Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.013039
Article

ALCencryption: A Secure and Efficient Algorithm for Medical Image Encryption

Jiao Ge[1,2,*]

[1]College of Computer Science and Technology, Hengyang Normal University, Hengyang, 421002, China
[2]Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China
*Corresponding Author: Jiao Ge. Email: jiaoge@126.com
Received: 22 July 2020; Accepted: 22 September 2020

Abstract: With the rapid development of medical informatization and the popularization of digital imaging equipment, DICOM images contain the personal privacy of patients, and there are security risks in the process of storage and transmission, so it needs to be encrypted. In order to solve the security problem of medical images on mobile devices, a safe and efficient medical image encryption algorithm called ALCencryption is designed. The algorithm first analyzes the medical image and distinguishes the color image from the gray image. For gray images, the improved Arnold map is used to scramble them according to the optimal number of iterations, and then the diffusion is realized by the Logistic and Chebyshev map cross-diffusion algorithm. The color image is encrypted by cross-diffusion algorithm of double chaotic map. Security and efficiency analysis show that the ALCencryption algorithm has the characteristics of small neighboring pixels, large key space, strong key sensitivity, high safety and short encryption time. It is suitable for medical image encryption of mobile devices with high real-time requirements.

Keywords: Patient privacy; DICOM; medical image encryption; scrambling degree; cross-diffusion

1 Introduction

With the progress of information technology and medical field, there is a growing demand for DICOM images to be transmitted through public network. DICOM images contain patients' privacy data. In order to ensure the safe access of medical images, encryption processing is required. DICOM image encryption algorithm based on chaos theory is a recognized new method with high security, fast and effective. According to the dimension of chaos, this algorithm can be divided into one-dimensional, multi-dimension and mixed chaotic mapping.

Most medical image encryption schemes only use a single chaotic map to transform and diffuse image pixels. Hu et al. [1] scrambled the image pixels with the real random numbers generated by chaotic sequences as the key, which was 100 times faster than scrambling the images with AES, but did not diffuse the pixel values, affecting the security of encryption. Sathishkumar et al. [2] proposed a medical image encryption algorithm based on double chaos, which is transformed and scrambled by chaotic cyclic shift. Kanso et al. [3] used Cat mapping

to scramble image pixels. Fu et al. [4] used Arnold Cat mapping to eliminate the correlation between adjacent pixels, and used Logistic mapping to achieve pixel confusion between plaintext image and ciphertext image. The medical image encryption algorithm based on one-dimensional chaotic mapping has the advantages of low algorithm complexity, simple implementation and fast encryption speed, but it has the disadvantages of uneven key distribution, limited unstable range of chaotic sequence, insufficient security in resisting differential attacks, choosing plaintext attacks and choosing ciphertext attacks.

2 Related Work

In order to make up for the shortage of one-dimensional chaotic image encryption, many researchers use high-dimensional chaotic map for medical image encryption. Fu et al. [5] used three-dimensional Chen chaotic mapping to carry out displacement and diffusion of DICOM image pixels, enhancing the security of attacks on known and selected plaintext. Chandrasekaran et al. [6] proposed a DICOM image encryption algorithm combining number theory method with Henon mapping, where key matrix is scrambled and chaotic controlled by Henon mapping, which can effectively resist statistical and differential attacks. Seyedzadeh et al. [7] proposed a color image encryption algorithm based on two-dimensional piecewise nonlinear chaotic mapping, which is characterized by high sensitivity, high security and high speed. In the medical image encryption algorithm with high dimensional chaos, three or four variables are generally used to transform the image pixels, which obviously enhances the security of the algorithm, but also increases the complexity and computational overhead of the algorithm.

Considering the advantages and disadvantages of one-dimensional and higher-dimensional chaotic maps, some researchers have begun to study medical image encryption algorithms that combine chaotic maps from multiple dimensions. Dai et al. [8] proposed a medical image encryption method based on the combination of Logistic mapping and Chebyshev mapping. By setting the parameters of Logistic mapping reasonably, this method firstly used Logistic mapping to encrypt the original image, and then used Chebyshev mapping to encrypt it again. This algorithm has higher transmission security and larger key space. Zhou et al. [9] proposed an image encryption method based on the combination of three one-dimensional chaotic maps, which has higher security and lower computing cost compared with higher-order chaotic maps. Ravichandran et al. [10] proposed a new encryption scheme based on Logistic, Tent and Sine chaotic mapping, which provided better protection for real-time medical image security applications. Boussif et al. [11] proposed an image encryption algorithm based on matrix product and separate addition, which realized the secure transmission of encrypted medical images by smart phones. The encryption algorithm has the characteristics of low real-time operation complexity and high security in embedded systems. Wen et al. [12] applied the DNA sequence and the chaotic system used to encrypt the sub-view image. Due to the limited processing capacity of smart mobile devices, the encryption algorithm based on hybrid chaos takes into account both security and timeliness, so it is very suitable for medical image encryption on mobile devices.

3 Design and Implementation of ALCencryption Algorithm

3.1 Algorithm Design Idea

Shannon proposed that the basic principle of designing encryption algorithm is scrambling and diffusion, which is usually performed on the plaintext image for many times to make the image become chaotic [13–18]. DICOM images have color and gray images, endoscopy and pathological section images are color ones, while X-ray, CT, MRI and ultrasound images are gray

ones. Most of the images in DICOM are grayscale. Since the values of the three channels of RGB are equal, the grayscale images need to be fully scrambled before diffusion. Arnold mapping has the characteristics of good chaos, simple and easy to implement, so the encryption algorithm is often used to scramble the image.

Arnold mapping can only scramble $N \times N$ images, which is not applicable for most medical images. Therefore, Arnold mapping needs to be improved to make it enable to scramble $M \times N$ images, the improved equation as shown in formula (1)–(3) [19–21].

$$\begin{bmatrix} x_i' \\ y_j' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_j \end{bmatrix} mod\ (M, N) \tag{1}$$

$$x_i' = (x_i + a * y_j)\ mod\ M \tag{2}$$

$$y_j' = (b * x_i + a * b * y_j + y_j)\ mod\ N \tag{3}$$

The Image $I_{M\times N}$ is scrambled to be $I'_{M\times N}$, (x_i, y_j), (x_i', y_j') respectively which represents the positions of pixel points before and after scrambling, $i \in [1, M]$, $j \in [1, N]$, a and b are positive integers. Using formula (4)–(6), we can calculate the scrambling degree and mean value of n-order images after each Arnold mapping.

$$d(x_i, y_j) = \sqrt{(x_i - x_i')^2 + (y_j - y_j')^2} \tag{4}$$

$$E(d) = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} d(x_i, y_j) \tag{5}$$

$$SH_n = \frac{1}{n} \sum_{k=1}^{n} a_k \frac{E(d_k(x_i, y_j))}{Var(d_k(x_i, y_j))} \tag{6}$$

The $d(x_i, y_j)$ represents the pixel moving distance value, $E(d)$ represents the average value of the moving distance of the entire image, SH_n represents the n-order scrambling degree, and a_k represents the weighting coefficient, which is set according to the effect of different order distance on the scrambling degree. In other words, the smallest pixel pair is scattered as far as possible first, and then the second pixel pair is dispersed. The larger the SH_n value, the more chaotic the image pixels. When the image is scrambled by Arnold mapping, the scrambling effect of the image is related to the iteration times, and the reasonable iteration times can be selected according to the scrambling degree. When the SH_n value is the maximum, the number of iterations is the best, and so is the scrambling effect of the image. In general, the optimal scrambling number of images is around half of the Arnold transformation period, so the optimal scrambling degree can be obtained with a small amount of computation.

Therefore, when Arnold mapping scrambling is performed on DICOM images, the corresponding number of iterations can be selected according to the optimal scrambling degree, which can not only achieve the best scrambling effect of images, but also reduce the number of iterations and the time cost of scrambling. For the diffusion of DICOM images, a dual chaotic cross-diffusion method based on Logistic mapping and Chebyshev mapping proposed by the author in literature [20,21] was used.

Combining the characteristics of mobile platform and DICOM image, we proposed a medical image encryption algorithm ALCencryption based on hybrid-chaotic mapping by using improved Arnold mapping, Logistic mapping and Chebyshev mapping. The algorithm first analyzes the real image part of DICOM image and distinguishes whether the image is color one or gray one. For DICOM gray scale image, the improved Arnold mapping is adopted to select the best iteration times according to the maximum scrambling degree, so as to achieve the best scrambling effect of DICOM image, while reducing the iteration times and the time cost of scrambling. Then, key generated by Logistic mapping and Chebyshev mapping cross iteration was used to spread the pixel position in the way of forward and backward cross diffusion based on parity, and the image encryption was completed. For DICOM color images, double chaotic cross diffusion encryption algorithm based on Logistic mapping and Chebyshev mapping proposed in literature [20,21] was used to realize encryption. The algorithm flow chart is shown in Fig. 1.

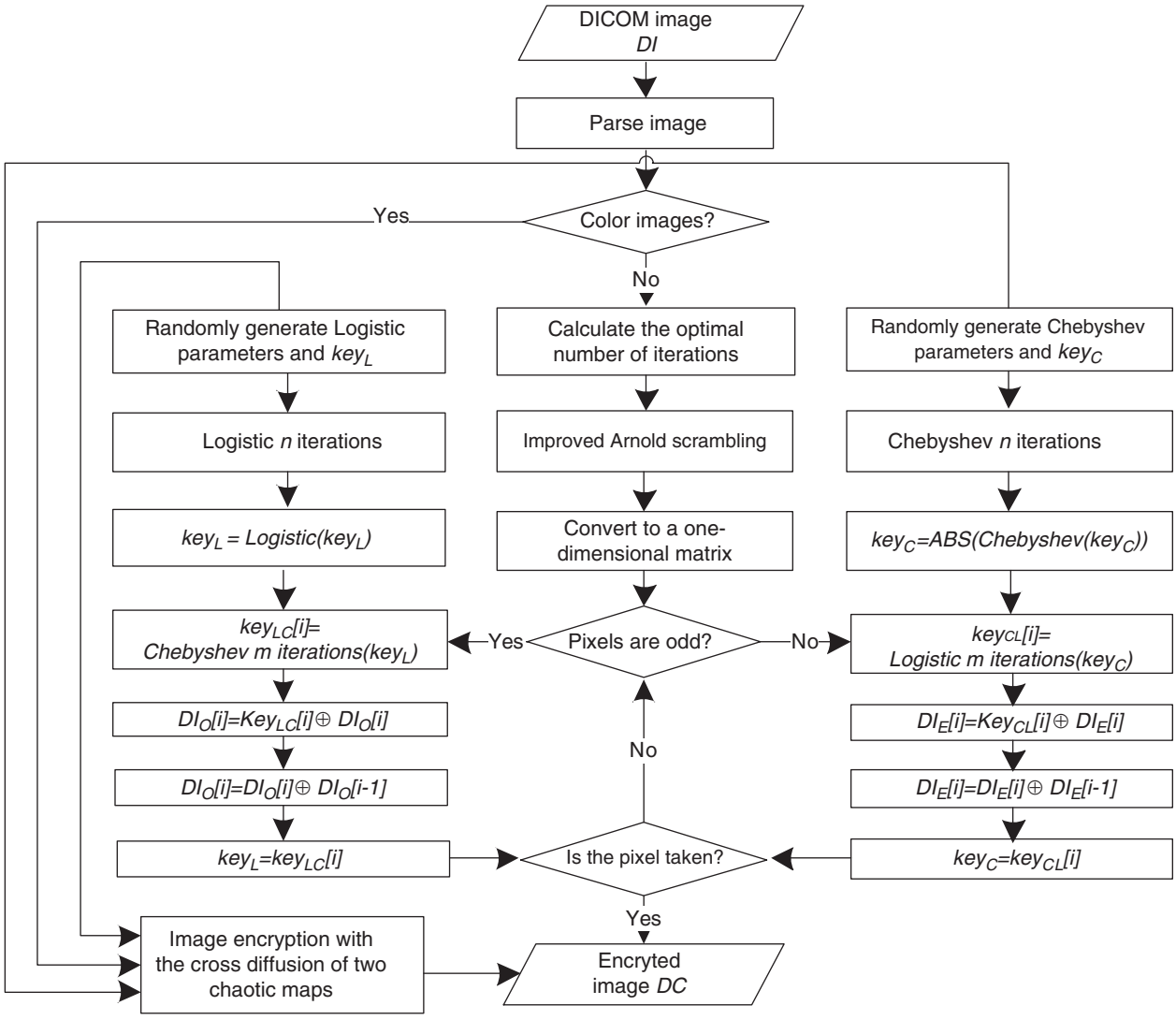

Figure 1: ALCencryption algorithm flow chart

3.2 ALCencryption Algorithm Implementation

The ALCencryption algorithm implementation includes the following steps:

Step 1: Input DICOM image DI;
Step 2: Analyze the DICOM image to get information such as the pixel value, image size, and image type (color or grayscale image);
Step 3: Parameters μ ($\mu \in (3.5699456, 4]$) of Logistic mapping and k ($k \geq 2$) of Chebyshev mapping are generated randomly;
Step 4: Generate random keys key_L ($key_L \in (0, 1)$) and key_C ($key_C \in [-1, 1]$), where key_L is the initial key for Logistic mapping iteration, and key_C is the initial key for Chebyshev mapping iteration;
Step 5: If the DI is a color image, encrypt it using μ, k, key_L, key_C and an image encryption algorithm based on cross diffusion of double chaotic maps, then go to Step (12);
Step 6: If DI is a grayscale image, use improved Arnold mapping to scramble its pixel position according to the optimal number of iterations, and then convert it into a one-dimensional matrix DI $[0, M * N - 1]$;
Step 7: Take key_L as the initial key, use Logistic mapping to iterate n times (eliminate the influence of transient, use it for 80 times in the experiment), and then use Logistic mapping to iterate for 1 time to save the result in key_L, which is $key_L = Logistic\,(key_L)$, as the initial key of Chebyshev mapping;
Step 8: Take key_C as the initial key, iterate n times with Chebyshev mapping (eliminate the influence of transient, and use it for 80 times in subsequent experiments), take the absolute value of the result of one iteration with Chebyshev mapping and save it in key_C, that is, $key_C = ABS\,(Chebyshev\,(key_C))$, as the initial key of Logistic mapping;
Step 9: If $DI_O[i]$ (the current pixel position) is odd, then key_L is iterated m times with Chebyshev (20 times in experiments) to produce the encryption key $key_{LC}[i] = Chebyshev\,(key_L)$; Use the $Key_{LC}[i]$ to encrypt the $DI_O[i]$, $DI_O[i] = Key_{LC}[i] \oplus DI_O[i]$; If $DI_O[I]$ is the first pixel, this operation is skipped, otherwise the following operations are performed: $DI_O[i] = DI_O[i] \oplus DI_O[i-1]$, $key_L = key_{LC}[i]$;
Step 10: If $DI_E[i]$ is even, then key_C is iterated m times with Logistic (20 times in experiments) to produce the encryption key $key_{CL}[i] = Logistic\,(key_C)$; Use the $key_{CL}[i]$ to encrypt the $DI_E[i]$, $DI_E[i] = Key_{CL}[i] \oplus DI_E[i]$; If $DI_E[i]$ is the first pixel, this operation is skipped, otherwise the following operations are performed: $DI_E[i] = DI_E[i] \oplus DI_E[i-1]$, $key_C = key_{CL}[i]$;
Step 11: If all the pixels in DI have been traversed, skip to Step (12), otherwise repeat Steps (9)–(10);
Step 12: Output ciphertext image DC.

3.3 ALCencryption Algorithm Description

Fig. 2 shows the ALCencryption described in C++ language.

3.4 Algorithmic Complexity

In the C++ language description of the ALCencryption algorithm, the time complexity of statements 13, 14 and 15–21 is $O(n)$, while the time complexity of statements 24–41 is $O(n * n)$, so the time complexity of the ALCencryption algorithm is $T(n) = O(n + n + n + n * n) = O(n^2)$. The space complexity of the ALCencryption algorithm is the memory space required to store the array of image pixels, that is, $S(n) = O(n)$.

ALCencryption Algorithm

```
Input: DI
Output: DC
Prase(DI);
KeyL=randomKey();
KeyC=randomKey();
μ=randomRef();
k=randomRef();
if(DI.isColor)
{
    DC=doubleChaosEcryption(DI);
    return;
}
else
{
    n=arnoldBestIterations();
    DC=improvedArnoldMap(DI,n);
    for(int i=0;i<n;i++)
    {
        Logistic=μ* KeyL *(1- KeyL);
        KeyL=Logistic;
        Chebyshev=cos(k*arcos(KeyC));
        KeyC=Chebyshev;
    }
    KeyL=μ* KeyL *(1- KeyL);
    KeyC= fabs(cos(k*arcos(KeyC)));
    for(int i=1;j<=M*N-1;j++)
      {
        if(j%2==1)
        {
          for(int j=0;i<m;i++)
            KeyLC(i)= cos(k*arcos(KeyL));
          DC(i)= KeyLC[i] ^DC(i);
          DC(i)= DC(i) ^DC(i-1);
        }
        else
        {
          for(int j=0;i<m;i++)
            KeyCL(i)= KeyC *μ*(1- KeyC);
          DC(i)= KeyCL[i] ^DC(i);
          DC(i)= DC(i) ^DC(i-1);
        }
    }
}
Display(DC);
```

Figure 2: ALCencryption algorithm description

4 Experimental Results and Analysis

The proposed ALCencryption algorithm is implemented in OpenCV 2.4, MATLAB R2014b, DCMTK in Windows 7 platform. The system configuration includes Intel core i5-7200 processor operating at 2.5 GHz and 4 GB RAM.

We comprehensively considered the image type, image size, image bit depth and other factors, and selected 10 medical images downloaded from http://www.barre.nom.fr/medical/samples/ as test samples, as shown in Tab. 1. Fig. 3 represents the sample DICOM images and Fig. 4 represents the corresponding encrypted ones.

Table 1: Description of DICOM image samples

ID	File name	Size	RGB/ monochrome	Bit Depth	Modality
D_CT_1	CT-MONO2-8-abdo	512 * 512	MONOCHROME	8	Computed tomography
D_CT_2	CT-MONO2-12-lomb	512 * 512	MONOCHROME	12	Computed tomography
D_OT_3	OT-MONO2-8-hip	512 * 512	MONOCHROME	8	Other
D_MR_4	MR-MONO2-12-angio	256 * 256	MONOCHROME	12	Magnetic resonance
D_MR_5	MR-MONO2-16-head	256 * 256	MONOCHROME	16	Magnetic resonance
D_MR_6	MRI-MONO2-16-Knee	256 * 256	MONOCHROME	16	Magnetic resonance
D_MR_7	MRI2-MONO2-16	512 * 512	MONOCHROME	16	Magnetic resonance
D_CR_8	CR-MONO1-10-chest	440 * 440	MONOCHROME	10	Computed radiography
D_US_9	US-RGB-8-epicard	640 * 480	RGB	8	Ultrasound
D_US_10	US-RGB-8-esopscho	256 * 120	RGB	8	Ultrasound

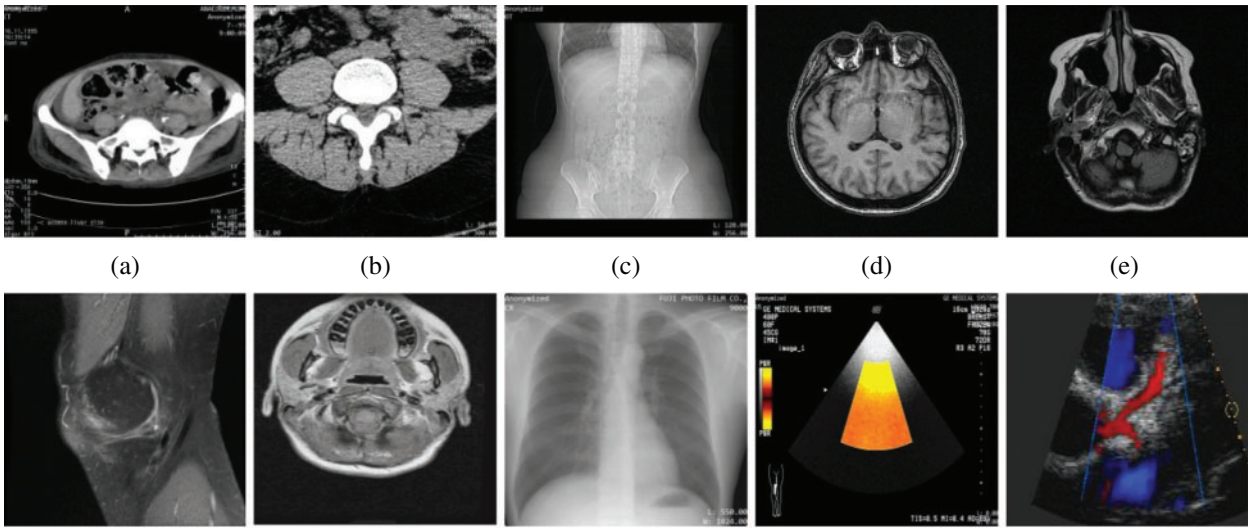

Figure 3: DICOM image samples. (a) D_CT_1 (b) D_CT_2 (c) D_OT_3 (d) D_MR_4 (e) D_MR_5 (f) D_MR_6 (g) D_MR_7 (h) D_CR_8 (i) D_US_9 (j) D_US_10

4.1 Histogram Analysis

The histogram of a DICOM image is a function of image brightness level, which describes the number of pixels and frequency of each brightness level [22]. The abscissa of the histogram is the brightness level, and the ordinate is the frequency or number of pixels of the brightness level.

During the period of encryption, image pixel values vary widely due to the diffusion effect, the histograms of plain and encrypted DICOM images in Figs. 5 and 6 are completely different from each other. Since the histogram of the encrypted image is evenly distributed without revealing any information about the image, it can be concluded that the proposed ALCencryption algorithm can completely resist the attack of statistical analysis.

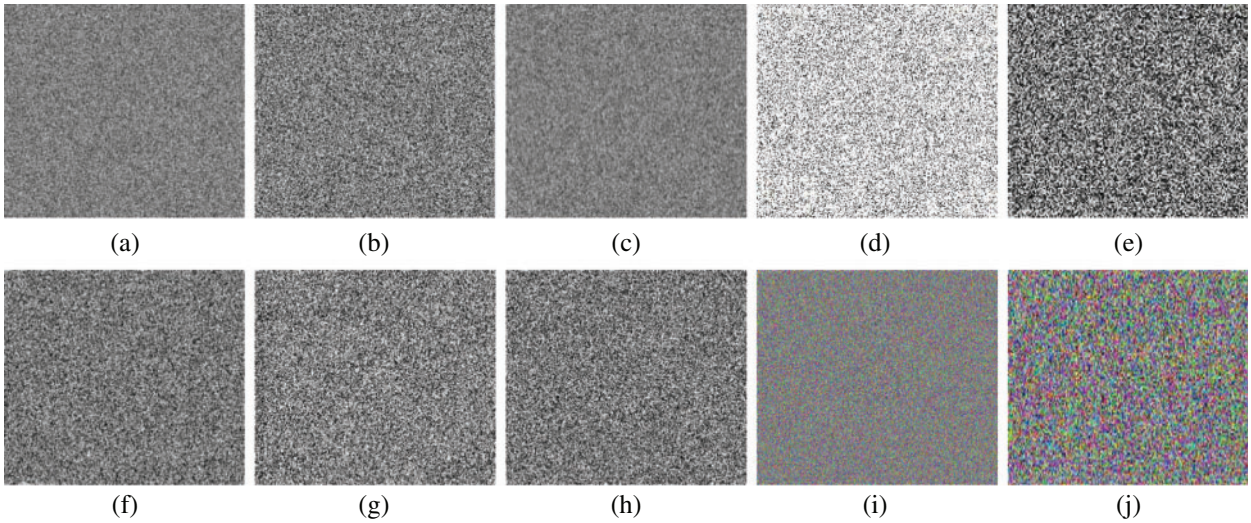

Figure 4: Encrypted DICOM image samples. (a) ED_CT_1 (b) ED_CT_2 (c) ED_OT_3 (d) ED_MR_4 (e) ED_MR_5 (f) ED_MR_6 (g) ED_MR_7 (h) ED_CR_8 (i) ED_US_9 (j) ED_US_10

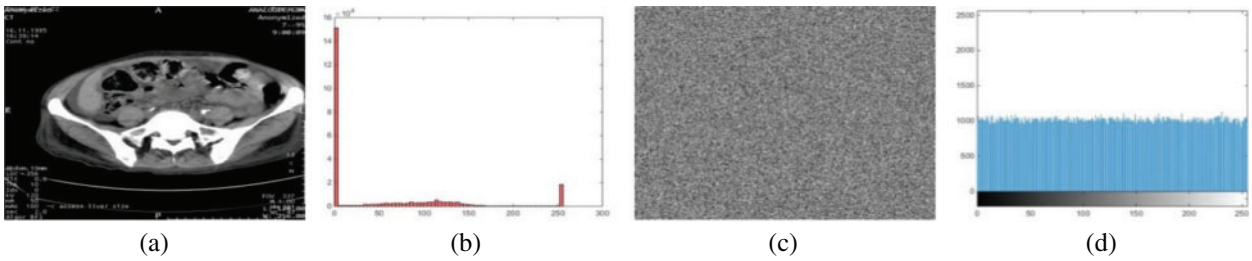

Figure 5: Histograms of plain (D_CT_1) and encrypted DICOM sample (ED_CT_1). (a) D_CT_1. (b) Histogram of D_CT_1. (c) ED_CT_1. (d) Histogram of ED_CT_1

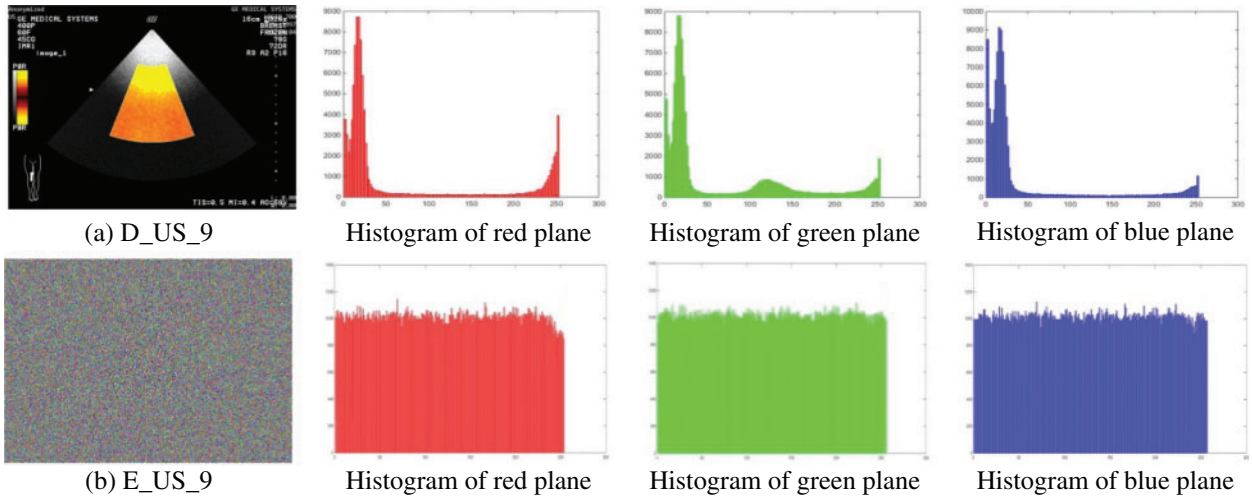

Figure 6: (a) Histograms of red, green and blue planes of plain DICOM image sample: D_US_9. (b) Histograms of red, green, and blue planes of encrypted DICOM image sample: ED_US_9

4.2 Pixel Correlation Analysis

The correlation coefficients of two adjacent pixels reflect the statistical characteristics of the image [23]. Therefore, the correlation coefficient of ciphertext image should be close to zero to withstand the statistical attack [24]. We encrypted the D_CT_1 grayscale medical image and D_US_9 color medical image, randomly selected 3000 pairs of adjacent pixel points from the plaintext image and ciphertext image, and calculated the correlation coefficient of adjacent pixels in the horizontal, vertical and diagonal directions according to formula (7)–(11).

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{7}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{8}$$

$$\sqrt{D(x)} \neq 0 \quad \text{and} \quad \sqrt{D(y)} \neq 0 \tag{9}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{10}$$

$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{11}$$

The correlation distribution of original and encrypted DICOM image sample is illustrated in Fig. 7. From Tab. 2, we see that adjacent pixels are strong correlated in the original DICOM image and almost irrelated in the encrypted DICOM image, our method shows smaller pixel correlation coefficients than other methods in the literature [6,25,26], which is higher than that in the literature [27]. It could be inferred that the proposed algorithm breaks the correlation between the pixels in each direction, and the statistical characteristics of the original image are randomly diffused to the cipher image, which can safely resist statistical analysis.

4.3 Information Entropy Analysis

The closer the information entropy is to the image bit depth, the better the randomness of the image pixel is. Using formula (12), we can calculate the image information entropy of the original image and the encrypted image, and the calculated results are shown in Tab. 3. Our encrypted DICOM images have larger entropy than the original one and method [6,26–28], which is equal to the information entropy of the method [27], which means the encrypted image is more random and more secure.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \tag{12}$$

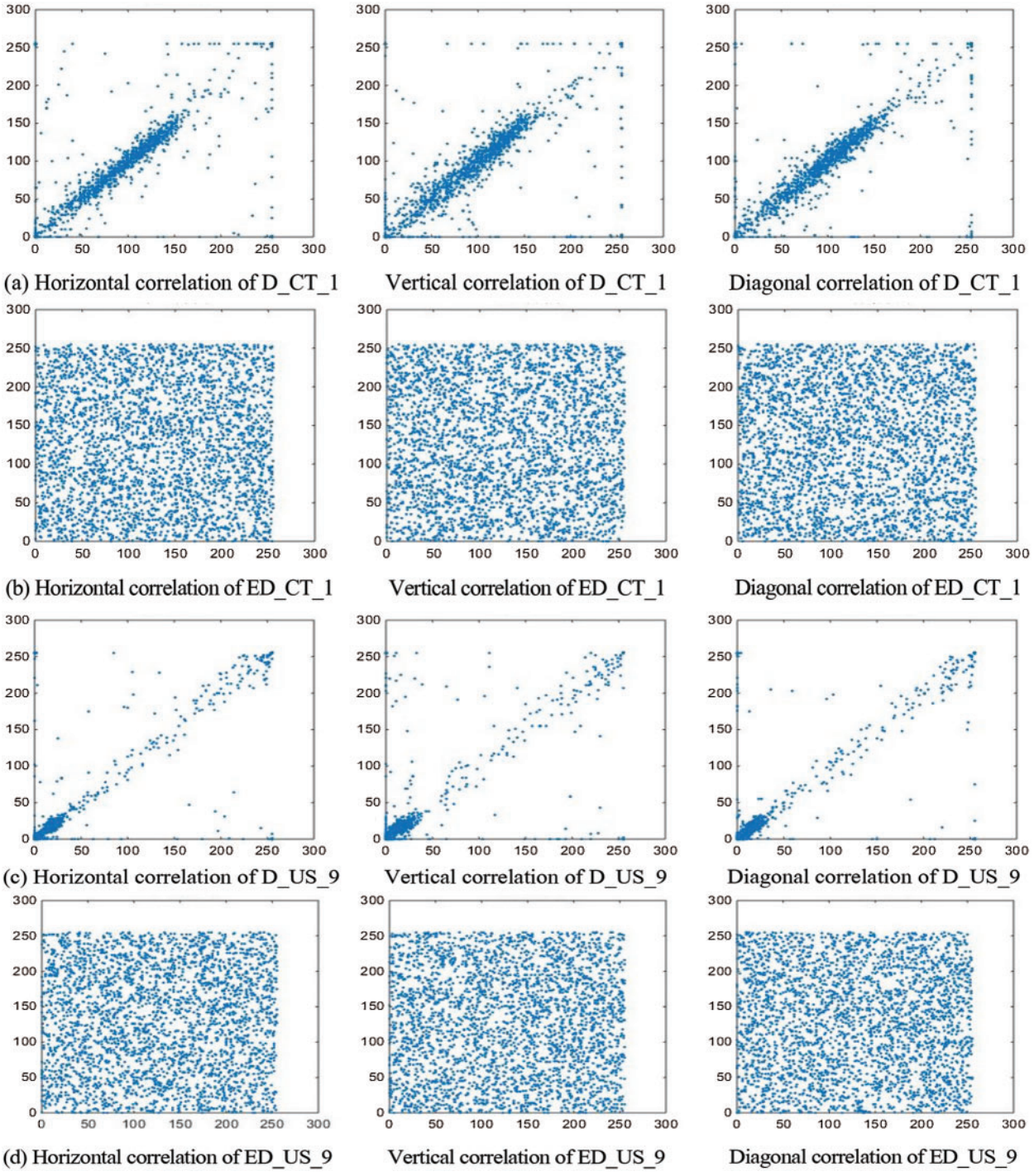

Figure 7: Correlation distributions of plain and encrypted DICOM image sample: D_CT_1, ED_-CT_1, D_US_9 and ED_US_9

Table 2: The comparison of correlation coefficient (absolute value)

Image	Method	Plain DICOM image			Encrypted DICOM image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
D_CT_1	[6]	0.91315	0.87131	0.00574	0.00757	0.01428	0.00614
	[25]	0.91359	0.87192	0.00576	0.00764	0.01429	0.00637
	[27]	0.93700	0.89490	0.90090	0.00250	0.00290	0.00270
	[26]	–	–	–	0.01960	0.01780	0.01690
	Ours	0.91382	0.87216	0.00568	0.00432	0.01407	0.00463
D_OT_3	[6]	0.98681	0.98927	0.98005	0.00145	0.00093	0.00652
	[25]	0.98685	0.98939	0.98036	0.00159	0.00097	0.00714
	Ours	0.98663	0.98902	0.98061	0.00123	0.00096	0.00628
D_MR_5	[26]	–	–	–	0.01590	0.01620	0.01680
	Ours	0.93894	0.19036	0.18949	0.00668	0.00519	0.00933
D_MR_6	[6]	0.98716	0.99375	0.98173	0.00152	0.00176	0.00309
	[25]	0.98720	0.99397	0.98287	0.00141	0.00109	0.00256
	Ours	0.98733	0.99389	0.98291	0.00138	0.00046	0.00136
D_MR_7	[6]	0.97255	0.97657	0.95534	0.00002	0.00247	0.00291
	[25]	0.97253	0.97653	0.95535	0.00002	0.00249	0.00307
	Ours	0.97257	0.97660	0.95532	0.00002	0.00245	0.00253
D_CR_8	[26]	–	–	–	0.01460	0.01940	0.01950
	Ours	0.99625	0.42717	0.42733	0.01706	0.00561	0.00806
D_US_9	[26]	–	–	–	0.01530	0.01530	0.01460
	Ours	0.76863	0.78760	0.64948	0.01035	0.01641	0.01118
D_CT_2	Ours	0.98065	0.34379	0.33430	0.01253	0.00148	0.00371
D_MR_4	Ours	0.19565	0.03768	0.00126	0.00290	0.00453	0.00097
D_US_10	Ours	0.25302	0.05379	0.07688	0.00460	0.02043	0.02043

Table 3: The information entropy analysis

Image	Plain DICOM image	Encrypted DICOM image				
		[6]	[25]	[27]	[26]	Ours
D_CT_1	3.73685	7.97881	7.78231	7.99932	7.86000	7.99932
D_OT_3	6.40026	7.99570	7.96483	–	–	7.99686
D_MR_5	6.91114	14.76907	15.15261	–	–	15.17216
D_MR_6	7.08823	15.32154	15.17716	–	–	15.80776
D_MR_7	10.44122	15.62353	15.80773	–	–	15.88637

4.4 Key Sensitivity Analysis

Key sensitivity is an important index to measure the strength of encryption algorithm. For an effectively encrypted system, a small change in any one key should result in a completely different output. We take K_1 as the correct key, K_1 and K_2 parameters were set as follows: $K_1 = Key_C = 0.678$, $Key_L = 0.567$, $\mu = 3.894762$, $k = 3.0$, $a = 32$, $b = 21$, $K_2 = Key_C = 0.678 + 10^{-16}$,

$Key_L = 0.567$, $\mu = 3.894762$, $k = 3.0$, $a = 32$, $b = 21$. The waveform of key changes when decrypted D_MR_5 image using K_1 and K_2 is shown in Fig. 8. It can be seen from the waveform diagram that the difference of key is obvious when the key changes only 10^{-16}.

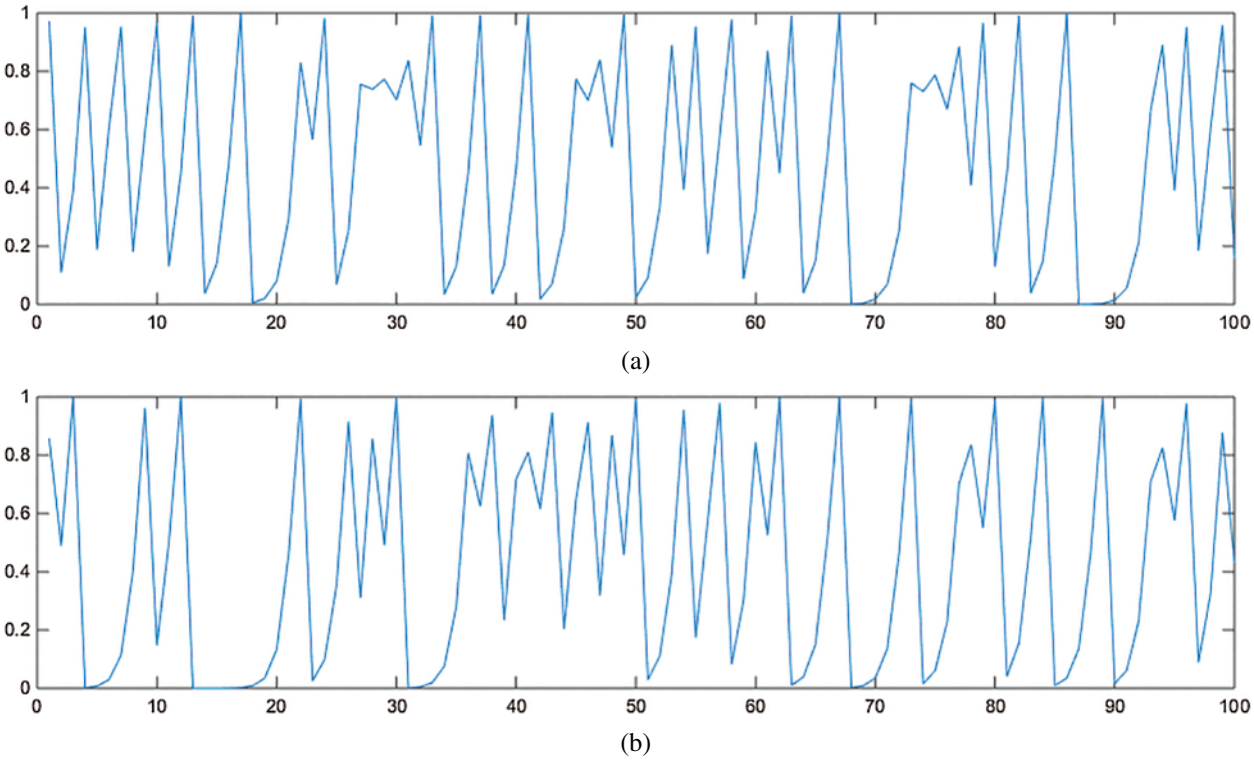

Figure 8: Waveform diagram of key changes when decrypted D_MR_5 image using K_1 and K_2. (a) $K_1 = Key_C = 0.678$, $Key_L = 0.567$, $\mu = 3.894762$, $k = 3.0$, $a = 32$, $b = 21$. (b) $K_2 = Key_C = 0.678 + 10^{-16}$, $Key_L = 0.567$, $\mu = 3.894762$, $k = 3.0$, $a = 32$, $b = 21$

4.5 Key Space Analysis

If the key space is large enough, then the cryptographic system can resist violent attacks. Our proposed algorithm uses 10 keys and controlled parameters, namely x_i, x_j, a, b, k, μ, Key_L, Key_C, Key_{LC} and Key_{CL} are computed in the accuracy of 10^{-16}. So the total key space is $(10^{16})^{10} = 10^{160}$. Our key space is larger than the existing works [6,25,26] and smaller than literature [25,27], which is sufficiently large to resist all presently known brute-force attacks, as shown in Tab. 4.

4.6 Differential Attack Analysis

The sensitivity of encryption algorithms to differential attacks can be quantitatively evaluated by NPCR (number of pixels change rate) and UACI (unified average changing intensity). We compute those two values from encrypted DICOM image samples according to formula (13), (14) as shown in Tab. 5. In general, the larger the value of NPCR and UACI, the better the sensitivity of the algorithm. Our method is superior to the literature [6,22,27] in both NPCR and UACI and

lower than literature [26] in UACI of some samples from Tab. 5, hence it can effectively resist differential attacks.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \tag{13}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \tag{14}$$

where

$$D(i,j) = \begin{cases} 1, & C_1(i,j) \neq C_2(i,j) \\ 0, & C_1(i,j) = C_2(i,j) \end{cases}$$

Table 4: Key space analysis

Method	Key space
[6]	10^{128}
[25]	10^{384}
[27]	2^{716}
[26]	10^{120}
Ours	10^{160}

Table 5: The NPCR and UACI of encrypted DICOM image samples

DICOM image	NPCR (%)					UACI (%)				
	[6]	[25]	[27]	[26]	Ours	[6]	[25]	[27]	[26]	Ours
D_CT_1	99.6240	99.8756	99.6173	99.7	99.9987	33.4712	33.3961	33.4756	33.7	33.4798
D_CT_2	99.6154	99.9824	–	–	99.9992	33.1567	33.3482		–	33.3561
D_OT_3	99.5777	99.5920	–	–	99.6875	33.0944	33.3259		–	33.3266
D_MR_4	99.2360	99.9765	–	–	99.9912	33.4301	33.4767	–	–	33.7103
D_MR_5	99.8947	99.9813	–	99.7	99.9964	33.2369	33.2903	–	33.55	33.3087
D_MR_6	99.5623	99.9992	–	–	99.9997	33.3537	33.3420	–	–	33.3815
D_MR_7	99.7545	99.9984	–	–	99.9998	33.3485	33.3526	–	–	33.3612
D_CR_8	99.8992	99.9971	–	99.8	99.9995	33.2976	33.3178	–	33.29	33.3258
D_US_9	99.6076	99.9887	–	99.6	99.9991	33.9995	33.7153	–	33.57	33.9996
D_US_10	99.5822	99.9899	–	–	99.9993	33.5912	33.4934	–	–	33.6105

4.7 Chosen-Plaintext Attack

As we all know, once the cryptosystem has the ability to resist selected plaintext attacks, then it can also resist known plaintext and known ciphertext attacks. With the same key set and plaintext image, different ciphertext images are usually generated when the encryption algorithm is repeated for different times. We used the same key set and algorithm to encrypt D_OT_3 DICOM

image twice to obtain two ciphertext images in Fig. 9. The first and second encrypted D_OT_3 DICOM image are referred to as DE_1 (Fig. 9b) and DE_2 (Fig. 9c), respectively. In order to prove the difference between the two images, we calculate $|DE_1\text{-}DE_2|$ use pixel-to-pixel difference. The difference image and its histogram (Figs. 9d and 9h) illustrate that the two cipher images obtained by changing the encryption times are completely different. It shows that our algorithm can resist the Chosen-Plaintext Attack.

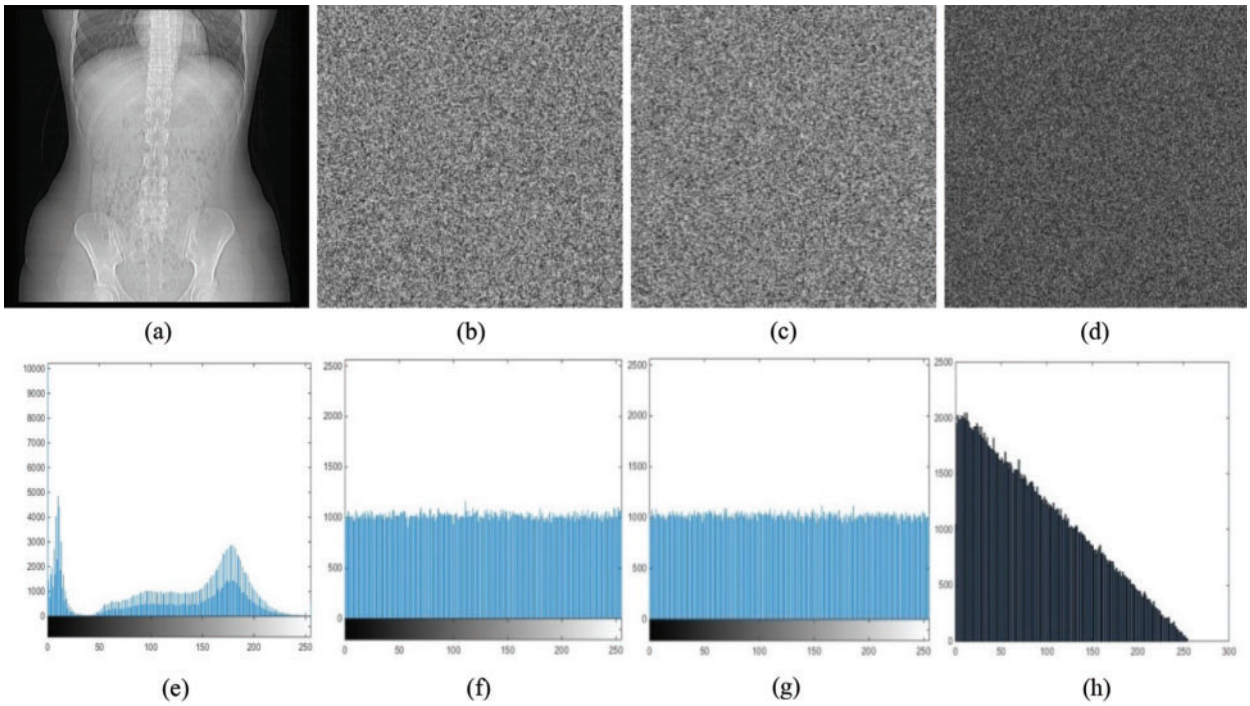

Figure 9: Chosen plain text attack analysis: (a, e) the D_OT_3 image and its histogram, (b, f) the first encrypted D_OT_3 image and its histogram, (c, g) the second encrypted image and its histogram, (d, h) difference image (b, c) and its histogram

4.8 Time Analysis

Under the same operating environment of Windows platform, we encrypted the DICOM sample images in Tab. 1 for 100 times to calculate the average encryption time. The results are shown in Tab. 6. As can be seen from Tab. 6, the ALCencryption algorithm is obviously superior to the algorithm in literature [6,26] in encryption and decryption time. With the increase of image pixels, the key generation time and encryption time of the ALCencryption algorithm increase slowly, because scrambling and key generation are carried out at the same time, the optimal iteration times are selected to reduce the time cost of scrambling, and the number of Logistic and Chebyshev mapping iterations did not increase.

We applied the ALCencryption algorithm to the Android platform to analyze the encryption and decryption time of DICOM images. The hardware and software required for the experiment

include Android Smartphone platform, Kirin 960 processor operating at 2.4 GHz, 4 GB RAM, and Android 9.0 OS. In order to verify the efficiency of our algorithm applicable to the Android platform, 17 medical images of different sizes were encrypted for 100 times in the same operating environment, and the average time of encryption and decryption was calculated. The calculated results are shown in Tab. 7. As can be seen from Tab. 7, the encryption and decryption time is within 10 ms if the image size is less than 100 KB. When the image size is 512 KB, the average encryption time is 206.6 ms and the average decryption time is 191.1 ms. It can be seen from the above data that the encryption and decryption time does not increase as the image size doubles. When the image size increased from 256 K to 2.2 MB, the encryption time of 2.2 MB image size was 557 ms, and the decryption time was 544 ms. The 2.2 MB image size was 8.8 times of the 256 KB image size, the encryption time was 4.64 times of the average encryption time of the 256 KB image, and the decryption time was 5.34 times of the average decryption time of the 256 KB image. When the image size increases from 512 KB to 2.8 MB, the encryption time of 2.8 MB image size is 693 ms, and the decryption time is 615 ms. The size of 2.8 MB image is 5.6 times of that of 512 KB image, the encryption time is 3.35 times of the average encryption time of 512 KB image, and the decryption time is 3.22 times of the average decryption time of 512 KB image. From the above data, it can be seen that the encryption and decryption time required by the algorithm increases significantly, but the encryption and decryption time is less than 0.7 s. Since the size of most medical images is less than 3 M, the speed advantage of this algorithm is obvious. When the image size is 12.5 MB and 15.3 MB, the encryption time needs 4.51 s and 5.58 s respectively, and the decryption time needs 3.86 s and 4.79 s respectively, which increases significantly, mainly because the image pixel matrix gets larger, the scrambling period of Arnold mapping increases, and the xor operation of pixel point diffusion increases. Although images of more than 10 MB are relatively few in actual use, the algorithm encryption and decryption speed is also fast for mobile phones with weak computing power.

Table 6: Time analysis on Windows platform

DICOM image	Size	Encryption time (s)			Decryption time (s)	
		[6]	[26]	Ours	[26]	Ours
D_CT_1	256 KB	4.5477	0.24	0.00197	0.27	0.00201
D_CT_2	512 KB	4.5235	–	0.00621	–	0.00613
D_OT_3	256 KB	4.5020	–	0.00186	–	0.00182
D_MR_4	96 K	4.0921	–	0.00221	–	0.00219
D_MR_5	128 K	4.2136	0.24	0.00177	0.26	0.00175
D_MR_6	128 K	4.2204	–	0.00180	–	0.00181
D_MR_7	512 KB	4.5268	–	0.00203	–	0.00205
D_CR_8	378 K	4.3699	0.2	0.00458	0.23	0.00463
D_US_9	900 K	4.5290	0.25	0.00672	0.21	0.00667
D_US_10	90 K	4.0818	–	0.00073	–	0.00071

Table 7: Time analysis on Android platform

Size	ID	Encryption time (s)	Decryption time (s)
90 KB	D_US_10	0.008987	0.007249
96 KB	D_MR_4	0.009150	0.007804
128 KB	D_MR_5	0.057595	0.054026
128 KB	D_MR_6	0.065336	0.063188
256 KB	D_CT_1	0.117701	0.098303
256 KB	D_OT_3	0.122349	0.105147
378 KB	D_CR_8	0.147434	0.116381
512 KB	D_CT_2	0.188416	0.164163
512 KB	D_MR_7	0.199622	0.168753
512 KB	D_CT_11	0.223189	0.248548
512 KB	D_CT_12	0.229456	0.195306
512 KB	D_CT_13	0.192591	0.178512
900 KB	D_US_9	0.270036	0.224905
2.2 MB	D_CR_14	0.556723	0.543656
2.8 MB	D_CR_15	0.693240	0.614558
12.5 MB	D_CR_16	4.506174	3.864400
15.3 MB	D_DX_17	5.581097	4.789767

5 Conclusions

In order to ensure the security of medical image in storage and network transmission, we proposed a hybrid mapping algorithm of ALCencryption for medical image encryption based on Logistic, Chebyshev and improved Arnold mapping. It is proved that the algorithm has high security from the aspects of encryption and decryption effect, histogram analysis, pixel correlation analysis, information entropy analysis, key sensitivity, key space, differential attack and selective plaintext attack. According to the optimal scrambling degree of Arnold mapping, the iteration times are determined, which does not only reduce the iteration frequency and time, but also makes the image scrambling effect the best. In future, the proposed algorithm will be implemented on Picture Archiving and Communication Systems to ensure the safe and efficient transmission of DICOM images.

Funding Statement: This work is partly supported by the Scientific Research Fund of Hunan Provincial Education Department (19B082), the Science and Technology Development Center of the Ministry of Education-New Generation Information Technology Innovation Project (2018A02020), the research supported by Science Foundation of Hengyang Normal University (19QD12), the Science and Technology Innovation Program of Hunan Province (2016TP1020), the Application-oriented Special Disciplines, Double First-Class University Project of Hunan Province (Xiangjiaotong [2018] 469), the Hunan Province Special Funds of Central Government

for Guiding Local Science and Technology Development (2018CT5001), the Subject Group Construction Project of Hengyang Normal University (18XKQ02).

Conflict of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Hu, J., Han, F. (2009). A pixel-based scrambling scheme for digital medical images protection. *Journal of Network and Computer Applications, 32(4),* 788–794. DOI 10.1016/j.jnca.2009.02.009.
2. Sathishkumar, G. A., Bhoopathybagan, K., Sriraam, S., Venkatachalam, P., Vignesh, R. (2011). A novel image encryption algorithm using two chaotic maps for medical application. *International Conference on Computer Science and Information Technology*, pp. 290–299. Berlin, Heidelberg: Springer.
3. Kanso, A., Ghebleh, M. (2015). An efficient and robust image encryption scheme for medical applications. *Communications in Nonlinear Science and Numerical Simulation, 24(1),* 98–116. DOI 10.1016/j.cnsns.2014.12.005.
4. Fu, C., Meng, W. H., Zhan, Y. F., Zhu, Z. L., Lau, F. C. M. (2013). An efficient and secure medical image protection scheme based on chaotic maps. *Computers in Biology and Medicine, 43(8),* 1000–1010. DOI 10.1016/j.compbiomed.2013.05.005.
5. Fu, C., Zhang, G. Y., Bian, O., Lei, W., Ma, H. (2014). A novel medical image protection scheme using a 3-dimensional chaotic system. *PLoS One, 9(12),* e115773. DOI 10.1371/journal.pone.0115773.
6. Chandrasekaran, J., Thiruvengadam, S. J. (2017). A hybrid chaotic and number theoretic approach for securing DICOM images. *Security and Communication Networks, 2017(3),* 1–12.
7. Seyedzadeh, S. M., Mirzakuchaki, S. (2012). A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Processing, 92(5),* 1202–1215. DOI 10.1016/j.sigpro.2011.11.004.
8. Dai, Y., Wang, X. (2012). Medical image encryption based on a composition of logistic maps and chebyshev maps. *IEEE International Conference on Information and Automation*, pp. 210–214, Shenyang, China.
9. Zhou, Y., Bao, L., Chen, C. P. (2014). A new 1D chaotic system for image encryption. *Signal Processing, 97,* 172–182. DOI 10.1016/j.sigpro.2013.10.034.
10. Ravichandran, D., Praveenkumar, P., Rayappan, J. B. B., Rengarajan, A. (2016). Chaos based crossover and mutation for securing DICOM image. *Computers in Biology and Medicine, 72,* 170–184. DOI 10.1016/j.compbiomed.2016.03.020.
11. Boussif, M., Aloui, N., Cherif, A. (2017). Smartphone application for medical images secured exchange based on encryption using the matrix product and the exclusive addition. *IET Image Processing, 11(11),* 1020–1026. DOI 10.1049/iet-ipr.2017.0229.
12. Wen, W., Wei, K., Zhang, Y., Yu, M. F., Ming, L. (2020). Colour light field image encryption based on DNA sequences and chaotic systems. *Nonlinear Dynamics, 99(2),* 1–14. DOI 10.1007/s11071-019-05378-8.
13. Yu, S. S., Zhou, N. R., Gong, L. H., Nie, Z. (2020). Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system. *Optics and Lasers in Engineering, 124,* 105816. DOI 10.1016/j.optlaseng.2019.105816.
14. Gong, L., Qiu, K., Deng, C., Zhou, N. (2019). An optical image compression and encryption scheme based on compressive sensing and RSA algorithm. *Optics and Lasers in Engineering, 121,* 169–180. DOI 10.1016/j.optlaseng.2019.03.006.
15. Zhou, N., Jiang, H., Gong, L., Xie, X. (2018). Double-image compression and encryption algorithm based on co-sparse representation and random pixel exchanging. *Optics and Lasers in Engineering, 110,* 72–79. DOI 10.1016/j.optlaseng.2018.05.014.

16. Zhou, N., Yan, X., Liang, H., Tao, X., Li, G. (2018). Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system. *Quantum Information Processing, 17(12),* 338. DOI 10.1007/s11128-018-2104-6.
17. Li, M., Lu, D., Xiang, Y., Zhang, Y., Ren, H. (2019). Cryptanalysis and improvement in a chaotic image cipher using two-round permutation and diffusion. *Nonlinear Dynamics, 96(1),* 31–47. DOI 10.1007/s11071-019-04771-7.
18. Jiao, G., Li, L., Zou, Y. (2019). Improved security for android system based on multi-chaotic maps using a novel image encryption algorithm. *International Journal of Performability Engineering, 15(6),* 1692–1701.
19. Das, R., Baykara, M., Tuna, G. (2019). A novel approach to steganography: Enhanced least significant bit substitution algorithm integrated with self-determining encryption feature. *Computer Systems Science and Engineering, 34(1),* 23–32.
20. Jiao, G., Peng, X. J., Duan, K. W. (2019). Image encryption with the cross diffusion of two chaotic maps. *TIIS, 13(2),* 1064–1079.
21. Jiao, G., Zhou, S., Li, L., Zou, Y. (2019). Hybrid chaotic encryption algorithm for securing DICOM systems. *International Journal of Performability Engineering, 15(5*), 1436–1444.
22. Liu, G., Li, J., Liu, H. (2014). Chaos-based color pathological image encryption scheme using one-time keys. *Computers in Biology and Medicine, 45,* 111–117. DOI 10.1016/j.compbiomed.2013.11.010.
23. Dzwonkowski, M., Papaj, M., Rykaczewski, R. (2015). A new quaternion-based encryption method for DICOM images. *IEEE Transactions on Image Processing, 24(11),* 4614–4622. DOI 10.1109/TIP.2015.2467317.
24. Parvees, M. M., Samath, J. A., Bose, B. P. (2016). Secured medical images a chaotic pixel scrambling approach. *Journal of Medical Systems, 40(11),* 232. DOI 10.1007/s10916-016-0611-5.
25. Chen, J. X., Zhu, Z. L., Fu, C., Zhang, L. B., Zhang, Y. S. (2015). An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach. *Communications in Nonlinear Science and Numerical Simulation, 23(1–3),* 294–310. DOI 10.1016/j.cnsns.2014.11.021.
26. Prema, T. A., Sumangala, B. (2020). Selective medical image encryption using DNA cryptography. *Information Security Journal: A Global Perspective, 29(2),* 91–101. DOI 10.1080/19393555.2020.1718248.
27. Belazi, A., Talha, M., Kharbech, S., Xiang, W. (2019). Novel medical image encryption scheme based on chaos and DNA encoding. *IEEE Access, 7,* 36667–36681.
28. Nejad, M. B., Shiri, M. E. (2019). A new enhanced learning approach to automatic image classification based on SALP swarm algorithm. *Computer Systems Science and Engineering, 34(2),* 91–100.