Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Improving Security of an Image Encryption Algorithm based on Chaotic Circular Shift

Weihai Li
MOE-Microsoft Key Laboratory of Multimedia Computing and Communication
University of Science and Technology of China
Hefei, China
whli@ustc.edu.cn

Yuan Yuan
School of Engineering and Applied Science
Aston University
Birmingham B4 7ET, U.K.
yuany1@aston.ac.uk

***Abstract*—An image encryption algorithm based on chaotic circular bit shift is proposed recently. This paper analyses the security of this algorithm and point out that the key space is not as large as they alleged and the algorithm can not resist chosen-plaintext attack or difference attack. The amount of chosen-plaintexts to carry out an attack is very few. This paper also introduces two methods to improve its security by changing chaotic sequences generators, altering orders of permutation and substitution, and applying feedback link mode. The improved algorithm has variable key space and has very good avalanche effect to resist chosen-plaintext attacks, chosen-ciphertext attacks, or difference attacks. The computation cost of improved algorithm is very low.**

***Keywords*—image encryption, chaotic-based encryption, circular bit shift, security, attack**

I. INTRODUCTION

Nowadays, the security of digital images becomes more and more important since web attacks are more and more serious. Unlike common data, image data have large amount, high correlation and high redundancy so that they are usually not encrypted with normal data cipher standard. Since the 1990s, many image content encryption algorithms have been proposed [1-2]. Many among these are chaotic-based [3-5] algorithms, since researchers prefer chaotic systems' properties, such as aperiodicity, sensitivity to initial conditions and system parameters, etc.

Recently, an image encryption algorithm, abbreviate it as CCSE (chaotic circular shift encryption), was proposed in 2008 [6]. In this algorithm, four Logistic chaotic system are applied to generate four pseudorandom streams KS I, KS II, KSIII and KS IV. Thus, the key length is extended to four times of single Logistic chaotic system. Theoretical and experimental analysis showed that those binary and decimal key streams have good statistical property. Encrypted images have good statistical property to avoid statistical attacks.

However, the key space of the CCSE algorithm depends on system precision strictly. For a high precision computer system, the key space is large enough to resist brute force search attack; but if the system is applied in a low precision system, for example an 8 bits embedded system, the key space is too small. Besides, the algorithm is insecure against chosen-plaintext attack or against difference attack. A detailed cryptanalysis of CCSE algorithm is presented in section III.

In section IV, some improvements are proposed to enhance security of the CCSE algorithm. The improved CCSE algorithm has variable key space, which is independent of system precision, and has higher security.

II. INTRODUCTION TO CCSE ALGORITHM

A. Encryption

The encryption block diagram of the proposed CCSE algorithm is given in figure 1.

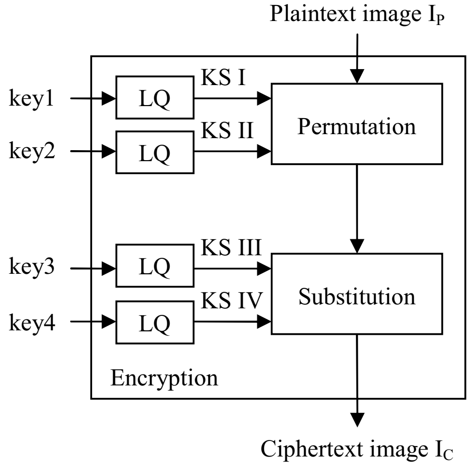


Figure 1. Encryption Block diagram of the CCSE algorithm.

From figure 1 we can see that there are two stages to encrypt an image: permutation and substitution. In the permutation stage, row permutated and column permutated are adopted according to two streams KS I and KS II, which are generated from Logistic chaotic systems with initial values key1 and key2 respectively. Here LQ module contains a Logistic system and a quantizer, which generates key stream by quantifying Logistic variable. Suppose the size of image is M×N, then row permutation map is a permutation of M numbers {1,2,…,M} described in KS I, and column

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

permutation map is a permutation of N numbers {1,2,…,N} described in KS II.

In the substitution stage, value of each pixel is circular bit shifted according to two streams KS III and KS IV, which are generated from Logistic chaotic systems with initial values key3 and key4 respectively. Stream KS III is a bit stream, which controls the direction of shift: 0 for left and 1 for right. Stream IV is a decimal number stream to control the times of shift. For an 8 bits gray-scale image, decimal numbers in stream IV are 0 to 7.

The LQ module contains a Logistic chaotic system and a quantizer. The Logistic chaotic system adopted in this algorithm is

$$x_{i+1} = \mu x_i \left(1 - x_i\right), \tag{1}$$

in which $\mu = 4$ and $0 < x_i < 1$.

The sequence $\{x_i\}$ is then quantified to bit stream $\{a_i\}$ by quantizer

$$a_i = \begin{cases} 1 & \left(x_i > 0.5\right) \\ 0 & \left(x_i < 0.5\right) \end{cases}, \tag{2}$$

and the decimal number stream can be generated from bit stream by grouping several bits.

B. Decryption

The decryption procedure is similar to the encryption procedure, but the permutation is replaced by inverse permutation and directions of circular bit shift are reversed.

C. The Secret Key

The secret key includes the four initial values key1 to key4 of Logistic chaotic systems in figure 1. For a 64 bits processor, the computing precision is 2^{64}, thus the key space is $(2^{64})^4=2^{256}$. It is large enough to avoid brute force search attack.

It is also proved that the quantified bit stream $\{a_i\}$ is Bernoulli sequence. Let the bit stream be a Markov chain, the next step transition probability matrix is

$$p = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}. \tag{3}$$

Then m (m > 0) step transition probability matrix is

$$p^m = p = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}. \tag{4}$$

So, each bit in $\{a_i\}$ is statistically independent and unpredictable. The good statistical property of bit stream satisfies security request.

D. Statistics of Ciphertext Image

Experimental results show that the ciphertext image is noise like, and the grayscale distribution of ciphertext image has good balance. Thus, the algorithm is good to resist statistical attack. Experimental result also demonstrates that the algorithm is very sensitive to key, so it is very hard to guess-and-test keys.

III. Cryptanalysis of the CCSE algorithm

A. Key Analysis

A chaotic system has a precision no more than 2^{-p} for a p-bits processor, so one Logistic chaotic system can only provides 2^p key space if parameter is fixed. Thus the total key space of the CCSE algorithm is $(2^p)^4=2^{4p}$. For a 64 bits processor, the key space is 2^{256}, which is large enough to avoid any brute force search attack. For a 32 bits processor, the key space is 2^{128}, which is good enough to avoid normal brute force search attack. But for 16 bits (or even 8 bits) processors, especially embedded micro processors, the key space is only 2^{64} (or 2^{32}), which is quite easy for modern personal computer to go through it.

Further more, it is quite difficult to determine initial values of a chaotic system from orbit. But when chaotic system is applied to generate cipher key stream, gaining the cipher key stream is equivalent to gaining the initial key. In the CCSE algorithm, if the key streams KS I to KS IV are broken, the system is broken too.

It should be noticed that KS III is used to control the directions of shift, and KS IV is used to control the times of shift. But to shifting r bits in any direction is equal to shifting (p-r) bits in counter direction. Clearly we can find a new key stream KS V to control the times of left circular bit shift, with equivalent effect of KS III and KS IV. Key stream KS V needs not to be an output of a Logistic chaotic system. Thus, cryptanalyst needs to break only KS I, KS II and KS V, as shown in figure 2.

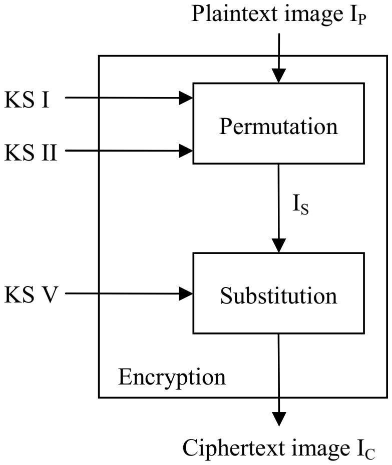


Figure 2. Encryption Block diagram of the CCSE algorithm.

B. Chosen Plaintext Attack

The following describes a chosen plaintext attack. Without losing generality, we discusses only grayscale image. The attack includes 3 steps, each breaks one key stream:

1) Breaking KS V:.Choose a plaintext image, in which all pixels have grayscale 1, and encrypt it. Since all pixels are equal, the first permutation stage is disabled. In the second substitution stage, grayscales of each pixels are circular bit shifted according to key stream KS III and KS IV, or accroding to key stream KS V. Compare the plaintext image and the ciphertext image, it is easy to know how each grayscales were shifted and thus KS V is obtained.

2) Breaking KS I: Choose a plaintext image I_P, in which all pixels in row i have grayscale i. Encrypt I_P with the first permutation stage to I_S (By encrypting I_P firstly and then decrypting ciphertext image with the second substitution stage, in which the KS V has been broken). Compare I_P and I_S, it is easy to determine the row permutation map KS I. If the height of image is larger than 256, then we need more plaintext images. Two proper chosen plaintext can solve images of height no more than 65536. For example, pixels in row i of first plaintext image are set to (i mod 256), and pixels in row i of second plaintext image are set to (i/256).

3) Breaking KS II: It is similar as breaking KS I, but all pixels in column i in plaintext is set to i. Compare I_P and I_S, the KS II can be obtained.

For a 256×256 image, this attack needs only 3 chosen plaintext to break all key streams. Obviously, the CCSE algorithm is insecure against chosen plaintext attack.

C. Difference Attack

It is noticed that the CCSE algorithm has no diffusion effect, so it is weak against difference attack [7].

Take two plaintext images with only one different pixel, and encrypt them. Pixels in same positions are permuted to same positions, and go through same shift process. It easy to know that there is only one different pixel between the two ciphertext images. Thus, we can determine one permutation item and one substitution item. After all items are determined, the encryption system is broken.

D. Lessons Learned

1) Key space should be independent of processors. Encryption algorithm should be flexible to be run on any processor without losing security. Classified security management also require variable key length. Thus the key should be designed dexterously.

2) Plaintext unrelated permutation should not be placed in the beginning or the end. Otherwise, chosen-plaintext attack or chosen-ciphertext attack can disable it easilly.

3) Nonlinear Diffusion must be implemented. To avoid chosen-plaintext attack and difference attack, each pixels in ciphertext should be influnced by as many as plaintext pixels. The relation between plaintext pixels, keys and ciphertext pixels should be nonlinear to avoid linear analysis attacks.

4) Algorithm should have good avalanche effect. Avalanche effect indicates that one bit change in plaintext (or in keys) will cause many bits change in ciphertext. Good avalanche effect means that 1/2 random bits in ciphertext are expected to change with one bit change in plaintext (or in keys). A strong secure encryption algorithm should have good avalanche effect to avoid differnce attacks or linear analysis attacks.

IV. Improved CCSE Algorithm

In this section, an improved CCSE algorithm is described.

A. Key Stream Generator

In this paper, a combined Logistic system is proposed in figure 3.

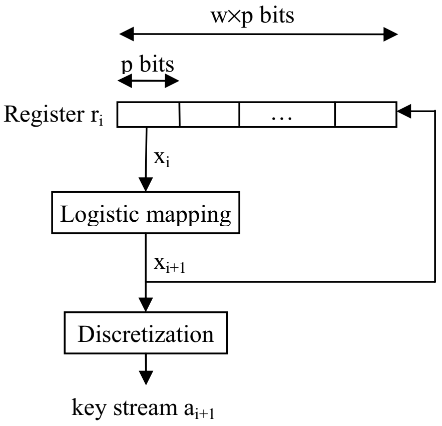


Figure 3. Key stream generator.

In figure 3, the register r_i stores w×p bits, p is the computing precision used in Logistic mapping, and w is a parameter used to control key space. Here, key is the initial register value r_0, which is shared between sender and receiver. The key length (or key space) is free to change by varying w. For example, if key length is chosen as 256 and computing precision is fixed to 32 bits, then w should be 256/32=8. The iteration rule is:

$$\begin{cases} x_i = r_i^p \times 2^{-p} \\ r_{i+1} = (r_i \ll p) \,|\, \lfloor x_{i+1} \times 2^p \rfloor \end{cases}, \tag{5}$$

where, r_i^p is the highest p bits of register r_i, "<<" is left bit shift operator, and "|" is or operator. The Logistic mapping is the same as formula (1) and the discretization rule is the same as formula (2). Obviously, sender and receiver get same key stream with their shared common key.

This key stream generator is equivalent to a combination of w LQs with their outputs are interlaced. LQ is the same as defined in figure 1.

B. Encryption

There are two kinds of procedures in our improved version to enhance security:

1) Permutation: This procedure contains three steps: row permutation, column permutation, and pixel exchange. The row permutation and column permutation are the same as above, except that KS I and KS II are picked from a new key stream. Pixel exchange is to ensure the bottom-right pixel in original image is place in top-left. Suppose the first two steps permutated the bottom-right pixel to position (x,y), then exchange pixels (x,y) and (1,1).

2) Substitution: In this procedure, the value of pixels are circular bit shifted according to key stream as what was done in original algorithm, except that KS III is removed and shift is fixed on left direction. To implement diffusion principle, a CBC (cipher block chain) mode is adopted here. After each pixel is substituted, ciphertext is fed back to modify register r_i. Suppose feedback ciphertext is c (8 bits grayscale), the register is modified according to

$$r_i \leftarrow \left[r_i \oplus \left(c >> 4 \right) \right] <<< \left(c \,\&\, 0x0F \right). \qquad (6)$$

In formula (5), "⊕" is exclusive or operator, ">>" is right shift operator, "<<<" is circular bit shift operator, and "&" is and operator. For color image, the r,g,b channels are fed back one by one.

Thus, the improved algorithm is described in figure 4.

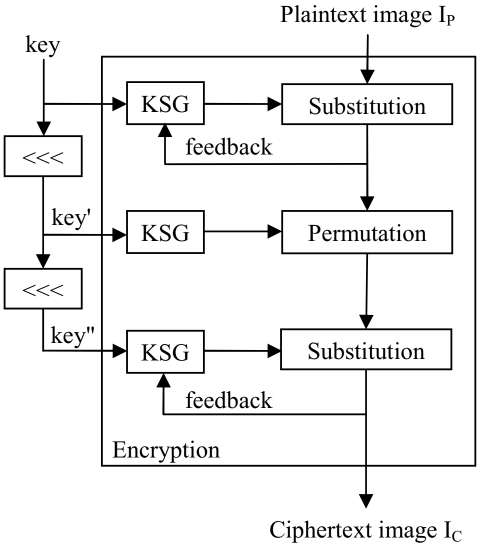


Figure 4. Improved CCSE algorithm.

In the first substitution stage, each pixel will influence all following pixels. After the second permutation stage, the final pixel is moved to the first place. Then in the third substitution, this pixel will influence all pixels. Thus, each ciphertext pixel is influenced by all plaintext pixels.

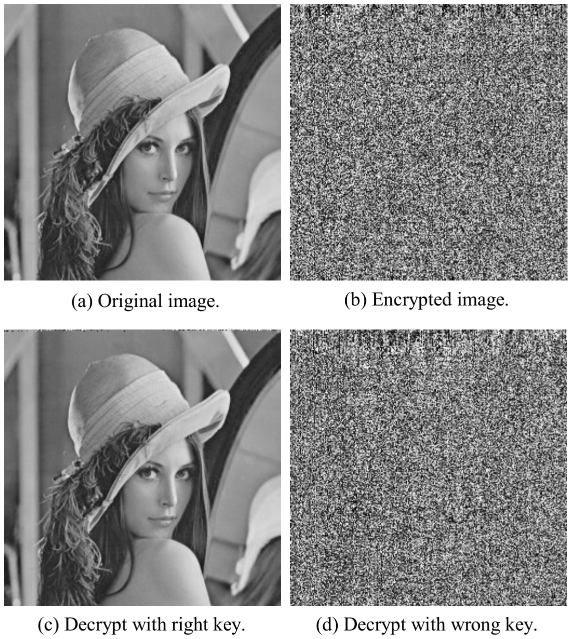
The Computation cost of the improved algorithm is very low. There are only two circular bit shift operations and one permutation operation for each pixel.

C. Decrypiton

The decryption algorithm is just the reverse of encryption one. Note that the three steps in permutation stage should also be reversed.

D. Security Analysis

Figure 5 shown a plaintext image and encrypted ciphertext image. Ciphertext image can be correctly decrypted with the right key, but totally wrong with an error key. In this experiment, the key length is 256 bits, computing precision is set to 32 bits and w=8. The encryption key is k_1="00000012 00000013 00000014 00000015 00000016 00000017 00000018 00000019", and the error key used in (d) is k_2="00000012 00000013 00000014 00000015 00000016 00000017 00000018 0000001a" in hex. It can be seen that the algorithm is very sensitive to key.



(a) Original image. (b) Encrypted image.

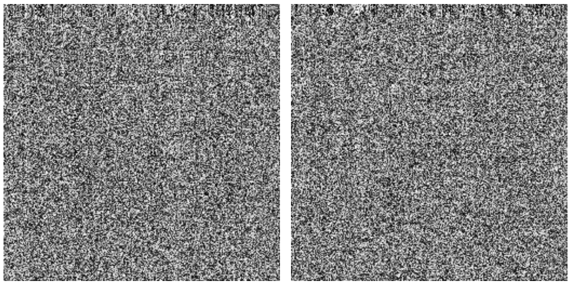
(c) Decrypt with right key. (d) Decrypt with wrong key.

Figure 5. Encryption and decryption wth improved CCSE algorithm.

From figure 4 we can see that the input and output of the permutation module are both protected by former and later substitution respectively. Neither chosen plaintext attack nor chosen ciphertext attack can deduce the input or output of permutation, so the permutation rules are secure. On the other side, permutation and feedback mechanisms ensure that each original pixel influences more encrypted pixels, which protects sustituion module from chosen text attack.

The avalanche effect to key of the original CCSE algorithm is good since all pixels are substituted with chaotic system. But the avalanche effect to plaintext is very weak since one original pixel affects only one result pixel. That is also why the original algorithm cannot resist chosen plaintext attack.

To test the avalanche effect of the improved CCSE algorithm, another two experiments are shown in figure 6. In the first experiment, image 5(a) is encrypted with another key k_3="00000012 00000013 00000014 00000015 00000016 00000017 00000018 00000018", and the encrypted image is given in figure 6(a). In the second experiment, image 5(a) is modified with the grayscale of (1,1) pixel is changed from 160 to 161, and then encrypted with key k_1. The encrypted image is given in figure 6(b). It can be seen from image 6 that although there is only 1 bit different in key or in original image, the encrypted images are greatly different.



(a) Slightly different key. (b) Slightly different original image.

Figure 6. Avalanche effect test.

NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) are two criterions to examine the avalanche effect [8]. The NPCR and UACI are defined as:

$$\mathrm{NPCR}=\frac{\sum_{i,j}D(i,j)}{W\times H}\times 100\%\,, \tag{7}$$

$$D(i,j)=\begin{cases}1 & C_1(i,j)=C_2(i,j)\\ 0 & C_1(i,j)\neq C_2(i,j)\end{cases}\text{, and} \tag{8}$$

$$\mathrm{UACI}=\frac{1}{W\times H}\left[\sum_{i,j}\frac{\left|C_1(i,j)-C_2(i,j)\right|}{255}\right]\times 100\%\,. \tag{9}$$

Here, W and H are the width and height of image, C_1 and C_2 are two ciphertext images encrypted from two plain images with just one different pixel or from two key with just one different bit.

Appling these two criterions between figure 5(b) and figure 6(a), we got NPCR=1.5% and UACI=30.9%. Appling these two criterions between figure 5(b) and figure 6(b), we got NPCR=19.5% and UACI=23.9%. Then it can be concluded from these results that most pixels had been changed when a bit changed in key or in plaintext image.

To estimate avalanche effect more directly, a NBCR can be defined as

$$\mathrm{NBCR}=\frac{\sum_{i,j,k}D(i,j)_k}{W\times H\times B}\times 100\%\text{, and} \tag{10}$$

$$D(i,j)_k=\begin{cases}1 & C_1(i,j)_k=C_2(i,j)_k\\ 0 & C_1(i,j)_k\neq C_2(i,j)_k\end{cases}, \tag{11}$$

where B is the depth of image. For a grayscale image, B=8. D is defined as whether a bit changed or not. $C(i,j)_k$ refer the kth bit of pixel C(i,j).

Compute NBCR on the above two experiments, we got NBCR=49.31% between figure 5(b) and figure 6(a), and NBCR=42.49% between figure 5(b) and figure 6(b). Obviously, the improved CCSE algorithm has very good avalanche effect, nearly half bits changed in ciphertext when only one bit changed in plaintext or key.

The improved CCSE algorithm implements diffusion and confusion principles adequately. Each pixel bit and each key bit influence all ciphertext bits, and the circular bit shift and the ciphertext feedback mechanism result in strong nonlinear operation. This strengthens the improved CCSE algorithm further from chosen-plaintext attack, chosen-ciphertext attack, or difference attack.

V. Conclusion

This paper analyses the security of a recently proposed image encryption algorithm, called CCSE algorithm. This algorithm based on chaotic circular bit shift. Although the algorithm is sensitive to key, and generated key streams have good statistical distribution, it can not resist chosen-plaintext attack or difference attack. Also the key space is not stable. Some attack examples are introduced.

An improved CCSE algorithm is also proposed in this paper. This algorithm has flexible-controlled variable key space, and very good avalanche effect. By implementing diffusion and confusion principles, the improved algorithm is secure against chosen-plaintext attack, chosen-ciphertext attack, and difference attack. The computation cost for each pixel is only two circular bit shifts and one permutation.

References

[1] P. Dang and P.M. Chau, “Image Encryption for Secure Internet Multimedia Applications”, IEEE Transactions on Consumer Electronics, vol.46, no.3, pp.395-403, Aug. 2000.

[2] W.H. Li and Y. Yuan, “Visual information encryption in frequency domain: risk and enhancement”, Lecture Notes in Computer Science, vol.4153, pp.225-234, Aug. 2006.

[3] L.H. Zhang, X.F. Liaom and X.B. Wang, “An image encryption approach based on chaotic maps”, Chaos, Solitons and Fractals, vol.24, pp.759-765, May 2005.

[4] S.J. Li, C.Q. Li, G.R. Chen, and K.T. Lo, “Cryptanalysis of RCES/RSES image encryption scheme”, Journal of systems and Software, vol.81, no.7, pp.1130-1143, July 2008.

[5] S.J. Xu, Y.L. Wang, J.Z. Wang, and M. Tian, “Cryptanalysis of two chaotic image encryption schemes based on permutation and XOR operations”, 2008 International Conference on Computational Intelligence and Security, vol.2, pp.433-437, Dec. 2008.

[6] C. Fu, and Z.L. Zhu, “A chaotic image encryption scheme based on circular bit shift method”, Proceedings of the 9th International Conference for Young Computer Scientists, pp.3057-3061, Nov. 2008;

[7] C.E. Shannon, “Communication theory of secrecy systems”, Bell System Technical Journal, vol.28-4, pp.656-715, Oct. 1949.

[8] G.R. Chen, Y.B. Mao, and C.K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps”, Chaos, Solitons and Fractals, vol.21, pp.749-761, July 2004.