Cryptanalyzing an Image Cryptosystem Based on Circular Inter-Intra Pixels Bit-Level Permutation

YONG ZHANG
School of Software and Internet-of-Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330013, China
e-mail: zhangyong@jxufe.edu.cn

This work was supported in part by the National Natural Science Foundation of China under Grant 61762043, in part by the Natural Science Foundation of Jiangxi Province under Grant 20192BAB207022, and in part by the Scientific Research Foundation of Jiangxi Provincial Education Department under Grant GJJ190249.

ABSTRACT Recently, a new image encryption algorithm based on chaos has been proposed [Circular inter-intra pixels bit-level permutation and chaos-based image encryption, Information Sciences, 355-356: 314-327, 2016]. In this paper, this image encryption algorithm was crypt-analyzed in detail. Some defects of it, such as its row-based scrambling being invalid for the special images and the lacking of diffusion operations in its processes, were pointed out. Meanwhile, two crack methods were presented in this paper. They are using only a pair of chosen plain-cipher images or a pair of known plain-cipher images to crack the image encryption algorithm. That is enough to demonstrate the image cryptosystem using this image encryption algorithm is insecure.

INDEX TERMS Cryptanalysis, image encryption, chosen/known plain-image attack, information security.

I. INTRODUCTION

The image information security is closely related to people’s daily life. In order to realize secure image transmission and storage on the internet and cloud, a large number of image encrypting and hiding systems have been proposed [1]–[8]. With the vigorous development of image cryptography, the cryptanalysis researches on image encryption systems have also achieved fruitful results. The image encryption algorithms in [9]–[18] are cracked by the cryptographic analysis methods in [19]–[28], respectively. Among these image cryptanalysis methods, the known/chosen plain-image attacks are the most commonly used methods. Advances in image cryptanalysis technologies are prompting the researchers to continue to study more secure image encryption algorithms.

Several recent image encryption algorithms and their attack methods are itemized below.

An image encryption algorithm based on information entropy and chaotic system was proposed in [29]. In this algorithm, the pseudo-random matrices for image encryption were produced using the Sine map modulated by Logistic map. To make the encryption process relate to the plain images, the information entropy of plain image was used as the implicit control parameter of the chaotic system. The whole encryption process of the algorithm was composed of row permutation, column permutation, grayscale value transformation and weighted diffusion operations. However, all the merits of the algorithm claimed in [29] are negated in [30]. Especially, the image encryption algorithm in [29] was pointed out in [30] to have weak key sensitivity and weak plaintext sensitivity. By constructing plain images with the same information entropy, the algorithm in [29] can be deciphered by means of chosen plaintext attack.

A pseudo-random number generator based on multiple chaotic maps (i.e. Logistic map, Sine map and Chebyshev map) was proposed in [31]. Subsequently, a pseudo-random sequence, whose size is the same length as vectorized plain image, was produced by the generator. The index sequence obtained by sorting the pseudo-random sequence is used to shuffle the image vector, and the original pseudo-random sequence is used to diffuse the shuffled vector by the addition modulus operations. Then, the diffused vector is circularly left-shifted to get the ciphered image vector. However, not long after that, two crack comments on the algorithm [31] were independently proposed in [32] and [33]. Interestingly, the same method of chosen plaintext attack was employed in both [32] and [33]. It was pointed out in [33] that the algorithm in [31] can be cracked thoroughly by choosing at most $3MN$ pairs of plain-cipher images. Here, $M$ and $N$ are...
the height and width of the image, respectively. In addition, some defects of the algorithm in [31] were listed in [32], [33], and then an improvement of the algorithm was also suggested in [32], [33].

An image encryption algorithm based on Tent map was proposed in [34]. Firstly, the plain image is converted into a two-dimensional bit-image. Secondly, the bit-image is shuffled by the Line map (controlled by a pseudo-random sequence). Thirdly, the shuffled image is diffused based on row and column by XOR operation. Finally, the above steps are looped many times to get the ciphered image. This algorithm can be cracked by the chosen plaintext attack when its loop time is less than three [35]. When its loop time is greater than or equal to three, the algorithm can be cracked by the chosen plaintext attack based on codebooks [35]. The space complexity of this attack is \(O(MN)\), where \(M \times N\) is the size of the image.

A color image encryption algorithm based on DNA computing was presented in [36]. In this algorithm, the red, blue and green components of the color image are transformed into DNA encoding matrices respectively. Then, DNA addition operations are performed between different layers of DNA matrices. And circular shift operations controlled by secret key are carried out in each layer. After that, all the layers of DNA matrices are interleaved with each other. Now, each layer is separately shuffled by the key streams. Finally, another interleaving operation is executed on the different layers to get the ciphered DNA matrices. However, the shuffle operation of the image algorithm was found to be invalid for some special images [37]. Furthermore, a chosen plaintext attack method was also proposed in [37] to crack the algorithm in [36] successfully. For the plain image of size \(256 \times 256\), only two pairs of plain-cipher images are needed to obtain the equivalent keys of the algorithm in [36].

Recently, a new image encryption algorithm based on chaos was proposed, which was claimed to be secure for communication systems [38]. The work of this paper is aimed at the cryptanalysis of the algorithm in [38]. The reset of the paper is organized as follows: In Section 2, the image encryption algorithm [38] was described in detail. Then some defects of this algorithm are discussed in Section 3. And two attack methods on the algorithm are suggested in Section 4. The conclusions are in Section 5.

II. THE IMAGE ENCRYPTION ALGORITHM UNDER STUDY

In the image cryptosystem [38], the secret key is denoted by \(\{x_{01}, x_{02}, r_1, r_2\}\), requiring \(x_{01} \in [-1, 1], x_{02} \in [-1, 1], r_1 > 1,\) and \(r_2 > 1\). The plain image is assumed to be an 8-bit grayscale image, denoted by \(I\) with the size of \(M \times M\). Then the equivalent keys of the system are produced by the following steps:

**Step 1.** Use \(\{x_{01}, r_1\}\) as the initial value and parameter of (1), and then iterate (1) for \(8M^2\) times to get a state sequence denoted by \(\{y_i\}, i = 0, 1, \ldots, 8M^2 - 1\). Subsequently, use \(\{x_{02}, r_2\}\) as the initial value and parameter of (1), and then iterate (1) for \(8M^2\) times to get another state sequence denoted by \(\{y_i\}, i = 0, 1, \ldots, 8M^2 - 1\).

\[
x_{n+1} = \frac{2}{\pi} \arctan(\cot(x_n \cdot r))
\]  
(1) 

**Step 2.** Obtain a new sequence \(\{z_i\}\) from \(\{x_i\}\) and \(\{y_i\}\) by (2), \(i = 0, 1, 2, \ldots, 8M^2 - 1\).

\[
z_i = (x_i + y_i)/(1 - x_i \cdot y_i)
\]  
(2) 

**Step 3.** Produce two bit sequences \(\{a_i\}\) and \(\{b_i\}\) form \(\{z_i\}\) by (3), \(i = 0, 1, 2, \ldots, 8M^2 - 1\).

\[
a_i = \begin{cases} 
0, & |z_i| < 0.5, \\
1, & \text{otherwise}.
\end{cases}
\]

\[
b_i = \begin{cases} 
0, & |z_i| < 0.25 \text{ or } 0.5 < |z_i| < 0.75,
\end{cases}
\]  
(3) 

**Step 4.** Generate a matrix \(Col\) from \(\{a_i\}\) by (4), and with the similar method generate a matrix \(Row\) from \(\{b_i\}\). Both \(Col\) and \(Row\) are of size \(M \times M\).

\[
Col(i, j) = \text{bin2dec}(a_0(Mi + j)a_0(Mi + j) + 1 \ldots a_0(Mi+j+7)),
\]

\[
i, j = 0, 1, 2, \ldots, M - 1.
\]  
(4) 

Here, the matrices \(Col\) and \(Row\) are the equivalent keys of the system.

Now, the process of the image encryption is as follows:

**Step 1.** Perform the following circular shift operation on each row of plain image \(I\) to get a new image, denoted by \(Q\) with the size of \(M \times M\). The algorithm is shown in Fig. 1.

Take the \(i\)th row of the image \(I\) as an example to introduce the circular shift operation. Firstly, produce a bit sequence \(T\) of length \(8M\) from the elements of the \(i\)th row of \(I\), satisfying \(T(8j to 8j+7) = \text{dec2bin}(I(i, j)), j = 0, 1, 2, \ldots, M - 1\). Secondly, count the number of bit 1 in \(T\) as \(ns\). Thirdly, let \(dir = \text{mod}(ns, 2)\). If \(dir = 0\), then circularly right-shift \(T\) for \(ns\) bits; if \(dir = 1\), then circularly left-shift \(T\) for \(ns\) bits. Finally, produce the \(i\)th row of \(Q\) by \(T\) such that \(Q(i, j) = \text{bin2dec}(T(8j to 8j+7)), j = 0, 1, 2, \ldots, M - 1\).

**Step 2.** Transform \(Q\) into a new matrix \(C\) with the equivalent keys of \(Col\) and \(Row\) as follows:

Let \(C=Q\), then

For \(i = 1\) to \(M\), do 

\[C(i, 1 to M) = C(i, 1 to M) \oplus Row(i, 1 to M),\]

End.

And \(C\) is the ciphered image.

As for the system, its decryption process is the inverse of its encryption process. For more details about the decryption process, please refer to [38].

III. SHORTCOMINGS OF THE SCHEME UNDER STUDY

According to Section II, the system in [38] consists of two steps. The first step is a row-based circular shift process, and the second step is a row-and-column-based XOR process. Both the steps have some defects.
In the first step, if the plain image is an all-white or an all-black image, then the circular shift operation does not work for such images. In fact, if each row of the plain image satisfies one of the following conditions, the first step will not work.

**Condition 1:** The number of bit 1 is an integer multiple of $8k$, and each $k$ adjacent bytes are the same. Where, $\text{mod}(M, 8k) = 0$, and $M$ is the width of the image.

**Condition 2:** The number of bit 1 is an integer multiple of $n$, and each $n$ adjacent bits are the same. Where, $n$ is a positive integer, and $\text{mod}(M, n) = 0$.

Let $M = 256$. For example, when each row of the image is $(171, 171, \ldots, 171)$, i.e. $(10101011, 10101011, \ldots, 10101011)$ (in binary format), satisfying Condition 1 ($k = 1$), it makes the first step of the system invalid. In another example, each row of the image is $(113, 33, 113, 33, \ldots, 113, 33)$, i.e. $(01110001, 00100001, 01110001, 00100001, \ldots, 01110001, 00100001)$ (in binary format), satisfying Condition 2 ($n = 16$), and this image also makes the first step of the system invalid.

According to Kerckhoff’s rule, it is assumed that one can use the encryption device at will. Therefore, the enemy can choose those images satisfying one of two above conditions as the plain images, thus bypassing the first step of the encryption system.

In the second step of the algorithm, there is no diffusion operation. The equivalent form of Step 2 in Section 2 is:

$$C(i, j) = Q(i, j) \oplus \text{Row}(i, j) \oplus \text{Col}(j, i)$$  \hspace{1cm} (5)

where, each symbol comes from Section II.

If let the matrix $S = \text{Row} \oplus \text{transpose(Col)}$, then $C = Q \oplus S$. Here, $S$ is the equivalent key that the enemy will attack.

According to its second step, the system has no diffusion function. So this system is plain-image insensitive. Here, take the image Lena of size $256 \times 256$ as the plain image (as shown in Fig. 2a), and take the secret key as $(x_01, x_02, r_1, r_2) = (0.68775492511773, -0.0134623354671, 5.938725025421, 10.257490188615) \text{ (partly from [38])}$. Then, use the encryption system to encrypt Lena to get its ciphered image (as shown in Fig. 2b). Use the decryption system to decrypt Fig. 2b to get its recovered image (as shown in Fig. 2c). Under these conditions, use the indexes of NPCR, UACI [2] and BACI [8] to test the plain-image sensitivity of the system. Here, conduct 1000 experiments to calculate the mean value of each index. In each experiment, randomly choose one pixel from the image Lena, and then randomly change one bit of the chosen pixel. And use the two images before and after the change as the plain test images. The experimental results of plain-image sensitivity are as shown in Table 1.

![FIGURE 1. Flowchart of Step 1 of image encryption algorithm.](image1)

![FIGURE 2. The encryption and decryption results of the system. (a) Lena; (b) Cipher image of Lena; (c) Recovered image of (b).](image2)

| Item | NPCR | UACI | BACI |
|------|------|------|------|
| Theoretical value (%) | 99.6094 | 33.4635 | 26.7712 |
| Calculated value (%) | 0.2970 | 0.1014 | 0.1605 |

From Table 1, one can see that the calculated value of each index deviates seriously from its respective theoretical value, indicating that the system is not plain-image sensitive.

**IV. CRACK METHODS ON THE SYSTEM UNDER STUDY**

According to the analysis in Sections II-III, the enemy only needs to attack the equivalent key $S$ in Section III. This section will discuss two attack methods on the system in [38].

**A. CHOSEN PLAIN-IMAGE ATTACK**

If the enemy can freely use the encryption device of the system in [38], he can arbitrarily choose the plain image and get the corresponding ciphered image. According to Section III, if he chooses the all-black image as the plain image, he will...
get the corresponding ciphered image which is \( S \), i.e. the equivalent key of the system.

The steps of chosen plain-image attack are as follows:

**Step 1.** Assume that the ciphered image to be attacked is labeled by \( C_1 \) with the size of \( M \times M \).

**Step 2.** Choose the all-black image of size \( M \times M \) as the input of the encryption device, and obtain the corresponding ciphered image, i.e. \( S \).

**Step 3.** Let \( Q_1 = C_1 \oplus S \).

**Step 4.** Execute the following circular shift operation on each row of the image \( Q_1 \) to get a new image, denoted by \( I_1 \) with the size of \( M \times M \).

Take the \( i \)-th row of the image \( Q_1 \) as an example to introduce the circular shift operation. Firstly, produce a bit sequence \( T \) of length \( 8M \) from the elements of the \( i \)-th row of \( Q_1 \), satisfying \( T(8j) \rightarrow 8(j+1) = \text{dec2bin}(Q_1(i,j)) \), \( j = 0, 1, 2, \ldots, M-1 \). Secondly, count the number of bit 1 in \( T \) as \( ns \). Thirdly, let \( dir = \text{mod}(ns, 2) \). If \( dir = 1 \), then circularly right-shift \( T \) for \( ns \) bits; if \( dir = 0 \), then circularly left-shift \( T \) for \( ns \) bits. Finally, produce the \( i \)-th row of \( I_1 \) by \( T \) such that \( I_1(i,j) = \text{bin2dec}(T(8j) \rightarrow 8(j+1)) \), \( j = 0, 1, 2, \ldots, M-1 \).

The image \( I_1 \) is the recovered image obtained by attacking \( C_1 \), and \( I_1 \) is exactly the same as the original input plain image.

**Attacking Example 1:** Assume that the plain image is a \( 4 \times 4 \)-sized matrix, denoted by \( I \) as shown in Fig. 3, and the secret key is set to \( K = (0.6878, -0.0135, 5.9387, 10.2575) \). Meanwhile, assume that the attacker can obtain the ciphered image of all-black image of size \( 4 \times 4 \), which is denoted by \( S \) as shown in Fig. 3. According to the above steps of chosen plain-image attack, the ciphered image \( C_1 \) is XORed with \( S \) to get the matrix \( Q_1 \), as shown in Fig. 3. Then, according to the inverse process of Fig. 1, \( Q_1 \) is converted into the matrix \( I_1 \). From Fig. 3, one can see that \( I_1 \) is exactly the same as the original plain image \( I \).

**Attacking Example 2:** Assume that there are three encryption devices, and their secret keys are \( K_1 = (0.6878, -0.0135, 5.9387, 10.2575) \), \( K_2 = (0.6006, -0.7162, 33.5067, 6.0334) \), and \( K_3 = (-0.1565, 0.8315, 18.1623, 27.9655) \), respectively. Use these three devices to encrypt the plain images (all of size \( 256 \times 256 \), as shown in Figs. 4a-4c, respectively) separately to get their corresponding ciphered images (as shown in Figs. 4d-4f, respectively).

Now, assume that the enemy can freely use these three encryption devices, and the enemy has intercepted the three ciphered images shown in Figs. 4d-4f. The enemy employs the above-mentioned method of chosen plain-image attack. And after he input the all-black image of size \( 256 \times 256 \) into the three devices, he will get the ciphered image of each device, i.e. the equivalent key of each device (as shown in Figs. 5a-5c, respectively). Then, he can continue cracking the ciphered images in Figs. 4d-4f to get their recovered images, as shown in Figs. 5d-5f.

Comparing Figs. 4a-4c with Figs. 5d-5f, one can see that the attacker has succeeded in cracking the encryption system with only one all-black plain image and its ciphered image. In the computer equipped with Intel®Core™i7-8650U@1.90GHz, 16GB RAM and MATLAB R2018a, the chosen plain-image attack time for cracking one ciphered image of size \( 256 \times 256 \) is about 0.4581s.

**B. KNOWN PLAIN-IMAGE ATTACK**

Sometimes the enemy cannot address the encryption device of opposite party, so he cannot conduct the chosen plain-image attack. But in general, the enemy can get some pairs of plain-cipher images more easily. In this section, one can see that the system in [38] is unsecure confronting the known plain-image attack. Only one pair of known plain-cipher images can reveal the equivalent key \( S \) of the system.

The steps of known plain-image attack are as follows:

**Step 1.** Denote the image to be attacked as \( C_1 \) with the size of \( M \times M \).
Step 2. Assume that the enemy has known a plain image \( P_0 \) of size \( M \times M \) and its corresponding ciphered image \( C_0 \).

Step 3. Execute the following circular shift operation on each row of the image \( P_0 \) to get a new image, denoted by \( Q_0 \) with the size of \( M \times M \).

Take the \( i \)th row of the image \( P_0 \) as an example to introduce the circular shift operation. Firstly, produce a bit sequence \( T \) of length \( 8M \) from the elements of the \( i \)th row of \( P_0 \), satisfying \( T(8j \text{ to } 8j + 7) = \text{dec2bin}(P_0(i,j)), j = 0, 1, 2, \ldots, M - 1 \). Secondly, count the number of bit 1 in \( T \) as \( ns \). Thirdly, let \( \text{dir} = \text{mod}(ns,2) \). If \( \text{dir} = 0 \), then circularly right-shift \( T \) for \( ns \) bits; if \( \text{dir} = 1 \), then circularly left-shift \( T \) for \( ns \) bits. Finally, produce the \( i \)th row of \( Q_0 \) by \( T \) such that \( Q_0(i,j) = \text{bin2dec}(T(8j \text{ to } 8j + 7)), j = 0, 1, 2, \ldots, M - 1 \).

Step 4. Let \( S = C_0 \oplus Q_0 \). And \( S \) is the equivalent key of the encryption device.

Step 5. Let \( Q_1 = C_1 \oplus S \).

Step 6. Execute the following circular shift operation on each row of the image \( Q_1 \) to get a new image, denoted by \( I_1 \) with the size of \( M \times M \).

Take the \( i \)th row of the image \( Q_1 \) as an example to introduce the circular shift operation. Firstly, produce a bit sequence \( T \) of length \( 8M \) from the elements of the \( i \)th row of \( Q_1 \), satisfying \( T(8j \text{ to } 8j + 7) = \text{dec2bin}(Q_1(i,j)), j = 0, 1, 2, \ldots, M - 1 \). Secondly, count the number of bit 1 in \( T \) as \( ns \). Thirdly, let \( \text{dir} = \text{mod}(ns,2) \). If \( \text{dir} = 1 \), then circularly right-shift \( T \) for \( ns \) bits; if \( \text{dir} = 0 \), then circularly left-shift \( T \) for \( ns \) bits. Finally, produce the \( i \)th row of \( I_1 \) by \( T \) such that \( I_1(i,j) = \text{bin2dec}(T(8j \text{ to } 8j + 7)), j = 0, 1, 2, \ldots, M - 1 \).

The image \( I_1 \) is the recovered image obtained by attacking \( C_1 \), and \( I_1 \) is exactly the same as the original plain image.

Attacking Example 3: Assume that the plain image is a \( 4 \times 4 \)-sized matrix, denoted by \( I \) as shown in Fig. 6, and the secret key is set to \( K = (-0.6263, -0.0205, 13.2808, 25.3911) \). Meanwhile, assume that the attacker can obtain the ciphered
image \( C_0 \) of a known image \( P_0 \) of size \( 4 \times 4 \). Both \( P_0 \) and \( C_0 \) are shown in Fig. 6. According to the above steps of known plain-image attack, \( P_0 \) is converted into \( Q_0 \) with the help of Fig. 1, and the ciphered image \( C_0 \) is XORed with \( Q_0 \) to get the matrix \( S \), as shown in Fig. 6. Then, the ciphered image \( C_1 \) is XORed with \( S \) to get the matrix \( Q_1 \). Finally, according to the inverse process of Fig. 1, \( Q_1 \) is converted into the matrix \( I_1 \). From Fig. 6, one can see that \( I_1 \) is exactly the same as the original plain image \( I \).

Attacking Example 4: Suppose that there are three encryption devices based on the algorithm in [38], whose secret keys are \( K_1 = (-0.6263, -0.0205, 13.2808, 25.3911) \), \( K_2 = (-0.1088, 0.2926, 24.6529, 9.8784) \), and \( K_3 = (0.4187, 0.5094, 8.5699, 19.9509) \), respectively. Employ these three devices in turn to encryption the images in Figs. 7a-7c separately to obtain their corresponding cipher images (as shown in Figs. 7d-7f, respectively). Then, employ these three devices in turn to encryption the images in Figs. 8a-8c separately to obtain their corresponding cipher images (as shown in Figs. 8d-8f, respectively).

Now, assume that the enemy cannot address the three devices, but he intercepts three pairs of plain-cipher images as shown in Fig. 8, i.e. he obtains one pair of plain-cipher images for each device. Simultaneously, the enemy intercepts the ciphered images as shown in Figs. 7d-7f. According to the above-mentioned method of known plain-image attack, the enemy can reveal the equivalent key of each device (as shown in Figs. 9a-9c), then he can crack the ciphered images in Figs. 7d-7f to get their recovered images (as shown in Figs. 9d-9f, respectively).

Comparing Figs. 9d-9f with Figs. 7a-7c, one can see that the cracked images are separately identical to their original plain images, which means that only one pair of known plain-cipher images are required to successfully crack the system in [38].

In the same computer used in Example 2, for the image of size \( 256 \times 256 \), it will take about 0.6484s to crack the encryption system using MATLAB 2018a with the known plain-image attack method.

V. CONCLUSION

The image encryption algorithm in [38] was reviewed in detail in this paper. It was pointed out that there are some shortcomings in this scheme, such as the scrambling operation does not work on many special images, and the algorithm possesses no diffusion operations. Moreover, two crack methods on the system were proposed, and the crack time of both methods on the used computer is less than 1s for the image of size \( 256 \times 256 \). The research work in this paper shows that the methods of both chosen plain-image attack and known plain-image attack can successfully crack the system in [38] with only one pair of plain-cipher images.

Furthermore, according to the crack methods in this paper, it is easy to know that if the enemy can freely use the decryption system in [38], then he can crack the decryption system with the method of chosen cipher-image attack, and get the
equivalent key of the system with only one pair of cipher-
plain images. Therefore, the image cryptosystem based on
the algorithm in [38] is insecure and cannot be applied to the
actual image security system.

REFERENCES

[1] X. Ouyang, Y. Luo, J. Liu, L. Cao, and Y. Liu, “A color image encryption
method based on memristive hyperchaotic system and DNA encryption,”
Int. J. Modern Phys. B, vol. 34, no. 04, Feb. 2020, Art. no. 2050014.

[2] G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme
based on 3D chaotic cat maps,” Chaos, Solitons Fractals, vol. 21, no. 3,
p. 749–761, Jul. 2004.

[3] X. Wang and H. Sun, “A chaotic image encryption algorithm based on
improved Joseph trasval and cyclic shift function,” Opt. Laser Technol.,
vol. 122, Feb. 2020, Art. no. 105854.

[4] X. Wang and S. Gao, “Image encryption algorithm for synchronously
updating Boolean networks based on matrix semi-tensor product theory,”
Inf. Sci., vol. 507, pp. 16–36, Jan. 2020.

[5] D. Huang and J. Wang, “High-capacity reversible data hiding in encrypted
image based on specific encryption process,” Signal Process. Image Comm.,
vol. 80, Feb. 2020, Art. no. 115632.

[6] H. Li, L. Deng, and Z. Gu, “A robust image encryption algorithm based on
a 32-bit chaotic system,” IEEE Access, vol. 8, pp. 30127–30151, Feb. 2020.

[7] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-El-Yazeed,
“A novel image encryption system merging fractional-order edge detection and
generalized chaotic maps,” Signal Process., vol. 167, Feb. 2020, Art. no. 107280.

[8] Y. Zhang, “The unified image encryption algorithm based on chaos and
cubic S-Box,” Inf. Sci., vol. 450, pp. 361–377, Jun. 2018.

[9] A. Shafique and J. Shahid, “Novel image encryption cryptosystem based on
binary bit planes extraction and multiple chaotic maps,” Eur. Phys. J. Plus,
vol. 133, no. 8, p. 331, Aug. 2018.

[10] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and Y. Zhang, “Reusing the permu-
tation matrix dynamically for efficient image cryptographic algorithm,”
Signal Process., vol. 111, pp. 294–307, Jun. 2015.

[11] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, “An efficient image
encryption algorithm using lookup table-based confusion and diffu-
sion,” Nonlinear Dyn., vol. 81, no. 3, pp. 1151–1166, Apr. 2015.

[12] A. M. Elshamy, A. N. Z. Rashed, E. A.-N.-A. Mohamed, O. S. Faragalla,
Y. Mu, S. A. Alshebeili, and F. E. El-Samie, “Optical image encryption
based on chaotic baker map and double random phase encoding,” J. Lightw.
Technol., vol. 31, no. 15, pp. 2533–2539, Aug. 2013.

[13] M. A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, R. M. López-Gutiérrez,
and O. R. A. Del Campo, “A RGB image encryption algorithm based on total plain image characteristics and
chaos,” Signal Process., vol. 109, pp. 119–131, Apr. 2015.

[14] Z. Hua and Y. Zhou, “Image encryption using 2D logistic-adjusted-sine model,”
Inf. Sci., vol. 339, pp. 237–253, Apr. 2016.

[15] L. Tawalbeh, M. Mouwaf, and W. Aljoby, “Use of elliptic curve cryptogra-
phy for multimedia encryption,” IET Inf. Secur., vol. 7, no. 2, pp. 67–74,
Jun. 2013.

[16] J. Wu, X. Liao, and B. Yang, “Image encryption using 2D Hénon-sine map
and DNA approach,” Signal Process., vol. 153, pp. 11–23, Dec. 2018.

[17] Z. Hua, S. Yi, and Y. Zhou, “Medical image encryption using high-
speed scrambling and pixel adaptive diffusion,” Signal Process., vol. 144,
pp. 134–144, Mar. 2018.

[18] X. Wang, U. Çavuşoğlu, S. Kacar, A. Akgul, V.-T. Pham, S. Jafari,
F. Alsaadi, and X. Nguyen, “S-Box based image encryption application
using a chaotic system without equilibrium,” Appl. Sci., vol. 9, no. 4,
p. 781, Feb. 2019.

[19] M. Li, D. Lu, Y. Xiang, Y. Zhang, and H. Ren, “Cryptanalysis and
improvement in a chaotic image cipher using two-round permutation and
diffusion,” Nonlinear Dyn., vol. 96, no. 1, pp. 31–47, Jan. 2019.

[20] H. Diab and A. M. El-samary, “Cryptanalysis and improvement of the
image cryptosystem reusing permutation matrix dynamically,” Signal Pro-
cess., vol. 148, pp. 172–192, Jul. 2018.

[21] M. Li, M. Xu, J. Luo, and H. Fan, “Cryptanalysis of an image encryption
using 2D henon-sine map and DNA approach,” IEEE Access, vol. 7,
p. 63336–63345, May 2019.

[22] J. Chen, N. Bao, J. Li, Z.-L. Zhu, and L. Y. Zhang, “Cryptanalysis of optical
ciphers integrating double random phase encoding with permutation,”
IEEE Access, vol. 5, pp. 16124–16129, Aug. 2017.

[23] H. Fan, M. Li, D. Lü, and K. An, “Cryptanalysis of a plaintext-related
chaotic RGB image encryption scheme using total plain image char-
acteristics,” Multimedia Tools Appl., vol. 77, no. 15, pp. 20103–20127,
Aug. 2018.

[24] W. Feng, Y. He, H. Li, and C. Li, “Cryptanalysis and improvement of the
image encryption scheme based on 2D logistic-adjusted-sine map,” IEEE
Access, vol. 7, pp. 12584–12597, Jan. 2019.

[25] M. S. Koirjom, D. S. Laihiprakam, and T. Themrichon, “Cryptanalysis of
multimedia encryption using elliptic curve cryptography,” Optik, vol. 168,
p. 370–375, Sep. 2018.

[26] J. Chen, L. Chen, and Y. Zhou, “Cryptanalysis of a DNA-based image
encryption scheme,” Inf. Sci., vol. 520, pp. 130–141, May 2020.

[27] Y. Chen, C. Tang, and R. Ye, “Cryptanalysis and improvement of medical
image encryption using high-speed scrambling and pixel adaptive diffu-
sion,” Signal Process., vol. 167, Feb. 2020, Art. no. 107286.

[28] Q. Lu, C. Zhu, and X. Deng, “An efficient image encryption scheme
based on the LSS chaotic map and single S-box,” IEEE Access, vol. 8,
p. 25664–25678, Jan. 2020.

[29] G. Ye, C. Pan, X. Huang, Z. Zhao, and J. He, “A chaotic image encryption
algorithm based on information entropy,” Int. J. Bifurcation Chaos, vol. 28,
n. 1, Jan. 2018, Art. no. 1850010.

[30] C. Li, D. Lin, B. Feng, J. Lu, and F. Hao, “Cryptanalysis of a chaotic image
encryption algorithm based on information entropy,” IEEE Access, vol. 6,
p. 75834–75842, Nov. 2018.

[31] C. Pak and L. Huang, “A new color image encryption using combination of
the 1D chaotic map,” Signal Process., vol. 138, pp. 129–137, Sep. 2017.

[32] J. Chen, F. Han, W. Qian, Y.-D. Yao, and Z.-L. Zhu, “Cryptanalysis and
improvement in an image encryption scheme using combination of the 1D
chaotic map,” Signal Process., vol. 93, no. 4, pp. 2399–2413, May 2018.

[33] H. Wang, D. Xiao, C. Chen, and H. Huang, “Cryptanalysis and enhance-
ments of image encryption using combination of the 1D chaotic map,”
Signal Process., vol. 144, pp. 444–452, Mar. 2018.

[34] G. Zhou, D. Zhang, Y. Liu, Y. Yuan, and Q. Liu, “A novel image encryption
algorithm based on chaos and line map,” Neurocomputing, vol. 169,
p. 150–157, Dec. 2015.

[35] L. Chen, B. Ma, X. Zhao, and S. Wang, “Differential cryptanalysis of a
novel image encryption algorithm based on chaos and line map,” Nonlin-
ear Dyn., vol. 87, no. 3, pp. 1797–1807, Feb. 2017.

[36] M. Kumar, A. Iqbal, and P. Kumar, “A new RGB image encryption
algorithm based on DNA encoding and elliptic curve Diffie–Hellman
encryption,” Signal Process., vol. 125, pp. 187–202, Aug. 2016.

[37] A. Akhavan, A. Samsudin, and A. Akhshani, “Cryptanalysis of an image
encryption algorithm based on DNA encoding,” Opt. Laser Technol.,
vol. 95, pp. 94–99, Oct. 2017.

[38] A.-V. Diacone, “Circular inter–intra pixels bit-level permutation
and chaos-based image encryption,” Inf. Sci., vol. 355–356, pp. 314–327,
Aug. 2016.