A COLOR IMAGE ENCRYPTION SCHEME WITH SYNCHRONOUS PERMUTATION-DIFFUSION STRUCTURE

Xiangguang Dai
Yuming Feng*
Key Laboratory of Intelligent Information Processing and Control
Chongqing Three Gorges University
Wanzhou, Chongqing
404100, China
and
Chongqing Engineering Research Center
of Internet of Things and Intelligent Control Technology
Chongqing Three Gorges University
Wanzhou, Chongqing, 404100
China
yumingfeng25928@163.com

Abstract. Permutation and diffusion are two basic principles in designing an image encryption algorithm. Almost all image encryption methods are based on a scheme that separates permutation and diffusion, namely, asynchronous permutation and diffusion scheme (APDS). This paper analyses the flaws of APDS and cracks it with a chosen plaintext attack, and then proposes a synchronous permutation-diffusion scheme (SPDS). Experimental simulations and performance evaluations in key space, key sensibility, correlation coefficient, Shannon entropy, differential attack and data loss/noise attacks all show that the proposed scheme processes better performance compared with the APDS and some others, and can ensure a secure communication in practical applications.

Keywords: Image encryption, Synchronous permutation-diffusion, Chaotic map.

1. Introduction

With the increasing degree of interconnection, openness, and sharing of computer networks, the Internet has rapidly developed and is used for an extensive number of applications. Data, images, and multimedia information have become the largest online information flow. It is precisely because of the popularization of the Internet, as well as the simplicity, visualization, and information richness of image information, image information has become the most common information transmitted on the Internet. However, images often include important personal information, business secrets, or even information containing national

∗. Corresponding author

secrets. Therefore, it is essential to encrypt the images securely for transmitting. However, the encryption technology used in image transmission is now facing the danger of being cracked with the increase of computing power. Therefore, new encryption algorithms are required for the security of image transmission.

Two basic methods used in image encryption are permutation and diffusion [1]. In permutation, each coordinate in the image is changed, and in diffusion, pixel values of the image are modified. A good image encryption scheme should present good permutation and diffusion effects which make the original image into a noisy one and can also resist all kinds of attacks. A permutation-only image encryption scheme based on chaos has been proposed in [2] and a diffusion-only scheme has been proposed in [3]. In [7, 5, 8, 9, 10, 11, 12, 13, 6, 15, 14], encryption schemes based on permutation and diffusion structures are proposed, where [7, 5, 8, 9, 10, 11, 12, 13, 6, 14] encrypts images using permutation first and is followed by diffusion, or diffusion is first, followed by permutation. Image encryption with permutation and diffusion performed separately has a defect that cryptanalysts can exploit to obtain one of the key streams of permutation and diffusion first, and can then obtain another by certain chosen plaintext attacks [5, 4]. Authors in [15] try to encrypt images using a synchronous permutation-diffusion scheme, but throughout their entire algorithm, the scheme is no different from permutation and diffusion done separately.

Chaos is a deterministic, random process in a nonlinear dynamic system that is neither periodic nor convergent, and has a very sensitive dependence on initial values. Given a discrete chaotic system and iterating it with two very close initial values, the output results are completely uncorrelated. Therefore, by using the extremely sensitive dependence of the chaotic system on the initial conditions, we can obtain a large number of uncorrelated, random, and deterministic chaotic sequences which have been widely used in image encryptions [7, 8, 9, 10, 11, 12, 13, 6, 15, 14]. Chaotic image encryption schemes [3, 6, 4] are usually based on a low-dimensional chaotic map which has the problems of a short code period and low accuracy, and therefore cannot guarantee the security of the scheme [14]. Subsequently, high-dimensional chaotic, or spatiotemporal chaotic encryption schemes, have been proposed to overcome the problems [7, 8, 9, 10, 11, 12, 13, 15, 14].

In this paper, we find the defects of APDS and propose a chosen plaintext attack to crack the APDS scheme and further propose a synchronous permutation-diffusion scheme to remedy the defects of APDS. The proposed scheme first determines the permutation sequence using a two-dimensional chaotic map, and then permutates the position of pixels one by one, and diffuses image pixel values related to the current permutation sequence, their former encrypted pixel values and a chaotic key stream; therefore, the diffusion is closely dependent on permutation and an attacker cannot crack the scheme by respectively extracting key-streams of permutation and diffusion using a chosen plaintext attack.

The rest of the paper is organized as follows: Section 2 gives a brief review of an asynchronous permutation and diffusion scheme proposed in [6] and analyses

its defects to further attack the scheme successfully. After that, a new image encryption scheme of synchronous permutation-diffusion is proposed in Section 3. Experimental performances of the proposed scheme are presented in Section 4, followed by the conclusions of the paper in Section 5.

2. APDS and its defects

2.1 Overview the APDS scheme

Image encryption using an APDS separately processes plain images with permutation and diffusion. The simple image encryption steps of this structure are listed as follows [6].

1. Reshape the plain image $I_{M\times N\times 3}$ into 1D array $P = \{p_1, p_2, \cdots, p_{3MN}\}$.
2. Iterate a 1D chaotic map $(3MN + N_0)$ times and throw out the former N_0 elements to obtain a sequence X with the length of $3MN$.
3. Sort X in ascending order to obtain an index sequence PX.
4. Let $P' = P(PX)$ be the permutated image matrix.
5. Let $D = mod(floor(X \times 10^{14}), 256)$ be the diffusion matrix.
6. Let $C(i) = mod(P'(i) + D(i), 256) \oplus C(i-1), i = 1, 2, \cdots, 3MN$ be the encrypted image matrix.
7. Circularly shift the elements in C towards left by the amount of lp: $C' = circshift(C, lp)$.
8. Reshape C into I_c with size of $M \times N \times 3$, then the final encryption image is I_c.

2.2 Defects of the APDS

Some defects of the above encryption algorithm are presented in detail below.

1. Vulnerable to CPA: using CPA as a chosen plaintext attack, adversaries can access the encryption machinery and choose arbitrary plaintexts from its corresponding ciphertexts. The adversary aims to obtain some useful information which helps divulge the other plaintexts' encrypted information with the same encryption scheme and the same secret keys.

   We choose two $M \times N \times 3$ plain images $I_1 = 0$ and $I_2 = 0$, and set one of the elements in I_2 to 1. Due to the encryption machinery available, the corresponding cipher image of I_1 and I_2 are denoted as I_{c1} and I_{c2}. Fig. 1 shows each step of the encryption process for two simple plain images. Next, we detail the steps of cracking APDS.

(a) Analyse and obtain key lp: As can be seen in Fig. 1, P_1 and P_2 are both 1D arrays which correspond to I_1 and I_2 respectively. Due to P_1 being a full 0 array, the permutation step is inoperative, thus the encryption algorithm is analogous to $C_1(i) = D(i) \oplus C_1(i-1)$, and $C_1' = circshift(C_1, lp)$. P_2 only has one element that equals 1; all other values equal 0. After permutated, the element 1 is moved to position s_1. The former s_1 elements of C_1 and C_2 are the same. The last lp elements of C_1' and C_2' are the former lp elements of C_1 and C_2 respectively. Thus, if $s_1 >= lp$, then lp is the number of identical elements between the tail of C_1' and C_2'; otherwise, lp can not be determined, and we should reset the index of element 1 in P_2 to meet the condition.

(b) Determine key stream D: After obtaining lp, we circularly shift the elements in C_1' towards the right lp times to get C_1, then $D(i) = C_1(i) \oplus C_1(i-1)$.

(c) Obtain the index array PX: We reselect a plain image and denote its 1D array as $P_{i1}, P_{i2}, \cdots, P_{in} (n = 3MN/256)$. The first 256 pixel values of P_{i1}, the 257th–512th pixel values of P_{i2}, $\cdots$, the $256(n-1)$th–$256n$th pixel values of P_{in} are all in the array $[0, 1, \cdots, 255]$ and the rest of the pixel values are all 0. We encrypt the n plain images to obtain their corresponding cipher images $I_{c1}, I_{c2}, \cdots, I_{cn}$ and then compare the n cipher images with I_1 respectively to extract all of the elements in PX.

The simple process of the CPA cryptanalysis algorithm is clearly shown in Fig. 2. A total of $\dfrac{3MN}{256} + 2$ images are needed to completely break the APDS.

2. Key stream generated by a low-dimensional chaotic map: A low-dimensional chaotic sequence cannot ensure the security of encryption schemes due to its shorter period and lower accuracy compared with high-dimensional maps and therefore cannot ensure the security of encryption schemes.

3. Synchronous permutation-diffusion scheme

Similar to the defects of APDS listed above, this section proposes a synchronous permutation-diffusion encryption scheme. In this scheme, 2D Logistic-adjusted-Sine map as Eq. (1) is used to generate the chaotic key streams [16].

$$\begin{cases} x(i+1) = \sin(\pi\mu(y_i + 3)x_i(1 - x_i)), \\ y(i+1) = \sin(\pi\mu(x_{i+1} + 3)y_i(1 - y_i)), \end{cases} \tag{1}$$

where $\mu \in [0, 1]$ is map parameter. When $\mu \in [0.37, 0.38] \cup [0.4, 0.42] \cup [0.44, 0.93] \cup \{1\}$, the LASM shows chaotic behavior. We iterate the map $3MN + N_0$ times

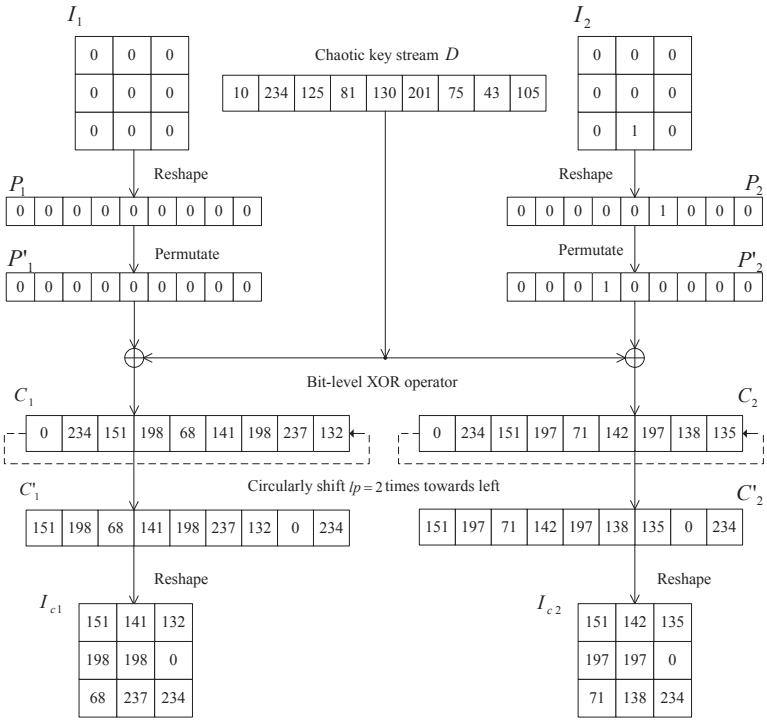


Figure 1: A simple demonstration of APDS image encryption

with proper initial values and parameters, and then discard the former N_0 data to obtain two chaotic sequences denoted as x, y. Then we obtain a permutation position matrix PX according to x and obtain a diffusion matrix D according to y. The detailed processes of the proposed encryption scheme are presented in the following subsection.

3.1 Encryption algorithm

1. Reshape the plain image $I_{M\times N\times 3}$ into 1D array $P = \{p_1, p_2, \cdots, p_{3MN}\}$.

2. Choose the secret keys μ, x_0, y_0, N_0 of LASM and iterate the chaotic map $(3MN + N_0)$ times and throw out the former N_0 elements to generate two chaotic sequences x, y with length $3MN$.

3. Sort x in ascending order to obtain an index sequence PX.

4. Permutate ith image pixel position with $P'(i) = P(PX(i))$.

5. Make $D(i) = mod(floor(y(i) \times 10^{14}), 256)$ the diffusion value.

6. Rotate $lp = mod(PX(i), 8)$ times.

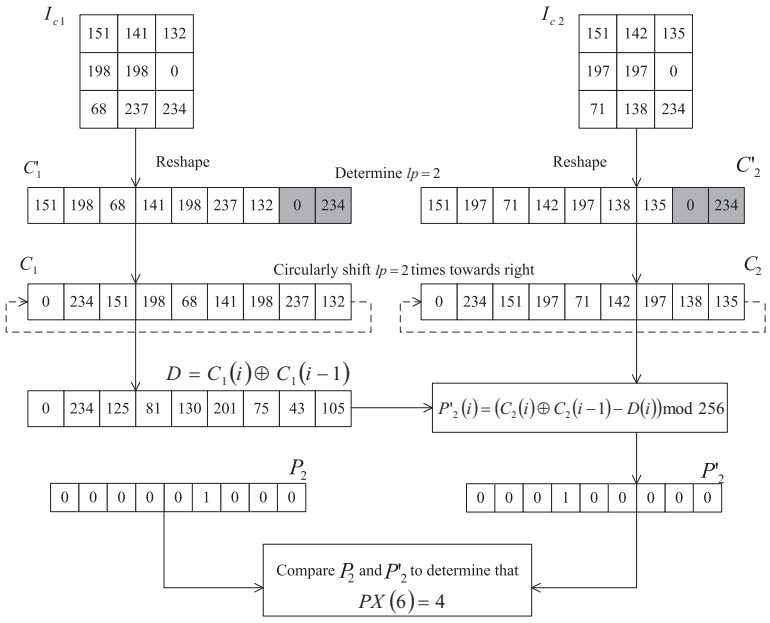


Figure 2: The process of cryptanalysis

7. Circularly shift the binary number that corresponds to the decimal number $C(i-1)$, towards the left by the amount of lp: $C'(i-1) = circshift(C(i-1), lp)$.

8. Encrypt the ith image pixel value by $C(i) = mod(P'(i) + D(i), 256) \oplus C(i-1)$.

9. Repeat steps 4 to 8 $3MN$ times to obtain all encrypted values in C.

10. Reshape C into I_c with a size of $M \times N \times 3$.

A flowchart of the encryption process is shown in the following Fig. 3. The decryption algorithm is an inverse process of the encryption algorithm.

The proposed algorithm encrypts plain images with permutation and diffusion associated encryption methods, thus, a cryptanalyst can break a cipher image only when he knows both PX and D. Therefore, the proposed scheme can effectively resist CPA.

4. Experimental results and discussion

The experimental simulations were run on desktop computer with an Intel(R) Core(TM) i5-3470 CPU 3.20GHz, 4GB RAM, and a 500GB hard drive. The operating system was Microsoft Windows 7 and the software run was Matlab

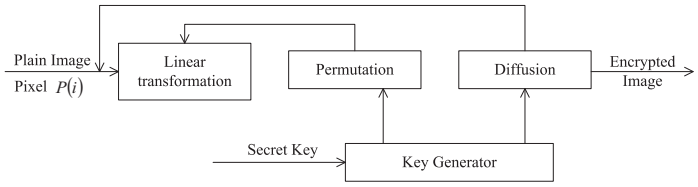
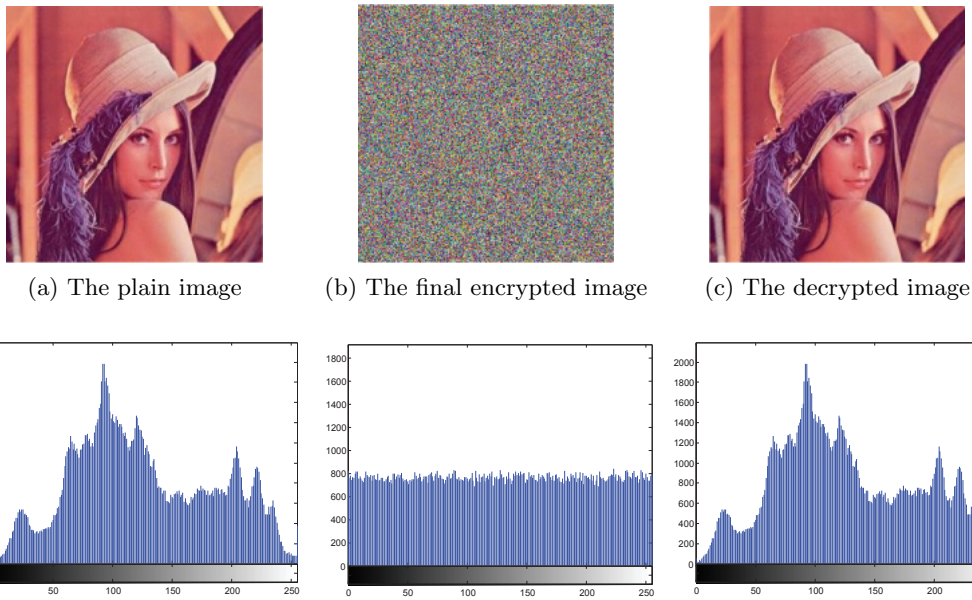


Figure 3: SPDS image encryption scheme

8.3.0.532 (R2014a). In order to evaluate the performance of the proposed encryption, the standard color image Lena.bmp is $256 \times 256 \times 3$ pixels, as shown in Fig. 4 (a), it was chosen as the plain image. The encrypted image and decrypted image of Lena are shown in Fig. 4 (b) and (c) respectively. Fig. 4 (d-f) are the histograms of the plain, ciphered, and deciphered image corresponding to Fig. 4 (a-c) respectively. Fig. 4 (e) shows an uniform distribution histogram, therefore, the cipher image encrypted by the proposed scheme cannot provide any useful information about the plain image.



(a) The plain image (b) The final encrypted image (c) The decrypted image

(d) Histogram of plain image (e) Histogram of encrypted image (f) Histogram of decrypted image

Figure 4: Plain image, encrypted image, decrypted image and their histogram.

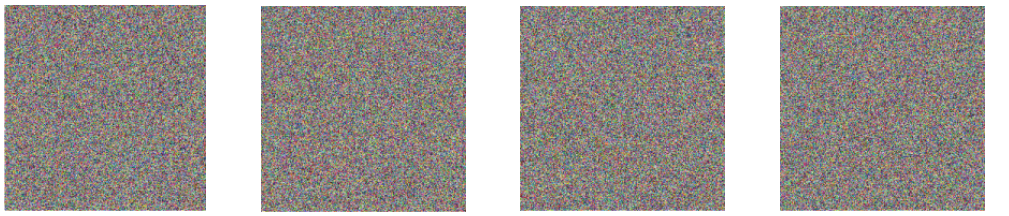
4.1 Key space

A larger space than 2^{100} is usually needed for an encryption scheme to resist a brute-force attack. Our proposed algorithm has four security keys: x_0, y_0, μ, N_0,

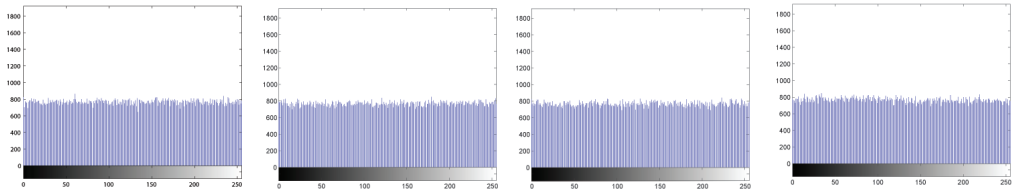
where $x_0, y_0 \in (0, 1]$ with accuracy 10^{16}, $\mu \in [0.37, 0.38] \cup [0.4, 0.42] \cup [0.44, 0.93] \cup \{1\}$ with accuracy 10^{16} and N_0 is a positive integer. Let $100 < N_0 < 10100$, then the space of N_0 is 10^4. The total key space can be calculated as $(10^{16})^3 \times 10^4 = 10^{52} \approx 2^{172}$, therefore, the proposed scheme can effectively resist a brute-force attack.

4.2 Key sensitivity

We encrypted the plain Lena image with secret keys $N_0 = 1000, x_0 = 0.3, y_0 = 0.4, \mu = 0.5$ to a cipher image, then we respectively changed one of the following four keys $N_0 = 1001, x_0 = 0.30001, y_0 = 0.40001, \mu = 0.50001$ and maintained the other three keys, then decrypted the ciphered Lena image with these changed keys. Fig. 5 (a-d) show the decrypted image with the wrong keys and their corresponding histogram are shown in Fig. 5 (e-h) respectively. From these figures, we know that a tiny change in any secret key can make the decrypted image noisy; because the decrypted image is noisy, we know the proposed scheme is extremely sensitive to a secret key.



(a) Decryption with wrong key N_0 (b) Decryption with wrong key x_0 (c) Decryption with wrong key y_0 (d) Decryption with wrong key μ



(e) Histogram of (a) (f) Histogram of (b) (g) Histogram of (c) (h) Histogram of (d)

Figure 5: Key sensitivity analysis

4.3 Correlation analysis

Due to the possibility that some images have strong correlation among adjacent pixels, cryptanalysts may be able to access some useful information. In our experiment, 2000 adjacent pixels of plain and cipher images were randomly selected from horizontal, vertical, and diagonal directions respectively, and their

correlation can be calculated by Eq. (2).

$$r_{xy} = \frac{N^2 \cdot \mathrm{cov}(x, y)}{\sum_{i=1}^{N}(x_i - E_x)^2 \cdot \sum_{i=1}^{N}(y_i - E_y)^2}, \tag{2}$$

where $E_x = \frac{\sum_{i=1}^{N} x_i}{N}$, $\mathrm{cov}(x, y) = E((x - E_x)(y - E_y))$; x, y are the two neighboring pixels' sequences. Table 1 shows experimental data and compares the correlation of the proposed scheme with some other references. From the table we know that the plain image has high correlation coefficients, as close to 1, in all three channels, but the cipher image encrypted by the proposed scheme has correlation coefficients close to 0, which represents that the encrypted image is a random-like image. Thus, the proposed scheme dramatically randomizes adjacent image pixel values.

Table 1: Correlation of two adjacent pixels in the plain and encrypted Lena image with different encryption scheme.

Lena	Orientation	R component	G component	B Component
Plain-image	Horizontal	0.9336	0.9501	0.8989
	Vertical	0.9746	0.9693	0.9361
	Diagonal	0.9299	0.9288	0.8485
Ref. [3]	Horizontal	-0.0463	0.0435	0.0136
	Vertical	-0.0587	-0.0682	-0.0688
	Diagonal	-0.0200	-0.0052	0.0127
Ref. [4]	Horizontal	0.0005	0.0011	-0.0023
	Vertical	-0.0070	0.0001	0.0078
	Diagonal	0.0005	-0.0016	-0.0009
Ref. [6]	Horizontal	0.0038	0.0069	0.236
	Vertical	0.0026	0.0125	0.0054
	Diagonal	0.0017	0.0037	0.0296
Ref. [17]	Horizontal	0.0104	0.0095	-0.0215
	Vertical	-0.0029	0.0126	0.0135
	Diagonal	0.0123	-0.0116	-0.0304
Ref. [18]	Horizontal	0.0049	0.0054	0.0053
	Vertical	0.0031	0.0001	0.0022
	Diagonal	0.0007	0.0017	0.0007
Proposed	Horizontal	0.0001	0.0004	0.0023
	Vertical	0.0081	0.0101	0.0003
	Diagonal	0.0004	0.0016	0.0002

4.4 Shannon entropy

Shannon entropy [4] is defined to measure the randomness of the test image. The greater entropy corresponds to the more uniform image gray value distribution.

Shannon entropy of an 8-bit image is defined as follows:

$$H(m) = -\sum_{i=0}^{255} P(m_i) \log P(m_i), \tag{3}$$

where m_i is the ith gray value and $P(m_i)$ is the probability of symbol m_i in a test image. Entropy for an 8-bit true random image is 8, which shows that the pixel values in an image are completely random. Therefore, entropy of encrypted an image for a good encryption algorithm is close to 8, and the closer the entropy is to 8, the smaller the possibility of information disclosure for the encryption scheme.

Entropy of the different images and their cipher images, encrypted by different schemes, are shown in Table 2. From this table, we know that the proposed algorithm has an entropy that is closer to 8 compared to the other schemes, which illustrates that the proposed scheme can rarely leak any useful information.

Table 2: Shannon entropy of different image encrypted by the IECTM and the improved scheme.

Image	Plain image	Ref. [3]	Ref. [4]	Ref. [18]	Proposed
Lena	7.758377	7.990966	7.997287	7.9972	7.99903
Baboon	7.774815	7.991296	7.998973	7.9972	7.99907
Girl	6.904487	7.991575	7.998989	-	7.99902
Couple	6.300791	7.991349	7.999136	-	7.99907

4.5 Differential attack

The plain image's sensitivity to an encryption scheme is the measure of the influence of a tiny change in a plain image compared to the cipher image. Differential attacks aim to find the relationship between the two plain images and their cipher images. Therefore, a slight change in a sensitive plain image can result in a completely different cipher image, and cryptanalysts cannot gather any useful information. The degree of an image's sensitivity can be reflected by $NPCR$ and $UACI$ [4] which are shown in Eq. (4).

$$NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\%, \tag{4}$$

$$UACI = \frac{1}{m \times n} \left(\sum_{i,j} \frac{|C(i,j) - C'(i,j)|}{255} \right) \times 100\%, \tag{5}$$

$$D_{i,j} = \begin{cases} 1, \text{ if } C(i,j) \neq C'(i,j), \\ 0, \text{ if } C(i,j) = C'(i,j), \end{cases} \tag{6}$$

where m, n are the size of the image; C and C' are two cipher images. An ideal $NPCR$ value is 1, which represents that a plain image has high sensitivity and therefore can resist differential attacks. The theoretical values of $UACI$ is 0.33 [4]. In our test, a randomly selected pixel from the plain image was changed, then we encrypted the changed and unchanged plain images to obtain two cipher images C and C', respectively. Comparison of $NPCR$ and $UACI$ scores between different encryption schemes are shown in Table 3 which illustrates that the proposed scheme has the largest $NPCR$ score and can therefore resist stronger differential attacks.

Table 3: $NPCR$ and $UACI$ of different encrypted image with one bit differ from the plain image.

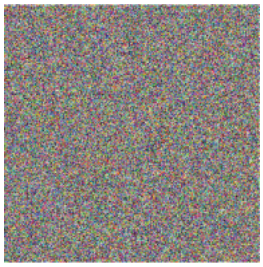
Encryption scheme		Lena	Baboon	Pepper	Barbarb
Ref. [6]	$NPCR$	0.9965	0.9955	0.9960	0.9940
	$UACI$	0.3348	0.3342	0.3340	0.3341
Ref. [3]	$NPCR$	5.0862×10^{-6}	5.0862×10^{-6}	5.0862×10^{-6}	5.0862×10^{-6}
	$UACI$	1.9946×10^{-8}	1.9946×10^{-8}	1.9946×10^{-8}	1.9946×10^{-8}
Ref. [17]	$NPCR$	0.9962	0.9943	0.9964	0.9960
	$UACI$	0.3377	0.3353	0.3353	0.3341
Ref. [18]	$NPCR$	0.9966	0.9965	0.9963	-
	$UACI$	0.3344	0.3350	0.3347	-
Proposed	$NPCR$	0.9967	0.9972	0.9970	0.9973
	$UACI$	0.3346	0.3350	0.3351	0.3355

4.6 Data loss and noise attack

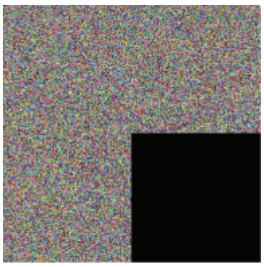
When images are transferred or stored, cipher images, noise, and data loss are inevitable. A good encryption scheme should resist these influences well. In order to test the ability of the proposed scheme in resisting noise and data loss attacks, we decrypted two cipher images which both encrypted the Lena image of size 256×256 and one was cut as Fig. 6 (b) , another was influenced by 3% 'salt&pepper' noise as Fig. 6 (c), and their deciphered images are Fig. 6 (e,f) which both contain massive visual information of the original Lena image as Fig. 6 (d). PSNR (Peak Signal to Noise Ratio) expressed in Eq. (7) is employed to evaluate the ability of a scheme to restore an image. The larger the PSNR value, the better the image recovered. Table 4 compares $NPCR$ coefficients between [6] and the proposed scheme. Experimental results show that the proposed scheme has excellent performance against noise attacks.

$$\begin{cases} PSNR = 10 \times \lg \dfrac{255^2}{MSE}, \\ MSE = \dfrac{1}{W \times H} \sum_{i=0}^{H} \sum_{j=0}^{W} (P(i,j) - D(i,j))^2, \end{cases} \tag{7}$$

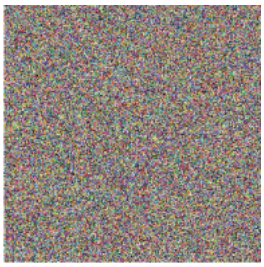
where M, N are the height and the weight of image respectively; $P(i,j)$ and $D(i,j)$ are the pixels from the plain and deciphered images respectively.



(a) Cipher image



(b) Cipher image with 25% occlusion



(c) Cipher image with 3% salt&pepper noise

(d) Decipher image of (a)

(e) Decipher image of (b)

(f) Decipher image of (c)

Figure 6: Robustness against data loss and noise attacks.

Table 4: Comparison the PSNR of the different scheme.

Encryption scheme	Ref. [6]	Proposed
Data loss	47.4199	77.0979
Noise attack	48.1770	70.0263

5. Conclusion

In this paper, we first introduced image encryption based on asynchronous permutation and diffusion, then presented its defects and proposed a synchronous permutation-diffusion encryption scheme to overcome the defects. Experiments and security comparisons between the synchronous and other schemes show that the proposed has better encryption effects and keeps all the merits of asynchronous ones.

Acknowledgements

The authors are grateful to Mr. Gregory Young for his helping to correct the English expression of paper.

This paper is supported by Chongqing Municipal Key Laboratory of Institutions of Higher Education (Grant No. [2017]3) and Chongqing Development and Reform Commission (Grant No. 2017[1007]).

References

[1] Y. Zhang, D. Xiao, *An image encryption scheme based on rotation matrix bit-level permutation and block diffusion*, Communications in Nonlinear Science Numerical Simulation 19 (2014), 74-82.

[2] J. Zhang, *An image encryption scheme based on cat map and hyperchaotic Lorenz system*, IEEE International Conference on Computational Intelligence and Communication Technology, IEEE, 2015, 78-82.

[3] C. Li, G. Luo, K. Qin et al., *An image encryption scheme based on chaotic tent map*, Nonlinear Dynamics, 87 (2017), 127-133.

[4] X. Wu, B. Zhu, Y. Hu et al., *A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps*, IEEE Access, 5 (2017), 6429-6436.

[5] X. Zhang, W. Nie, Y. Ma et al., *Cryptanalysis and improvement of an image encryption algorithm based on hyper-chaotic system and dynamic S-box*, Multimedia Tools and Applications, 2016, 1-19.

[6] C. Pak, L. Huang, *A new color image encryption using combination of the 1D chaotic map*, Signal Processing, 138 (2017), 129-137.

[7] W. Zhang, C. Zhang, C. Chen, et al., *Hybrid chaotic confusion and diffusion for physical layer security in OFDM-PON*, IEEE Photonics Journal, 9 (2017), 1-10.

[8] X.J. Tong, M. Zhang, Z. Wang, et al., *A joint color image encryption and compression scheme based on hyper-chaotic system*, Nonlinear Dynamics, 84 (2016), 2333-2356.

[9] L. Wang, H. Song, P. Liu, *A novel hybrid color image encryption algorithm using two complex chaotic systems*, Optics and Lasers in Engineering, 77 (2016), 118-125.

[10] P. Praveenkumar, R. Amirtharajan, K. Thenmozhi, et al., *Triple chaotic image scrambling on RGB a random image encryption approach*, Security and Communication Networks, 8 (2016), 3335-3345.

[11] L. Xu, Z. Li, J. Li, et al., *A novel bit-level image encryption algorithm based on chaotic maps*, Optics and Lasers in Engineering, 78 (2016)0, 17-25.

[12] H. Zhu, X. Zhang, H. Yu, et al., *An image encryption algorithm based on compound homogeneous hyper-chaotic system*, Nonlinear Dynamics, 2017, 1-19.

[13] M. Khan, T. Shah, S.I. Batool, *Construction of S-box based on chaotic Boolean functions and its application in image encryption*, Neural Computing and Applications, 27 (2016), 677-685.

[14] J. Wu, X. Liao, B. Yang, *Color image encryption based on chaotic systems and elliptic curve ElGamal scheme*, Signal Processing, 141 (2017), 109-124.

[15] R. Enayatifar, A.H. Abdullah, I.F. Isnin, et al., *Image encryption using a synchronous permutation-diffusion technique* Optics and Lasers in Engineering, 90 (2017), 146-154.

[16] Z.Hua, Y. Zhou, *Image encryption using 2D Logistic-adjusted-Sine map*, Information Sciences, 339 (2016), 237-253.

[17] X.J. Tong, M. Zhang, Z. Wang et al., *A joint color image encryption and compression scheme based on hyper-chaotic system*, Nonlinear Dynamics, 84 (2016), 2333-2356.

[18] A.Y. Niyat, M.H. Moattar, M.N. Torshiz, *Color image encryption based on hybrid hyper-chaotic system and cellular automata*, Optics and Lasers in Engineering, 90 (2017), 225-237.

Accepted: 4.07.2018