Image Encryption with The Cross Diffusion of Two Chaotic Maps

Ge Jiao[1,2*], Xiaojiang Peng[3], Kaiwen Duan[2]

[1]School of Environment Protection and Safety Engineering, University of South China
Hengyang, China

[2]College of Computer Science and Technology, Hengyang Normal University
Hengyang, China
[E-mail: jiaoge@126.com]

[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[E-mail: xj.peng@siat.ac.cn]
*Corresponding author: Ge Jiao

Received January 24, 2018; revised August 20, 2018; accepted October 9, 2018; published February 28, 2019

Abstract

Information security has become increasingly important with the rapid development of mobile devices and internet. An efficient encryption system is a key to this end. In this paper, we propose an image encryption method based on the cross diffusion of two chaotic maps. We use two chaotic sequences, namely the Logistic map and the Chebyshev map, for key generation which has larger security key space than single one. Moreover, we use these two sequences for further image encryption diffusion which decreases the correlation of neighboring pixels significantly. We conduct extensive experiments on several well-known images like Lena, Baboon, Koala, etc. Experimental results show that our algorithm has the characteristics of large key space, fast, robust to statistic attack, etc.

Keywords: Image encryption, chaotic map, key space, NPCR, UACI

1. Introduction

Smartphones and tablets have accounted for over 60 percent of all smart connected consumer devices, and the global smartphone installed base will grow from four billion in 2016 to more than six billion smartphones in use by 2020, according to new analysis released by IHS Markit. With the wide application of mobile devices, the security and personality of transfered images have attracted increasing attention recently.

When an image is transmitted over Internet, it may be easily accessed, illegally used or maliciously tampered [1]. To protect the user's image data, image encryption and data hiding in images have become two main aspects in both academia and industry. Data hiding techniques are mainly used to hide user information in images for transmitting while image encryption aims to mask the content of user images. For data hiding in images, some works [1-3] introduce reversible data hiding which is useful for recovering hidden information in the receiver. As in most data hiding researches, image encryption is the first step for masking the original image. We focus our research on image encryption in this paper.

Early methods of image encryption are mainly based on off-the-shelf data encrytion technologies [4, 5], such as DES (Data Encryption Standard), AES (Advanced Encryption Standard) [6], etc. However, those methods are demonstrated to be weak in efficiency and anti-attack ability due to the intrinsic features of images [5, 7, 8]. Recently, more attention is paid to chaos-based image encryption methods [9–13]. Compared to traditional non-chaotic image encryption, the chaos-based image encryption has larger key space, is faster, and is easier to implement.

In this paper, along the chaos-based pipeline, we propose an image encryption algorithm based on the cross diffusion of two chaotic maps, namely the Logistic map and the Chebyshev map. To increase the security key space, we first run the Logistic and the Chebyshev chaotic systems starting from two random initial keys with enough iterations, and then put the result of one of them into the other with several other iterations, finally we take the results as initial conditions of chaotic maps. Pixels are encrypted by XOR operations with chaotic sequences. To decrease the correlation of pixels, we switch the used two sequences to each other according to the parity of the pixel positions. For encryption diffusion, the current encrypted pixel (in the order of RGB) is conducted another XOR operation with the previous encrypted pixel (in the order of BGR). Compared to those methods based on single chaotic map, our method has larger security key space. And our encryption method can reduce the correlation of neighbouring pixels significantly due to the cross diffusion of two maps.

We summarize our contributions as following. First, we propose to use two chaotic sequences, namely the Logistic map and the Chebyshev map, for key generation which has larger security key space without adding large extra computation. Second, we use these two sequences for further image encryption diffusion which decreases the correlation of neighboring pixels significantly.

2. Related Work

According to the visualization of encrypted images, the image encryption algorithms can be classified into three major categories: (i) position permutation based algorithm [14](ii) value transformation based algorithm and [15, 16] (iii) visual transformation based algorithm [14].

In our work we mainly focus on the value transformation based algorithm. Along this line, there are two main categories: traditional image encryption methods and chaos-based methods.

The characteristics of low correlation, large key space, high key sensitivity, high entropy, and low time complexity are the main issues for a good image encryption algorithm. We refer to [15] for a detail survey on all these points.

Traditional image encryption methods are mainly based on standard data encryption algorithms [5, 7, 8]. Since the chaotic encryption algorithm proposed in [18], many works based on chaotic systems have been proposed [8, 19–30]. Pak and Huang proposed a simple and effective image encrytion method based on the difference of outputs from two identified 1D chaotic systems [26]. Compared to single choatic map, this method improves the chaotic performance and is more robust to any attacks. Rostami et al. proposed a parallel image encryption with chaotic windows based on logistic map [25], which has large key space and uniform histogram. This method first divides an image to image patches, and then performs XOR operation on patches with chaotic windows. The NPCR(number of pixels change rate) and UACI(unified average changing intensity) measurements show that it is effective to against differential attacks. Patidar et al. presented a modified substitution diffusion algorithm using chaotic standard and logistic maps [17]. This method is very fast in permutation and diffusion, and is effective to resist plaintext attacks. Praveenkumar et al. combined chaotic and Chebyshev keys and proposed a multi-level encryption system based on the frequency domain and the chaotic pixel permutation [21]. Yoon et al. first generated a random sequence using choatic systems, and then permuted the rows and columns of an image according to the random sequence [22]. All those methods need large computation cost in value transformantion or generating random sequences, and degrade on mobile devices in encryption speed.

3. Preliminaries on chaotic systems and chaos based image encrytion

Chaos-based algorithms have shown some remarkable properties in security, complexity, performance, speed etc. The following are the characteristics of chaotic maps:

(1)They are deterministic, i.e. they have several mathematical equations or formulation which ruling their behavior.

(2)They are sensitive to initial conditions.

(3)They are unpredictable and non-linear that is a small change can produce large or huge effects.

(4)They appear to be random and disorderly.

(5)They usually produce fractal patterns.

Fig. 1 shows the first part of our cross diffusion of Logistic and ChebyShev. The outputs, X_0^{cl} and X_0^{lc}, are the initial conditions of Logistic map and ChebyShev map for the three channels of an image (i.e. RGB). Note that the second part is embeded in the image encryption algorithm where the Logistic sequence and the Chebyshev sequence are used alternately.

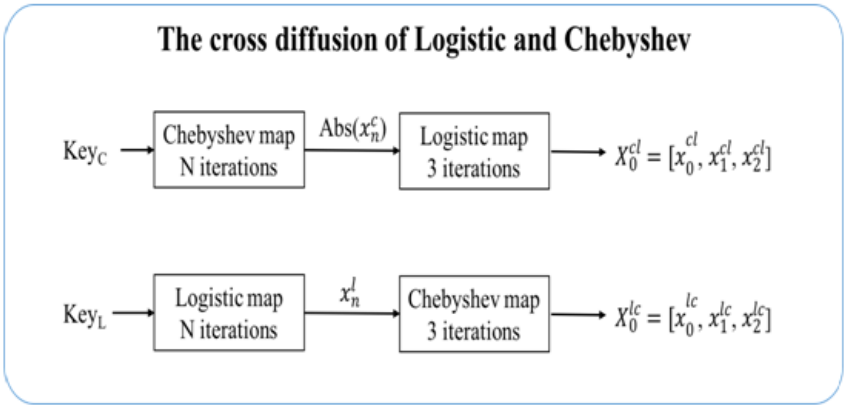


Fig. 1. The cross diffusion of Logistic and Chebyshev

In fact, chaotic systems and cryptographic algorithms have many similar properties [27], such as both are sensitive to initial parameters or keys, which makes chaos-based image encryption more and more popular.

There are two widely-used chaotic maps for encryption, namely the Logistic map and the Chebyshev map. The Logistic map is a classic non-linear chaotic function which has complex chaotic behavior. The formulation of Logistic map is as follows,

$$x_{n+1} = \mu x_n(1 - x_n), x_n \in (0, 1) \tag{1}$$

where x_n is the value after n iterations, and μ is the system parameter which makes the system chaotic and ensures $x_n \in (0, 1)$ when $\mu \in (3.5699456, 4]$.

The ChebyShev map is very sensitive to its initial conditions and can generate long-term unpredictable chaotic sequences. With increasing iterations, the generated chaotic trajectory becomes uniform and mixing, and the initial adjacent points will be seperated exponentially. The formulation of ChebyShev map is as follows,

$$x_{n+1} = cos(k\, arccos(x_n)), x_n \in [-1, 1] \tag{2}$$

where k is the control parameter which makes the function chaotic when k≥2.

Chaos-based image encrytion. The most popular pipeline of chaos-based image encryption mainly includes three steps: (i) generating a key as initial conditions for chaotic systems, (ii) generating a chaotic sequence, and (iii) conducting XOR operations on pixels with the chaotic sequence.

4. The proposed image encryption approach

We propose an image encrytion approach based on the cross diffusion of the Logistic map and the ChebyShev map. We present the cross diffusion scheme and the image encryption algorithm in the following two subsections.

4.1 The cross diffusion of Logistic and ChebyShev

The cross diffussion scheme is shown in two parts. Figure 1 illustrates the first part where the cross diffussion scheme is mainly used for generating initial conditions. Since we apply two chaotic maps for color image encryption, we need initial conditions for both maps of three channels (i.e. RGB).

Fig. 2 shows our image encryption algorithm.

Algorithm 1 Our image encryption algorithm

Require: $I \in R^{3\times M\times N}$: the original image; X_0^{cl}: the initial condition of Chebyshev map for RGB channels; X_0^{lc}: the initial condition of Logistic map for RGB channels;
Ensure: the encrypted image I^*
for each pixel $P_i \in R^3, i \in [0, M \times N)$ **do**
  if i is even **then**
    generate X_{i+1}^{cl} using Chebyshev function;
    $P_i^{'} = X_{i+1}^{cl} \oplus P_i$;
    if $i > 0$ **then**
      $P_i^* = P_i^{'} \oplus P_{i-1}^*$, where P_i^* is the ciphertext of P_i;
    else
      $P_i^* = P_i^{'}$;
    end if
  else
    generate X_{i+1}^{lc} using Logistic function;
    $P_i^{'} = X_{i+1}^{lc} \oplus P_i$;
    $P_i^* = P_i^{'} \oplus P_{i-1}^*$;
  end if
end for

Fig. 2. Our image encryption algorithm

For the Chebyshev map, starting from a $Key_c \in [1, 1]$, we first conduct N iterations (we fix N to 100 in our implementation) using the Chebyshev function, and then input the absolute value of x_n^c into the Logistic function, finally we generate three initial conditions by another 3 iterations. For the Logistic map, starting from a $Key_l \in (0, 1)$, we first perform N iterations using the Logistic function, and then input the result to the Chebyshev function, finally we also generate three initial conditions by another 3 iterations.

4.2 The image encryption algorithm

Our image encryption method is described in Algorithm 1. The inputs of our image encryption methods are the outputs from Figure 1 and an original image I.

Given a pixel $P_i \in I$ with position i, the main steps of our approach is as follows:

(1) Check the parity of i, go to (2) if it is even, otherwise go to (4).

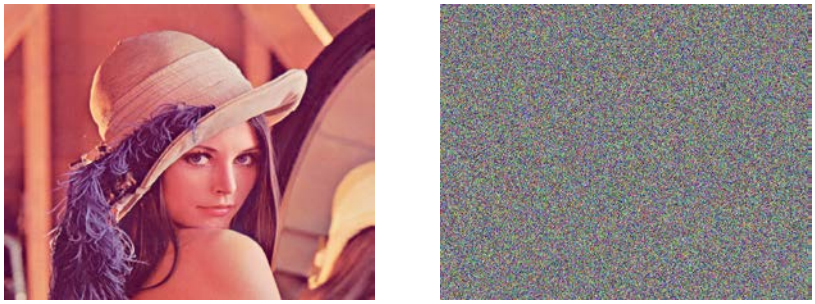
(2) Compute x_{i+1}^{cl} by the Chebyshev function (i.e. equation (2)), conduct XOR operation with x_{i+1}^{cl} and P_i for diffusion, and denote the result as $P_i^{'}$.

(3) Perform another XOR operation with $P_i^{'}$ and the previous encrypted pixel P_{i-1}^* if $i>0$, the result P_i^* is the target pixel. This *step* shows our cross-diffusion scheme implictly *since* P_{i-1}^* is obtained by the other chaotic map.

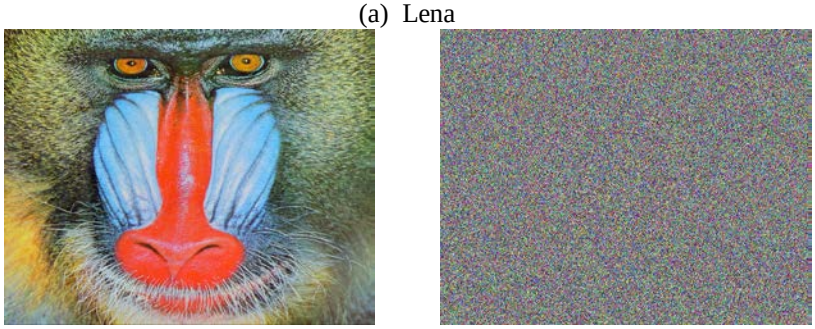
(4) Compute x_{i+1}^{lc} by the Logistic function (i.e. equation (1)), conduct XOR operation with x_{i+1}^{lc} and P_i for diffusion, and denote the result as $P_i^{'}$.

(5) Perform another XOR operation with $P_i^{'}$ and the previous encrypted pixel P_{i-1}^*, and get the final targe pixel P_i^*.

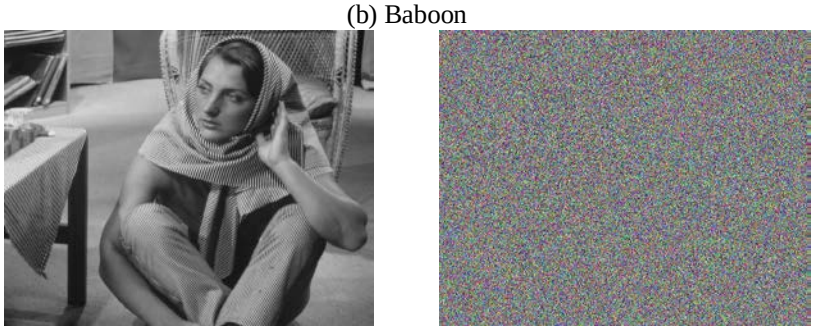
Fig. 3 shows several examples using our image encryption algorithm. The encrypted images are noise-like images as expected. We use these 8 images(a-h) for our experiments and present extensive analysis in the next section.



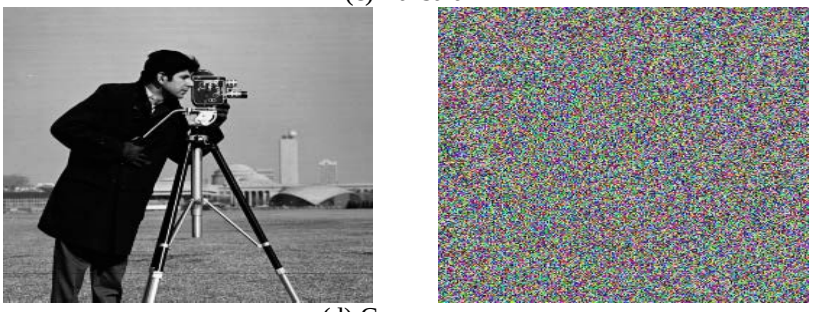
(a) Lena



(b) Baboon

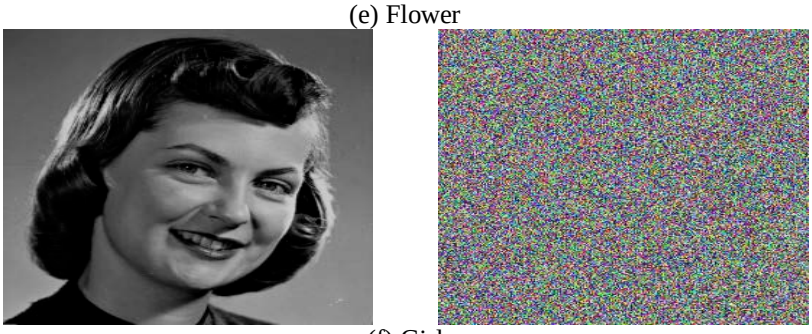


(c) Barbara

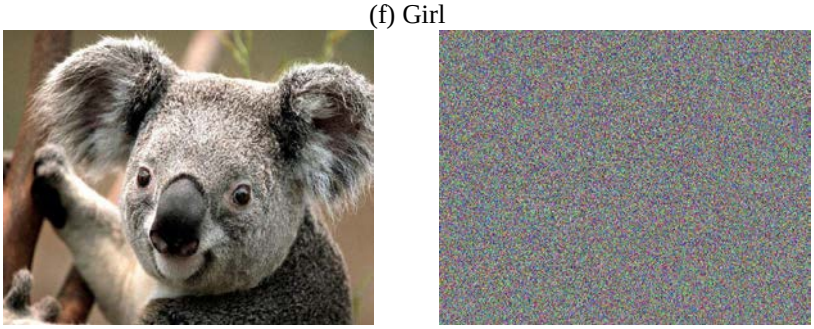


(d) Cameraman

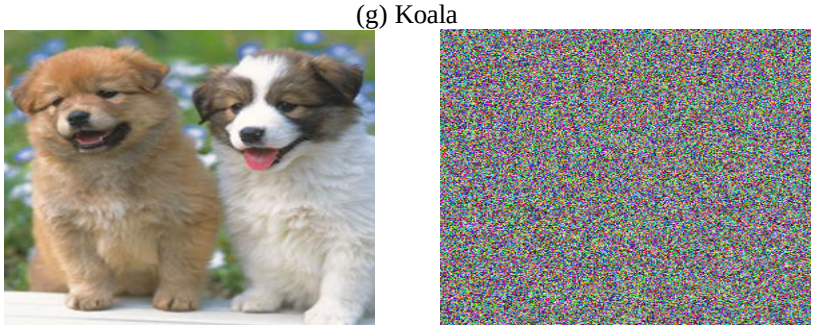
(e) Flower



(f) Girl



(g) Koala



(h) dog

Fig. 3. Original test images and the corresponding encryted images.

5. Experimental Analysis

To evaluate our image encryption algorithm, we conduct extensive experimental analysis on several standard images with varied sizes. In this section, we make time analysis, statistical analysis, key space analysis, information entropy analysis, and sensitivity analysis.

5.1 Time analysis

Regardless the security, the time delay is the most important user experience for mobile devices. For the time analysis, we conduct experiments on the Lena.bmp image with varied scales, and other standard images such as Girl.bmp, Baboon.bmp, and Barbara.bmp. Table 1 shows a comparison between our algorithm and other algorithm designed for mobile platform [23, 28]. We compute the average running time after 100 runs using Opencv2.4.13 (Android version). The running time is obtained on a smartphone with Android6.0 OS, 1.7 GHz Hisilicon Kirin920 CPU, and 3 GB RAM. From **Table 1**, we see that our algorithm is very efficient in both encryption and decryption.

Table 1. The time analysis of encryption and decryption
"-" indicates that they are not available in the paper

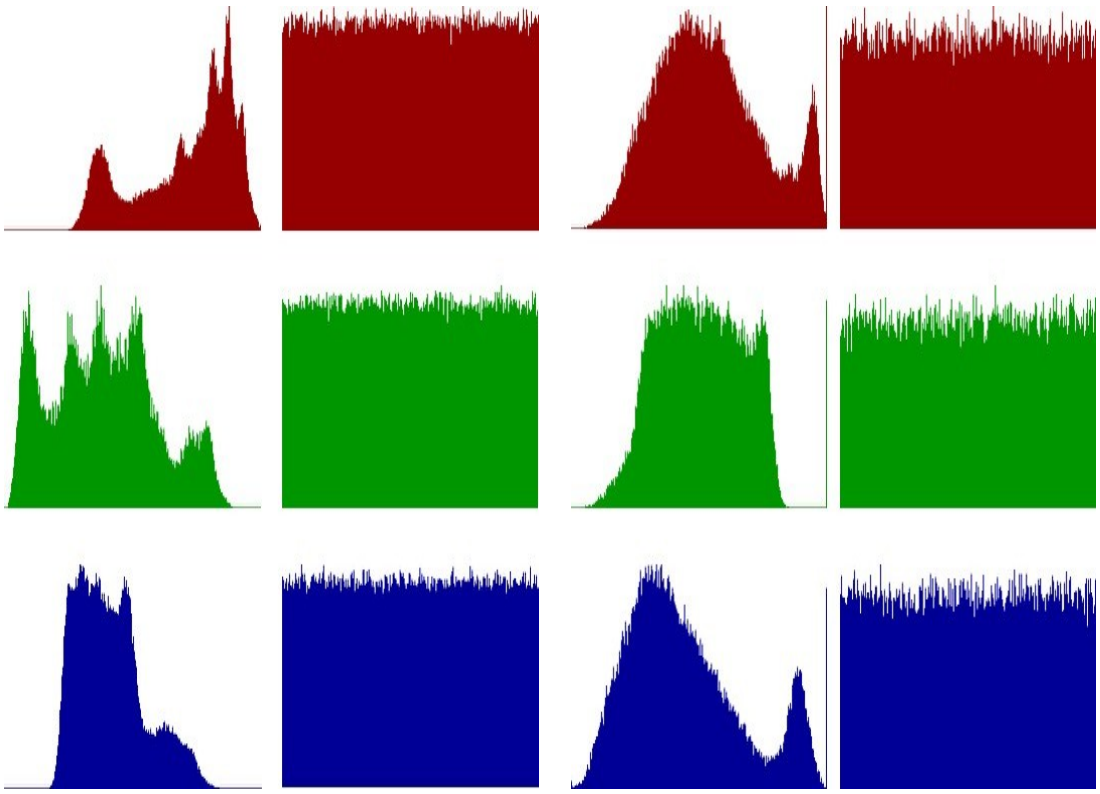
Image size	[23](s)		[28](s)		Ours(s)	
	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
Lena.bmp (128×128)	0.070	0.068	-	-	0.0039	0.0033
Lena.bmp 256×256)	0.261	0.269	-	-	0.0152	0.0130
Lena.bmp (512×512)	0.298	0.287	-	-	0.0621	0.0524
Lena.bmp (1366×768)	1.175	1.127	-	-	0.2460	0.2007
Girl.bmp (256×256)	0.198	0.201	0.047	0.050	0.0355	0.0085
Baboon.bmp (512×512)	0.256	0.249	0.181	0.180	0.0868	0.0347
Barbara.bmp (787×576)	0.446	0.449	0.306	0.312	0.0992	0.0658

5.2 Statistical analysis

We mainly analyze the histograms and the correlation of two adjacent pixels using the Baboon.bmp image in this section.

5.2.1 Histogram analysis

Image histogram describes the distribution of pixel values of an image. Flatter is better to resist statistic attacks for the image histogram. We conduct the histogram analysis on the Baboon.bmp image. Figure 4 shows the histograms of the original image and the encrypted image, The histogram comparison of RGB channels (from top to down) on Lena and Baboon. In each subfigures, the original image histograms are shown in the left, and the encrypted image histograms in the right. From Figure 4, we see that all the channels of the encrypted image have good uniform distributions, thus it is effective to resist statistic attacks.



(a) Lena (b) Baboon

Fig. 4. Comparison of histogram of original image and encrypted image

5.2.2 Correlation analysis of two adjacent pixels

Neighbouring pixels in an image are strong correlated in the directions of vertical, horizontal and diagonal. This correlation of encrypted image can be used by statistic attacks. To this end, an encrytion algorithm should be able to remove the correlation of adjacent pixels.

The correlation coefficient is calculated as follows:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{3}$$

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

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

where x and y are the color values of two neighbouring pixels in images, N is the number of selected pixel pairs, and r_{xy} is in [-1, 1]. It is strongly related if $|r_{xy}| > 0.8$, otherwise weakly related if $|r_{xy}| < 0.3$.

We conduct correlation analysis on three standard images, namely Cameraman.bmp, Lena.bmp, and flower.bmp. The correlation coefficients among adjacent pixels at horizontal, vertical and diagonal directions on these images are shown in Table 2. Those results are obtained by randomly selected 3000 pixel pairs in corresponding directions. From Table 2, we see that adjacent pixels are strong correlated in the original image and almost irrelated in the

encrypted image. The last two rows in **Table 2** compare our method with a recent chaos based method. Our method shows smaller correlation coefficients in average.

Table 2. The comparison of correlation coefficient
The results on Lena.bmp and flower.bmp are not available in [25]

Method	Image	Channel	Original image			Encrypted image		
			Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal
Ours	Cameraman.bmp(256×256)	Red	0.96148	0.93074	0.91664	0.01722	0.02364	0.01949
		Green	0.96148	0.93074	0.91664	0.03334	0.01521	0.03959
		Blue	0.96148	0.93074	0.91664	0.00992	0.02257	0.00792
		Average	-	-	-	**0.0201**6	**0.02047**	**0.02233**
[25]	Cameraman.bmp	Average	-	-	-	0.02608	0.02424	0.02446
Ours	Lena.bmp(256×256)	Red	0.96860	0.94530	0.92132	0.01425	0.02421	0.02745
		Green	0.97094	0.94633	0.92572	0.00557	0.00061	0.02194
		Blue	0.94140	0.90402	0.87477	0.01317	0.00166	0.00498
		Average	-	-	-	0.01099	0.00883	0.01812
Ours	Flower.bmp(1200×761)	Red	0.99452	0.99467	0.99042	0.03383	0.01811	0.01331
		Green	0.99454	0.99493	0.99106	0.01992	0.01440	0.00373
		Blue	0.98617	0.98836	0.97799	0.00774	0.00118	0.00737
		Average	-	-	-	0.02050	0.01123	0.00814

5.3 Key space analysis

The key space is a basic measurement of any encryption algorithms. In our algorithm we have 8 keys, namely Key_c, Key_l, x_0^{cl}, x_1^{cl}, x_2^{cl}, x_0^{lc}, x_1^{lc}, and x_2^{lc}. Key_c and Key_l are computed in the accuracy of 10^{-16}. x_0^{cl}, x_1^{cl}, x_2^{cl} are generated simultaneously, and the joint key space of them is 3×10^{16}. The key space of [x_0^{lc}, x_1^{lc}, x_2^{lc}] is 3×10^{16} as well. So the total key space is $10^{16}\times10^{16}\times(3\times10^{16})\times(3\times10^{16})\approx2^{213}$. It means our algorithm has large enough key space to withstand the brute force attack. And our key space is significantly larger than that (2^{138}) in a recent chaos based algorithm [26].

5.4 Information entropy analysis

Information entropy refers to disorder or uncertainty [29]. We compute the image information as follows,

$$H = -\sum_{i=0}^{255} p_i \log p_i \tag{4}$$

where p_i is the probability of gray value i in an image, which can be obtrained from the histogram. Table 3 shows the image informantion entropies of both the original image (Baboon.bmp) and the encryted image. A uniform distribution has the maximum value 8 in

Equation (4). Our encrypted image has larger entropy than the original one, which means the encrypted image is more random.

Table 3. The information entropy analysis

Image	Original image			Encrypted image		
	R	G	B	R	G	B
Baboon(512*512)	7.64129	7.34797	7.66837	7.99747	7.99748	7.99721
Lena(512*512)	7.25310	7.59404	6.96843	7.99929	7.99932	7.99922
Flower(1200*761)	7.43894	7.64537	7.12677	7.99976	7.99979	7.99978

5.5 Sensitivity analysis

We make qualitative and quantitative sensitivity analysis in our experiment. For qualitative analysis, we add a small noise (10^{-16}) to both Key_c (we use 0.456) and Key_l (we use 0.456), and show the decrypted images in **Fig. 5**. The very different decryted images indicate the good sensitivity of our algorithm.

Fig. 5. The decrypted Lena image with correct keys (left) and with a small noise added to Key_c and Key_l (right).

The sensitivity also can be quantitatively evaluated by NPCR(number of pixels change rate) and UACI(unified average changing intensity). We compute those two values from two encrypted images c_1 (with Key_c=Key_l=0.456) and c_2(with Key_c=Key_l=0.456+10^{-16}). NPCR is formulated as follows,

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \tag{5}$$

where $D(i, j) = 0$ if $c_1(i, j) == c_2(i, j)$, otherwise $D(i, j) = 1$. And the UACI is formulated as follows,

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|c_1(i,j) - c_2(i,j)|}{255} \times 100\%. \tag{6}$$

Table 4 shows NPCR and UACI measurements on four images. In general, larger values indict that the algorithm is more sensitive and better. We compare our algorithm to the latest chaos-based method [26] with similar initial keys and the same small noise(i.e. 10^{-16}).

From the comparison, our algorithm presents better performance in NPCR and UACI measurements.

Table 4. The NPCR and UACI in three channels. Larger is better

Image	Method	NPCR(%)			UACI(%)		
		R	G	B	R	G	B
Lena.bmp	[26]	99.6552	99.6277	99.588	33.4846	33.4132	33.3441
	ours	**99.6561**	**99.6309**	**99.6063**	**33.5069**	**33.4212**	**33.5187**
Koala.bmp	[26]	99.5834	99.6262	99.5926	33.5202	33.4907	33.4115
	ours	**99.6249**	**99.6365**	**99.6173**	**33.5293**	**33.4928**	**33.4234**
Flower.bmp	[26]	99.6277	99.6063	99.5575	33.3951	33.5272	33.4008
	ours	**99.6315**	**99.6137**	**99.6347**	**33.6273**	**33.5286**	**33.4341**
Dog.bmp	[26]	99.6353	99.6063	99.5926	33.4502	33.3814	33.5937
	ours	**99.6396**	**99.6127**	**99.6276**	**33.4502**	**33.4559**	**33.6018**

To evaluate the sensitivity, we also mimic differential attacks as in [25,31]. The image was primarily encrypted. Next, we changed one pixel in the plain image and encrypted it. Then the NPCR and UACI between the corresponding cipher images are calculated. This process was iterated 100 times by changing the random pixels. We conduct extensive experiments on the widely-used USC-SIPI image processing dataset. A comprehensive comparison of NPCR and UACI are shown in Table 5. As shown in the last row of Table 5, our method is superier or compariable to [25,31] in both mean NPCR and UACI.

Table 5. NPCR and UACI values of encrypted images for the Proposed Method and other algorithms running a hundred times on USC-SIPI image dataset

Image	Method	NPCR(%)			UACI(%)		
		Max	Min	Mean	Max	Min	Mean
5.1.09	[25]	99.68262	99.54376	99.61049	33.69240	33.20391	33.42682
	[31]	99.66736	99.53461	99.60907	33.73600	33.29381	33.48055
	ours	99.67874	99.54415	99.60912	33.67090	33.21256	33.44120
5.1.10	[25]	99.65515	99.55597	99.61079	33.71333	33.29039	33.47612
	[31]	99.69330	99.54681	99.61218	33.69135	33.18796	33.44003
	ours	99.67041	99.55798	99.61112	33.72295	33.21158	33.46348
5.1.11	[25]	99.65668	99.53461	99.60550	33.67225	33.25851	33.47719
	[31]	99.65820	99.53461	99.60997	33.71989	33.21694	33.46591
	ours	99.67041	99.53579	99.60962	33.69273	33.21263	33.47443
5.1.12	[25]	99.66125	99.55444	99.60843	33.62589	33.25152	33.45952
	[31]	99.65820	99.54681	99.60915	33.64215	33.19594	33.43716
	ours	99.67521	99.54734	99.60912	33.70610	33.22198	33.47070
5.1.13	[25]	99.67041	99.53918	99.60722	33.67809	33.20683	33.45420
	[31]	99.66888	99.55750	99.61072	33.66077	33.28392	33.47259
	ours	99.67041	99.54028	99.60912	33.76906	33.20987	33.48710
5.2.08	[25]	99.64409	99.58458	99.61077	-	-	-
	[31]	99.63646	99.58496	99.60844	-	-	-
	ours	99.64218	99.57390	99.60872	33.60928	33.38309	33.48207
5.2.09	[25]	99.63455	99.57886	99.61013	-	-	-
	[31]	99.63951	99.57695	99.60905	-	-	-
	ours	99.64218	99.57390	99.60872	33.60068	33.34674	33.46001

5.2.10	[25]	99.64943	99.58649	99.60957	-	-	-
	[31]	99.63837	99.57504	99.60878	-	-	-
	ours	99.64218	99.57390	99.60872	33.57785	33.34855	33.46840
5.3.01	[25]	99.62454	99.59812	99.60938	-	-	-
	[31]	99.62435	99.59164	99.60940	-	-	-
	ours	99.62330	99.59421	99.60950	33.51622	33.39734	33.45913
5.3.02	[25]	99.63417	99.59249	99.60966	-	-	-
	[31]	99.62502	99.59478	99.60907	-	-	-
	ours	99.62330	99.59421	99.60950	33.52143	33.39220	33.46370
7.1.01	[25]	99.64523	99.57809	99.60949	-	-	-
	[31]	99.63531	99.58115	99.60944	-	-	-
	ours	99.64218	99.57390	99.60872	33.59882	33.35273	33.47480
boat.512	[25]	99.63646	99.58305	99.60901	33.53040	33.36722	33.44522
	[31]	99.64027	99.56818	99.60839	33.59018	33.34697	33.47754
	ours	99.64218	99.57390	99.60872	33.62001	33.35694	33.46451
elaine	[25]	99.63150	99.57771	99.61025	33.57845	33.34303	33.44522
	[31]	99.64180	99.57008	99.60872	33.58067	33.34871	33.47754
	Ours	99.67041	99.57087	99.60912	33.72630	33.33219	33.45041
gray21.512	[25]	99.63951	99.57504	99.60947	33.59614	33.36760	33.48123
	[31]	99.63875	99.58191	99.61034	33.57239	33.33608	33.47169
	ours	99.64218	99.57465	99.61012	33.58678	33.34796	33.46776
ruler.512	[25]	99.64600	99.58458	99.61077	33.53510	33.37913	33.47222
	[31]	99.63303	99.58305	99.60969	33.50485	33.24784	33.42034
	ours	99.64218	99.57390	99.60872	33.64218	33.34359	33.47028
testpat.1k	[25]	99.62378	99.59635	99.60941	-	-	-
	[31]	99.62206	99.59440	99.60900	-	-	-
	ours	99.62329	99.59421	99.60949	33.53071	33.41411	33.46971
Cameraman	[25]	99.66583	99.55750	99.61101	33.71245	33.29961	33.50756
	[31]	99.68109	99.55292	99.60669	33.72510	33.20944	33.46393
	ours	99.66857	99.54087	99.61075	33.76228	33.22437	33.48043
total results	[25]	**99.68262**	**99.53461**	**99.60917**	**33.74719**	**33.20391**	**33.46566**
	[31]	**99.69330**	**99.53461**	**99.60966**	**33.73600**	**33.18796**	**33.45982**
	ours	**99.67874**	**99.53579**	***99.60935***	**33.76906**	**33.20987**	***33.46754***

5. Conclusion

This paper proposed a new image encryption algorithm based on the cross diffusion of two chaotic maps. The proposed algorithm has four main properties. First, it has very large security key space due to the cross diffusion of Logistic map and ChebyShev map in key generation. Second, it makes the correlation of adjacent pixels very weak due to the cross diffusion of Logistic map and ChebyShev map in encryption process. Third, it is very sensitive to tiny differences of the initial condition due to the XOR operation between previous and current pixels. Forth, it is realtime on mobile devices.

Acknowledgment

This work is partly supported by the Hunan Provincial Natural Science Foundation of China (Grant No. 2017JJ2010), the Scientific Research Fund of Hunan Provincial Education Department (Grant No. 16B039), the Natural Science Foundation of China (Grant No. 61502152), the Science and Technology Plan Project of Hunan Province (Grant No.

2016TP1020), Open fund project of Hunan Provincial Key Laboratory of Intelligent Information Processing and Application for Hengyang normal university (Grant No. IIPA18K03).

References

[1] Chuan Qin, Zhihong He, Xiangyang Luo, Jing Dong, "Reversible data hiding in encrypted image with separable capability and high embedding capacity," *Information Sciences*, 465: 285-304, 2018. Article (CrossRef Link)
[2] Xinpeng Zhang, "Separable Reversible Data Hiding in Encrypted Image," *IEEE Transactions On Information Forensics And Security*, Vol. 7, No. 2, 2014. Article (CrossRef Link)
[3] Chuan Qin, Xinpeng Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *J. Vis. Commun. Image R.* 31, 154–164, 2015. Article (CrossRef Link)
[4] J. Li, H. Liu, "Colour image encryption based on advanced encryption standard algorithm with two-dimensional chaotic map," *IET information security*, 7(4), pp. 265–270, 2013. Article (CrossRef Link)
[5] Q. Zhang, Q. Ding, "Digital image encryption based on advanced encryption standard (aes)," in *Proc. of Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2015 Fifth International Conference on*, pp. 1218–1221, IEEE, 2015. Article (CrossRef Link)
[6] J. Daemen, V. Rijmen. "Aes proposal: Rijndael," 1999. Article (CrossRef Link)
[7] G. Chen, Y. Mao, C. K. Chui. "A symmetric image encryption scheme based on 3d chaotic cat maps," *Chaos, Solitons & Fractals*, 21(3), pp. 749–761, 2004. Article (CrossRef Link)
[8] N. B. Slimane, K. Bouallegue, M. Machhout, "Nested chaotic image encryption scheme using two-diffusion process and the secure hash algorithm sha-1," in *Proc. of Control Engineering & Information Technology (CEIT), 2016 4th International Conference on*, pp. 1–5, IEEE, 2016. Article (CrossRef Link)
[9] Y. Abanda, A. Tiedeu, "Image encryption by chaos mixing," *IET Image Processing*, 10(10), pp. 742–750, 2016. Article (CrossRef Link)
[10] L. Liu, S. Miao, H. Hu, M. Cheng, "N-phase logistic chaotic sequence and its application for image encryption," *IET Signal Processing*, 10(9), pp. 1096–1104, 2016. Article (CrossRef Link)
[11] H. Huang, S. Yang, "Colour image encryption based on logistic mapping and double random-phase encoding," *IET Image Processing*, 11(4), pp. 211–216, 2016. Article (CrossRef Link)
[12] H. Liu, A. Kadir, X. Sun, "Chaos-based fast colour image encryption scheme with true random number keys from environmental noise," *IET Image Processing*, 11(5), pp. 324–332, 2017. Article (CrossRef Link)
[13] M. Boussif, N. Aloui, A. Cherif, "Smartphone application for medical images secured exchange based on encryption using the matrix product and the exclusive addition," *IET Image Processing*, 11(11), pp. 1020–1026, 2017. Article (CrossRef Link)
[14] J. I. Guo, J. C. Yen. "A new chaotic mirror-like image encryption algorithm and its VLSI architecture," *Pattern Recognition and Image Analysis*, 10(2), pp. 236–247, 2000. Article (CrossRef Link)
[15] Manju Kumari, Shailender Gupta, Pranshul Sardana. "A Survey of Image Encryption Algorithms". *3D Research*, 2017(8): 1-35. Article (CrossRef Link)
[16] A. Sinha, K. Singh, "A technique for image encryption using digital signature," *Optics Communications*, 218(2203), pp. 229–234, 2003. Article (CrossRef Link)
[17] S. S. Maniccam, N. Bourbakis, "Lossless image compression and encryption using scan," *Pattern Recognition*, 34, pp. 1229–1245, 2001. Article (CrossRef Link)
[18] R. Matthews, "On the derivation of a "chaotic" encryption algorithm", *Cryptologia*, 13(1), pp. 29–42, 1989. Article (CrossRef Link)

[19] H. Gao, Y. Zhang, S. Liang, D. Li, "A new chaotic algorithm for image encryption," *Chaos, Solitons & Fractals*, 29(2), pp. 393–399, 2006. Article (CrossRef Link)
[20] N. K. Pareek, V. Patidar, K. K. Sud, "Image encryption using chaotic logistic map," *Image and vision computing*, 24(9), pp. 926–934, 2006. Article (CrossRef Link)
[21] V. Patidar, N. Pareek, G. Purohit, K. Sud, "Modified substitution–diffusion image cipher using chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, 15(10), pp. 2755–2765, 2010. Article (CrossRef Link)
[22] J. W. Yoon, H. Kim, "An image encryption scheme with a pseudorandom permutation based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, 15(12), pp. 3998–4006, 2010. Article (CrossRef Link)
[23] W. Wei, J. Cong, "Image encryption scheme for android mobile platform," *Computer Science*, 1(8), pp. 94–96, 2014. Article (CrossRef Link)
[24] P. Praveenkumar, R. Nisha, K. Thenmozhi, J. B. B. Rayappan, R. Amirtharajan, "Image merger encryptor: A chaotic and chebyshev key approach," *Research Journal of Information Technology*, 2016. Article (CrossRef Link)
[25] M. J. Rostami, A. Shahba, S. Saryazdi, H. Nezamabadi-pour, "A novel parallel image encryption with chaotic windows based on logistic map," *Computers & Electrical Engineering*, 2017. Article (CrossRef Link)
[26] C. Pak, L. Huang, "A new color image encryption using combination of the 1d chaotic map," *Signal Processing*, 138, pp. 129–137, 2017. Article (CrossRef Link)
[27] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits &Systems Magazine*, 1(3), pp. 6–21, 2001. Article (CrossRef Link)
[28] Nanrun Zhou, Haolin Li, Di Wang, Shumin Pan, Zhihong Zhou, "Image compression and encryption scheme based on 2D compressive sensing and fractional Mellin transform," *Optics Communications*, 343, 10-21, 2015. Article (CrossRef Link)
[29] Nanrun Zhou, Shumin Pan, Shan Cheng, Zhihong Zhou, "Image compression-encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Optics and Laser Technology*, 82: 121-133, 2016. Article (CrossRef Link)
[30] L. H. Gong, X. B. Liu, F. Zheng, N. R. Zhou, "Flexible multiple-image encryption algorithm based on log-polar transform and double random phase encoding technique," *Journal of Modern Optics*, 60(13): 1074-1082, 2013. Article (CrossRef Link)
[31] Y. Wu, Y. Zhou, J.P. Noonan, S.Agaian, "Design of image cipher using Latin squares," *Information Sciences*, 264: 317–339, 2014. Article (CrossRef Link)

Ge Jiao received his MA.Eng from Hunan University, Changsha, China, in 2010. He is currently working toward the PhD degree in safety engineering at University of South China. He works at Hengyang Normal University as a associate professor. His research interest focuses on information security, Image encryption and encryption chip side channel attacks and protection.

Xiaojiang Peng received his master degree in school of Information Science and Technology from Southwest Jiaotong University in 2014. He currently is an Associate Professor at the Multimedia Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. He was a postdoctoral researcher at Idiap Institute, Switzerland from 2016 to 2017, and was a postdoctoral researcher in LEAR Team, INRIA, France, working with Prof. Cordelia Schmid from 03/2015 to 07/2016. He serves as a reviewer for IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, Image and Vision Computing, Machine Vision and Applications, IEEE Signal Processing Letter, Multimedia Tools and Applications, Neurocomputing, IET Computer Vision, FG, etc. His research focus is in the area of image processing, action recognition and detection, face recognition and deep learning.

Kaiwen Duan is currently pursuing a bachelor degree at the College of Information Engineering in Hengyang Normal University, Hengyang, China. His current research interests include the image processing and information security.