 (2014) 419–429.
[6] X. Chai, Z. Gan, Y. Chen, Y. Zhang, A visually secure image encryption scheme based on compressive sensing, Signal Process. 134 (2017) 35–51.
[7] X.-W. Li, I.-K. Lee, Modified computational integral imaging-based double image encryption using fractional fourier transform, Opt. Lasers Eng. 66 (2015) 112–121.
[8] H. Liu, X. Wang, A. kadir, Image encryption using DNA complementary rule and chaotic maps, Appl. Soft Comput. 12 (5) (2012) 1457–1466.
[9] L. Gong, X. Liu, F. Zheng, N. Zhou, Flexible multiple-image encryption algorithm based on log-polar transform and double random phase encoding technique, J. Modern Opt. 60 (13) (2013) 1074–1082.
[10] X.-Y. Wang, Y.-Q. Zhang, X.-M. Bao, A novel chaotic image encryption scheme using dna sequence operations, Opt. Lasers Eng. 73 (2015) 53–61.
[11] N. Zhou, Y. Hu, L. Gong, G. Li, Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations, Quant. Inf. Process. 16 (6) (2017) 164.
[12] Y.-Q. Zhang, X.-Y. Wang, A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice, Inf. Sci. 273 (2014) 329–351.
[13] Z. Hua, Y. Zhou, C.-M. Pun, C.L.P. Chen, 2D Sine logistic modulation map for image encryption, Inf. Sci. 297 (2015) 80–94.
[14] M. Zanin, A.N. Pisarchik, Gray code permutation algorithm for high-dimensional data encryption, Inf. Sci. 270 (2014) 288–297.
[15] M. Prakash, P. Balasubramaniam, S. Lakshmanan, Synchronization of Markovian jumping inertial neural networks and its applications in image encryption, Neural Networks 83 (2016) 86–93.
[16] A. Belazi, A.A. Abd El-Latif, S. Belghith, A novel image encryption scheme based on substitution-permutation network and chaos, Signal Process. 128 (2016) 155–170.
[17] Z. Hua, Y. Zhou, Design of image cipher using block-based scrambling and image filtering, Inf. Sci. 396 (2017) 97–113.
[18] M. Dzwonkowski, M. Papaj, R. Rykaczewski, A new quaternion-based encryption method for DICOM images, IEEE Trans. Image Process. 24 (11) (2015) 4614–4622.
[19] P. Mildenberger, M. Eichelberg, E. Martin, Introduction to the DICOM standard, Eur. Radiol. 12 (4) (2002) 920–927.
[20] Y. Zhang, D. Xiao, An image encryption scheme based on rotation matrix bit-level permutation and block diffusion, Commun. Nonlinear Sci. Numer. Simul. 19 (1) (2014) 74–82.
[21] N. Zhou, S. Pan, S. Cheng, Z. Zhou, Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing, Optics Laser Technol. 82 (2016) 121–133.
[22] L.-B. Zhang, Z.-L. Zhu, B.-Q. Yang, W.-Y. Liu, H.-F. Zhu, M.-Y. Zou, Medical image encryption and compression scheme using compressive sensing and pixel swapping based permutation approach, Math. Probl. Eng. 2015 (2015) 940638.
[23] C. Li, D. Lin, J. Lü, Cryptanalyzing an image scrambling encryption algorithm of pixel bits, IEEE MultiMedia (2017). arXiv:1607.01642.
[24] E.Y. Xie, C. Li, S. Yu, J. Lü, On the cryptanalysis of Fridrich's chaotic image encryption scheme, Signal Process. 132 (2017) 150–154.
[25] L.Y. Zhang, Y. Liu, F. Pareschi, Y. Zhang, K.-W. Wong, R. Rovatti, G. Setti, On the security of a class of diffusion mechanisms for image encryption, IEEE Trans. Cybern. (2017), doi:10.1109/TCYB.2017.2682561.
[26] Z. Hua, S. Yi, Y. Zhou, C. Li, Y. Wu, Designing hyperchaotic cat maps with any desired number of positive Lyapunov exponents, IEEE Trans. Cybern. (2017), doi:10.1109/TCYB.2016.2642166.
[27] C. Li, Y. Liu, T. Xie, M.Z.Q. Chen, Breaking a novel image encryption scheme based on improved hyperchaotic sequences, Nonlinear Dyn. 73 (3) (2013) 2083–2089.
[28] S. Ergün, On the security of chaos based "true" random number generators, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 99 (1) (2016) 363–369.
[29] T. Hütter, M. Preishuber, J. Hämmerle-Uhl, A. Uhl, Weaknesses in Security Considerations Related to Chaos-Based Image Encryption, Springer, 2016.
[30] M. Larobina, L. Murino, Medical image file formats, J. Digital Imag. 27 (2) (2014) 200–206.
[31] X. Wang, L. Liu, Y. Zhang, A novel chaotic block image encryption algorithm based on dynamic random growth technique, Optics Lasers Eng. 66 (2015) 10–18.
[32] Y. Wu, Y. Zhou, J.P. Noonan, S. Agaian, Design of image cipher using latin squares, Inf. Sci. 264 (2014) 317–339.
[33] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, B.-Q. Yang, Exploiting self-adaptive permutation-diffusion and dna random encoding for secure and efficient image encryption, Signal Process. 142 (2018) 340–353.
[34] Y. Zhou, Z. Hua, C.-M. Pun, C.L.P. Chen, Cascade chaotic system with applications, IEEE Trans. Cybern. 45 (9) (2015) 2001–2012.
[35] Z. Hua, Y. Zhou, Image encryption using 2D logistic-adjusted-sine map, Inf. Sci. 339 (2016) 237–253.
[36] Y. Zhang, D. Xiao, Y. Shu, J. Li, A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations, Signal Process. 28 (3) (2013) 292–300.
[37] Y. Zhou, L. Bao, C.L.P. Chen, A new 1D chaotic system for image encryption, Signal Process. 97 (2014) 172–182.
[38] G. Bhatnagar, Q.M.J. Wu, Biometric inspired multimedia encryption based on dual parameter fractional fourier transform, IEEE Trans. Syst. Man Cybern. 44 (9) (2014) 1234–1247.
[39] R. Tao, X.-Y. Meng, Y. Wang, Image encryption with multiorders of fractional fourier transforms, IEEE Trans. Inf. Foren. Secur. 5 (4) (2010) 734–738.
[40] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, Int. J. Bifur. Chaos 16 (08) (2006) 2129–2151.
[41] C. Li, Cracking a hierarchical chaotic image encryption algorithm based on permutation, Signal Process. 118 (2016) 203–210.
[42] D. Arroyo, F. Hernandez, A.B. Orue, Cryptanalysis of a classical chaos-based cryptosystem with some quantum cryptography features, Int. J. Bifur. Chaos 27 (1) (2017). Art. no. 1750004
[43] J.C.H. Castro, J.M. Sierra, A. Seznec, A. Izquierdo, A. Ribagorda, The strict avalanche criterion randomness test, Math. Comput. Simul. 68 (1) (2005) 1–7.
[44] J. Wu, W. Liu, Z. Liu, S. Liu, Correlated-imaging-based chosen plaintext attack on general cryptosystems composed of linear canonical transforms and phase encodings, Optics Commun. 338 (2015) 164–167.
[45] R. Van Den Assem, W. Van Elk, A chosen-plaintext attack on the microsoft basic protection, Comput. Secur. 5 (1) (1986) 36–45.
[46] C. Fu, J. Chen, H. Zou, W. Meng, Y. Zhan, Y. Yu, A chaos-based digital image encryption scheme with an improved diffusion strategy, Optics Express 20 (3) (2012) 2363–2378.
[47] C.-Y. Chen, C.-H. Chen, C.-H. Chen, K.-P. Lin, An automatic filtering convergence method for iterative impulse noise filters based on PSNR checking and filtered pixels detection, Expert Syst. Appl. 63 (2016) 198–207.