Image content-based encryption algorithm using high-dimensional chaotic system

Zhongyun Hua†, Binghang Zhou‡ and Yicong Zhou†*

†Department of Computer and Information Science, University of Macau, Macau 999078, China
Email:huazyum@gmail.com, *yicongzhou@umac.mo
‡College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
Email:hn_zhoubh@qq.com

Abstract—This paper introduces a 2D chaotic system, called 2D-LSM. It is derived from the Logistic and Sine maps. Using 2D-LSM, a new image encryption algorithm based on the image content is also proposed. It can encrypt different kinds of images into random-like ones. Simulation results and security analysis show that the proposed image encryption algorithm can encrypt images with a high security level.

1. Introduction

Image encryption attracts more and more attentions in the past few decides [1–3]. Because there are many similar properties between the chaotic systems and image encryption, the chaotic systems are widely used in image encryption. Many chaos-based image encryption algorithms have been developed [4–6]. When chaotic systems are used in image encryption, their chaos performance usually partly determines the security level. For some 1D chaotic systems, they usually have simple structures and poor chaos performance that make their behaviors easily to be predicted [7].

In this paper, we first introduce a new 2D chaotic system, which is derived from the Logistic and Sine maps. Using the proposed 2D chaotic system, we further propose an image encryption algorithm. The image content is used as a portion of security key to generate the initial conditions of 2D chaotic system. Simulation results and security analysis show that the proposed algorithm has high encryption performance.

The rest of this paper is organized as follows. Section 2 introduces a new 2D chaotic system; Section 3 proposes an image encryption algorithm; Section 4 simulates the image encryption algorithm and analyzes its performance and Section 5 gets a conclusion.

2. The 2D Logistic-Sine map

Firstly, we give the definition of the 2D Logistic-Sine map (2D-LSM).

$$\begin{cases} x_{i+1} = a(3\sin(\pi y_i) + 1)x_i(1 - x_i) \\ y_{i+1} = a(3\sin(\pi x_{i+1}) + 1)y_i(1 - y_i) \end{cases} \quad (1)$$

where a is the control parameter within the range of $[0, 1]$. From its definition, we can see that the 2D-LSM is derived from the Logistic and Sine maps. It first uses the output of Sine map to control the parameter of Logistic map, and then extends the phase plane from 1D to 2D. Fig. 1 shows the distribution of its attractors in the 2D phase plane.

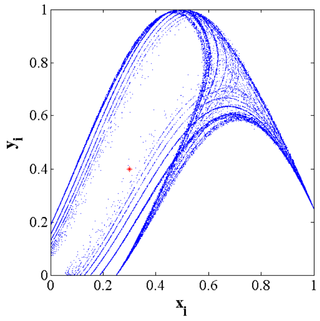

Figure 1: The distribution of attractors of 2D-LSM.

3. Image encryption using 2D-LSM

This section proposes an image encryption algorithm using 2D-LSM introduced in the previous section. Its structure diagram is depicted in Fig. 2. As can be seen, P is the original image that to be encrypted and C is the encrypted result. P is first hashed to get a fixed length bit-stream. The bit-stream is combined with the encryption key to generate initial conditions for 2D-LSM. For 2D-LSM, its output chaotic

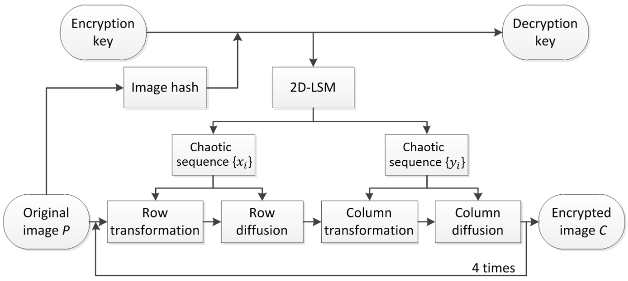

Figure 2: The structure diagram of the proposed image encryption algorithm.

sequence $\{x_i|i = 1, 2, \cdots\}$ is used to do the row transform and diffusion and its another output sequence $\{y_i|i = 1, 2, \cdots\}$ is used to do the column transform and diffusion. After 4 times of these operations, the original image P can be encrypted into a random-like encrypted image C. The decryption is the inverse operations of the encryption and the decryption key is generated by appending the hash result of P to the end of the encryption key. The encryption procedure is represented as $C = En(P, K_e)$ and the decryption procedure is represented as $D = De(C, K_d)$.

3.1. Security key schedule

The encryption key is combined with the hash result of the original image to generate the initial conditions for 2D-LSM. Here, a 128-bit stream H will be generated by the hash operation and the encryption key $K_e = \{x_0, y_0, a_0\}$. Firstly, transform H into a float number T and an integer vector $P = \{p_1, p_2, p_3, p_4\}$. Then the detailed operations to generate initial conditions for 2D-LSM in each iteration are defined as

$$\begin{cases} x_0^i = (x_0 + T \times P(i)) \mod 1 \\ y_0^i = (y_0 + T \times P(i)) \mod 1 \\ a_0^i = ((a_0 + T \times P(i)) \mod 0.1) + 0.9 \end{cases} \tag{2}$$

where $i = 1, 2, 3, 4$. From Eq. (2), we can get that the initial values (x_0^i, y_0^i) are into the range of $[0, 1]$ and the control parameter a_0^i falls into the range of $[0.9, 1]$ to make 2D-LSM achieve good chaos performance. Total 4 groups of initial conditions $\{x_0^i, y_0^i, a_0^i\}$ $(i = 1, 2, 3, 4)$ can be generated. They will be used for 2D-LSM to generate chaotic sequences $X = \{x_i|i = 1, 2, \cdots\}$ and $Y = \{y_i|i = 1, 2, \cdots\}$ in each encryption/decryption iteration.

3.2. Transformation

The transformation is to randomly change the pixel positions within the image. Here, we randomly change the pixel positions horizontally and vertically. The row transformation is to shuffle the pixel positions in each row while the column transformation shuffles the pixel positions in each column, which are similar with the operations proposed in [5].

3.2.1. Row transformation

Suppose the original image is with size of $M \times N$, a chaotic sequence $S_1 = \{x_1, x_2, \cdots, x_N\}$ is generated by 2D-LSM. Sort S_1 to get the index vector I and the sorted sequence $\mathcal{S}_1$. Then

$$\mathcal{S}_1(i) = S_1(I(i)) \quad \text{for} \quad i = 1, 2, \cdots, N \tag{3}$$

Using the index vector I, a row transformation matrix W_1 can be generated

$$W_1(i, j) = \begin{cases} 1 & \text{if} \quad I(i) = j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $i, j \in [1, N]$. Then the row transformation is defined as $C = PW_1$.

In the decryption procedure, the inverse row transformation is defined as $P = CW_1^{-1}$.

3.2.2. Column transformation

A chaotic sequence $S_2 = \{y_1, y_2, \cdots, y_M\}$ is generated by 2D-LSM. Sort S_2 to get the index vector R and the sorted sequences $\mathcal{S}_2$. Then

$$\mathcal{S}_2(i) = S_2(R(i)) \quad \text{for} \quad i = 1, 2, \cdots, M \tag{5}$$

Using the index vector R, a column transformation matrix W_2 can be obtained

$$W_2(i, j) = \begin{cases} 1 & \text{if} \quad R(j) = i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $i, j \in [1, M]$. Then the column transformation is defined as $C = W_2P$.

In the decryption procedure, the inverse column transformation is defined as $P = W_2^{-1}C$.

3.3. Diffusion

The diffusion operation is to make the little change in the original image spread over all the pixels in the encrypted image. In our algorithm, we do the diffusion operations in the finite field $(GF(2^8))$.

3.3.1. Row diffusion

When doing row diffusion, a chaotic sequence $S_3 = \{x_1, x_2, \cdots, x_{N\times N}\}$ is generated by 2D-LSM. Firstly, rearrange S_3 with the size of $N \times N$ and then convert its elements into the range of $[0, 255]$ by $S_3 = \lfloor S_3 \times 2^{20} \rfloor \mod 256$.

Convert the values in the original image P and S_3 into the finite field $GF(2^8)$, and then the row diffusion is defined as $C = PS_3$.

The inverse operation of row diffusion is defined as $P = CS_3^{-1}$.

3.3.2. Column diffusion

In the procedure of column diffusion, a chaotic sequence $S_4 = \{y_1, y_2, \cdots, y_{M\times M}\}$ is generated by 2D-LSM. We first rearrange S_4 with the size of $M \times M$ and then convert its elements into the range of $[0, 255]$ by $S_4 = \lfloor S_4 \times 2^{20} \rfloor \mod 256$.

Convert the values in the original image P and S_4 into the finite field $GF(2^8)$, and then the column diffusion is defined as $C = S_4P$.

The inverse operation of column diffusion is defined as $P = S_4^{-1}C$.

4. Simulation results and security analysis

This section simulates the proposed image encryption algorithm and analyzes its security performance.

4.1. Simulation results

The proposed image encryption algorithm can encrypt different kinds of images into random-like ones with a high security level. Fig. 3 shows the simulation procedures of the grayscale and color images. As can be seen, the original images are all with some patterns, especially the text image in Fig. 3(a), but their encrypted images are all randomly distributed. Attackers can not obtain any useful information by analyzing their pixel statistic distributions.

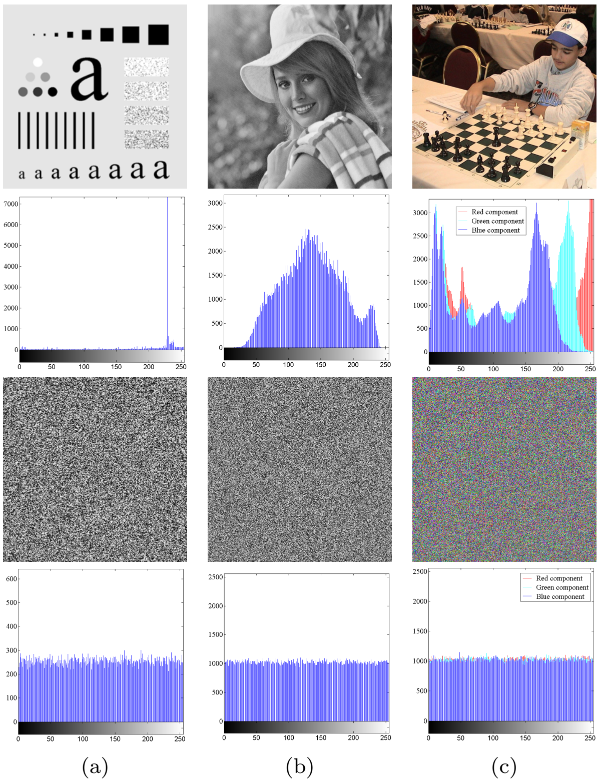

Figure 3: Simulation results of different types of images. (1) The text image; (b) the natural image and (c) the color image.

4.2. Security analysis

For a good encryption algorithm, it should have high security level. This sub-section analyzes the security level of the proposed image encryption algorithm.

4.2.1. Key sensitivity

The encryption algorithm should be very sensitive with its key's change. This means that when the encryption/decryption key has little changes, the encryption/decryption results should be totally different.

For the proposed image encryption algorithm, we randomly generate two encryption keys K_{e1} and K_{e2}, which have one bit difference. K_{d1} is the corresponding decryption key of K_{e1}. K_{d2} and K_{d3} are two decryption keys that have one bit difference with K_{d1} and they are also different with each other. Fig. 4 shows the analysis results of key sensitivity. As can be seen, when encrypting the original image with K_{e1} and K_{e2}, the two encryption results are totally different, which can be verified by Fig. 4(d). When decrypting the encrypted image C_1 with K_{d1}, K_{d2} and K_{d3}, respectively, only the decryption result D_1 can successfully reconstruct the original image. Using other decryption keys that only have one bit difference result in random-like images (Figs. 4(f) and (g)), which are also different with each other (Fig. 4(h)).

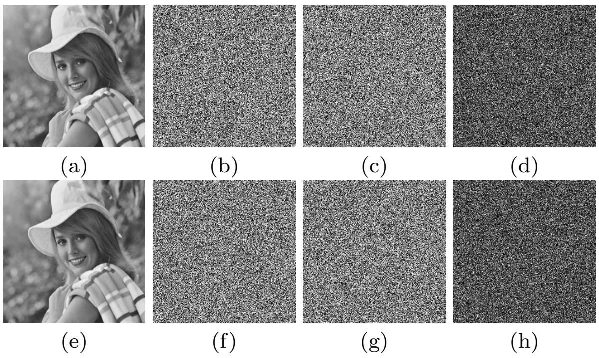

Figure 4: Key sensitivity analysis. (a) The original image P; (b) the encrypted image $C_1 = En(P, K_{e1})$; (c) the encrypted image $C_2 = En(P, K_{e2})$; (d) the difference between C_1 and C_2, $|C_1 - C_2|$; (e) the decrypted image $D_1 = De(C_1, K_{d1})$; (f) the decrypted image $D_2 = De(C_1, K_{d2})$; (g) the decrypted image $D_3 = De(C_1, K_{d3})$; (h) the difference between D_2 and D_3, $|D_2 - D_3|$.

4.2.2. Adjacent pixels correlation

Natural images usually have high data redundancy, and thus their pixels are highly related with the adjacent pixels. A secure image encryption algorithm should have the ability to break up the high correlations between adjacent pixels.

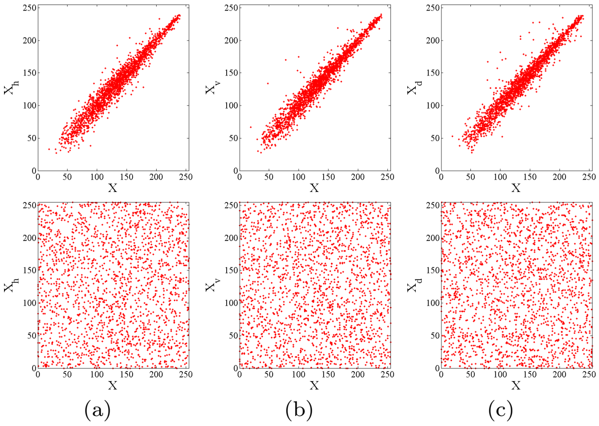

Figure 5: Pixel correlation analysis results. The first and second rows plot the distributions of adjacent pixel pairs of the original and its encrypted images along with the (a) horizontal, (b) vertical and (c) diagonal directions. The original and its encrypted images are from Figs. 4(a) and (b), respectively.

In our experimental, we randomly select 2000 pixels in the original and its encrypted images, and then plot these pixels with their adjacent pixels along with the horizontal, vertical and diagonal directions. The original and its encrypted images are from Figs. 4(a) and (b), respectively. Fig. 5 plots the results. As can be seen, for the original image (Fig. 4(a)), the pixel pairs of the randomly selected pixels with their adjacent pixels are mostly distributed on or nearby the diagonal line, which means that the pixels in the original image have high correlation with their adjacent pixels. For its encrypted image (Fig. 4(b)), the pixel pairs are randomly distributed in the whole data range, which means the pixels are highly independent and they have no relationship with their adjacent pixels.

4.2.3. Differential attack

The differential attack is one of the commonly used chosen-plaintext attacks. By encrypting little different original images with the same key, the attackers attempt to find the regular connections between the encrypted results.

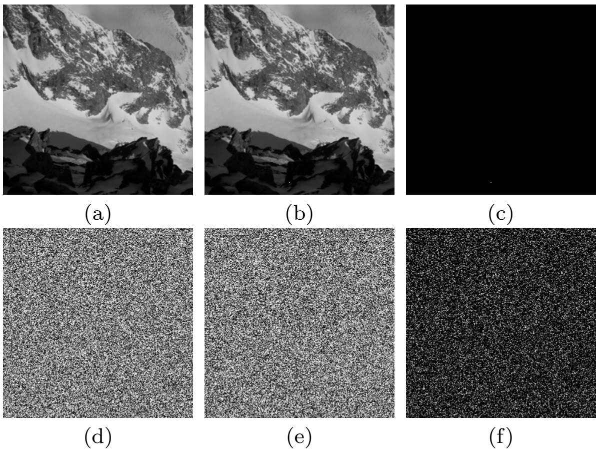

Figure 6: Differential attack analysis results. (a) The original image P_1; (b) the original image P_2 that has one pixel difference with P_1; (c) the difference between P_1 and P_2, $|P_1 - P_2|$; (d) the encrypted image $C_1 = En(P_1, K_e)$; (e) the encrypted image $C_2 = En(P_2, K_e)$; (f) the difference between C_1 and C_2, $|C_1 - C_2|$.

We simulate the procedure of the differential attack in our experiment and Fig. 6 shows the simulation results. As can be seen, by randomly changing one pixel of an original image to obtain another original image, and then encrypt the two original images using the same encryption key, the two encrypted images are random-like and totally independent. They have no relationship with each other, which can be seen from their difference in Fig. 6(f).

5. Conclusion

In this paper, we have introduced a new 2D chaotic system, which is derived from the Logistic and Sine maps. Using this new 2D chaotic system, we have further propose an image encryption algorithm. It can encrypt digital images into noise-like ones. Simulation results and security analysis have shown that the proposed image encryption algorithm can encrypt images with a high security level.

Acknowledgments

This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/017/2012/A1 and by the Research Committee at University of Macau under Grants MYRG2014-00003-FST, MRG017/ZYC/2014/FST, MYRG113(Y1-L3)-FST12-ZYC and MRG001/ZYC/2013/FST.

References

[1] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Information Sciences*, vol. 297, no. 0, pp. 80–94, 2015.

[2] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A secret key cryptosystem by iterating a chaotic map," in *Advances in CryptologyEUROCRYPT91*. Springer, 1991, pp. 127–140.

[3] Y. Zhou, Z. Hua, C. M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–1, 2014.

[4] M. Franois, T. Grosges, D. Barchiesi, and R. Erra, "A new image encryption scheme based on a chaotic function," *Signal Processing: Image Communication*, no. 0.

[5] Y. Zhou, L. Bao, and C. L. P. Chen, "Image encryption using a new parametric switching chaotic system," *Signal Processing*, vol. 93, no. 11, pp. 3039–3052, 2013.

[6] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "Image encryption using 2d logistic-sine chaotic map," in *2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2014, pp. 3229–3234.

[7] A. Skrobek, "Cryptanalysis of chaotic stream cipher," *Physics Letters A*, vol. 363, no. 12, pp. 84–90.