

Digital Color Image Encryption Using RC4 Stream Cipher and Chaotic Logistic Map

Ni G. A. P. Harry Saptarini*, Yosua Alberth Sir**

* Informatics Management, Department of Electrical Engineering, Politeknik Negeri Bali

** Department of Computer Science, Faculty of Science and Engineering, Universitas Nusa Cendana

Keywords:

Digital color image encryption
RC4 stream cipher
Chaotic logistic map

ABSTRACT

Doing a digital image transmission over internet need a secure protection against illegal copying. Unfortunately, many current data encryption methods such as DES, RSA, AES, and other only suitable for text data, but not for digital image. In this paper, we propose new secure algorithm for image encryption, which based on RC4 stream cipher algorithm and chaotic logistic map. The proposed algorithm works by converting the key to initial value, and then this value is used as an input to chaotic logistic map function to generate a sequence of pseudo random numbers or 256-byte key array. The final step is a permutation process and the result will be XORed with the bytes stream of plain-image to produce cipher-image or XORed with the cipher-image to produce plain-image. The experiment results show that our proposed algorithm: (i) able to make the cipher-image cannot be visually identified, (ii) can eliminate the statistical correlation between the plain-image and cipher-image, (iii) very sensitive to small changes of key, (iv) has no change in image contents (lossless encryption) during encryption or decryption process.

*Copyright © 2013 Information Systems International Conference.
All rights reserved.*

Corresponding Author:

Ni G. A. P. Harry Saptarini,
Informatics Management, Department of Electrical Engineering,
Politeknik Negeri Bali,
Kampus Bukit Jimbaran, Badung, Bali , Indonesia.
Email: ayu_harry@yahoo.com

1. INTRODUCTION

Image encryption is one of the secure methods to protect digital color images against illegally copying when transmitted over unsecure channel. Unfortunately, according to [1,2], many popular encryption methods such as DES, RSA, AES, and others only work well for plaintext but not for digital color images. In most of the natural digital images, the values of the neighboring pixels are strongly correlated (i.e. the value of any given pixel can be reasonably predicted from the values of its neighbors). This unique characteristic lead to huge changes of each pixel of plain-image is not going to drastically reduce the quality of the cipher-image which will make the content of cipher-image can still be visually identified by human. Due to some desirable properties such as pseudo-random behavior and very sensitive to a small change in the initial value [3], the chaotic system (i.e. chaotic logistic map) has become popular method for highly secure image encryption. Many attempts have been made in the past to encrypt digital images using chaotic logistic map or CLM for short (in the rest of the paper, we will use CLM and chaotic logistic map interchangeably). In [4], they use 2D cat map and in [1], they use two chaotic logistic map functions, the first CLM is used to generate 24 real numbers and then these numbers is converted to integer form. The 24 integer numbers are used to generate initial value $\{x_p\}$ for the second CLM, which is used to perform image encryption process. Different approaches tend to do in [5], where they use a combination of CLM function and genetic algorithm to encrypt the digital color image. CLM is used to generate 4 chaotic sequences which are then converted to 4 key streams. The generated key streams are used to control the process of crossover and mutation.

In this paper, we proposed a digital color image encryption using the combination of RC4 stream cipher and chaotic logistic map function. We use this combination for the following reasons: (i) the simplicity of RC4 algorithm, (ii) RC4 requires only byte-length manipulations so it is suitable for embedded systems, (iii) even though RC4 has vulnerabilities [6], we combined with chaotic systems to make it almost impossible to break. We adopt the key generator used by [1] to convert key to initial value, then use this initial value on CLM function to generate pseudo random number sequence. The pseudo random numbers will be combined with the byte streams of plain-image using exclusive-or (XOR) operation when doing an encryption process (or XOR the byte streams of cipher-image with a stream of pseudo random numbers when doing a decryption process).

The rest of the paper is organized as follows. In Section 2 we will present chaotic function and proposed image encryption method, in Section 3 we report the experimental results, and finally some conclusions in Section 4.

2. RESEARCH METHOD

2.1. Chaotic Logistic Map Function

Chaos is phenomena that exist in nonlinear systems, in which seemingly random events are actually predictable from simple deterministic equations [2]. One of the important properties of chaos is extreme sensitivity to initial conditions. Chaotic logistic map is one of the popular chaotic systems. Consider a CLM function as shown in Equation (1).

$$X_{n+1} = \lambda X_n (1 - X_n) \quad (1)$$

where λ is a control parameter on the interval $\lambda = [0,4]$ and X_n is real number on the interval $X_n = [0,1]$. This system is said to be chaotic if λ has a value on the interval $\lambda = [3.569955672,4]$. In this paper, we use $\lambda = 4$ so the complete formula is shown in Equation (2).

$$X_{n+1} = 4X_n (1 - X_n) \quad (2)$$

2.2. Proposed Image Encryption Algorithm

The structure of our proposed encryption method (as shown in Figure 1) consists of three main units: (i) converter unit, (ii) CLM function unit, and (iii) RC4 stream cipher unit. The converter unit will convert key to initial value X_0 using Equation (3), (4), (5), (6), (7) and (8). The output of converter unit is an initial value X_0 which will be used by CLM function unit to generate 256-bytes of array $U[i]$ or also known as key array. The last step is RC4 stream cipher process where the content of array $S[i]$ and array $U[i]$ is swapped between each other then the result will be XORed with the byte streams of plain-image to produce cipher-image or XORed with the cipher-image to produce plain-image.

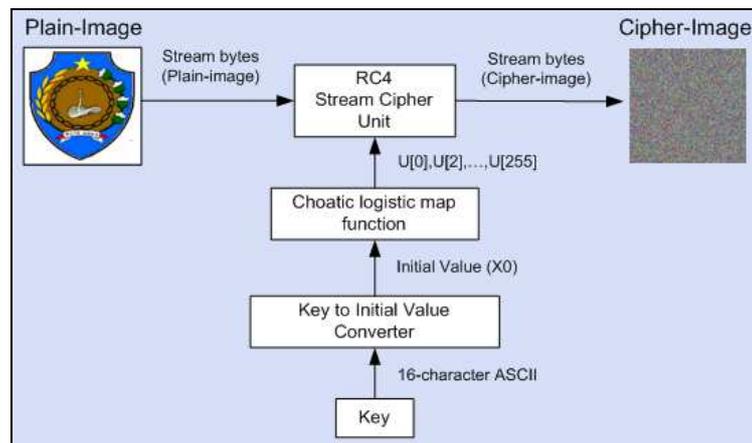


Figure 1. The structure of the proposed encryption method

In our proposed encryption method, we do not encrypt the header part of image (*.BMP) but only the RGB pixel part. Our proposed encryption method consists of three steps. The entire encryption method is described as follows:

Step (1). Convert key to initial value.

The key has 16 ASCII characters in length where each character of K_i consist of 8-bit.

$$K = K_1, K_2, \dots, K_{16} \text{ (ASCII code)} \quad (3)$$

For each K_i value, we convert them to bit stream B_0 and B_1 .

$$B_0 = K_{11} \dots K_{18} K_{21} \dots K_{28} K_{31} \dots K_{38} K_{41} \dots K_{48} K_{51} \dots K_{58} K_{61} \dots K_{68} K_{71} \dots K_{78} K_{81} \dots K_{88} \quad (4)$$

$$B_1 = K_{91} \dots K_{98} K_{101} \dots K_{108} K_{111} \dots K_{118} K_{121} \dots K_{128} K_{131} \dots K_{138} K_{141} \dots K_{148} K_{151} \dots K_{158} K_{161} \dots K_{168} \quad (5)$$

where each K_{ij} from Equation (4) and (5) has binary representation (0 or 1), where i refers to character position ($i=1,2,\dots,16$) and j refers to bit position of character ($j=1,2,\dots,8$). Using binary representation of K_{ij} value, the real number X_{B_0} and X_{B_1} will be counted.

$$X_{B_0} = (K_{11} \times 2^0 + \dots + K_{18} \times 2^7 + K_{21} \times 2^8 + \dots + K_{28} \times 2^{15} + K_{31} \times 2^{16} + \dots + K_{38} \times 2^{23} + K_{41} \times 2^{24} + \dots + K_{48} \times 2^{31} + K_{51} \times 2^{32} + \dots + K_{58} \times 2^{39} + K_{61} \times 2^{40} + \dots + K_{68} \times 2^{47} + K_{71} \times 2^{48} + \dots + K_{78} \times 2^{55} + K_{81} \times 2^{56} + \dots + K_{88} \times 2^{63}) / 2^{64} \quad (6)$$

$$X_{B_1} = (K_{91} \times 2^0 + \dots + K_{98} \times 2^7 + K_{101} \times 2^8 + \dots + K_{108} \times 2^{15} + K_{111} \times 2^{16} + \dots + K_{118} \times 2^{23} + K_{121} \times 2^{24} + \dots + K_{128} \times 2^{31} + K_{131} \times 2^{32} + \dots + K_{138} \times 2^{39} + K_{141} \times 2^{40} + \dots + K_{148} \times 2^{47} + K_{151} \times 2^{48} + \dots + K_{158} \times 2^{55} + K_{161} \times 2^{56} + \dots + K_{168} \times 2^{63}) / 2^{64} \quad (7)$$

Next step, real number X_{B_0} in Equation (6) and X_{B_1} in Equation (7) is used to create initial value X_0 . The

complete formula for creating initial value is shown in Equation (8).

$$X_0 = (X_{B_0} + X_{B_1}) \bmod 1 \quad (8)$$

Step (2). Generate a key array (pseudo random number sequence) using chaotic logistic map function.

The initial value X_0 in Equation (8) will be used by CLM function to generate a key array of pseudo

random number sequence by using the formula in Equation (2). Generally, the chaotic process uses initial value X_0 to get X_1 value, then X_1 value will be used to get X_2 value, and so on. In order to strengthen

CLM against any statistical attacks, we generate X_n value after a certain number of iterations. We

determine the number of iterations by taking two digits after decimal point. For example, the initial value X_0 is 0.937696878979928 then the number of iterations required to get the first value of chaos X_1 is 93,

thus after 93^{rd} iterations, X_1 value will be 0.8080204084200282. We can say that the value of X_n which

obtained at the end of iteration will act as a new " X_n " to calculate X_{n+1} and so on. After each X_n value is

obtained, it will be converted to integer form by taking eight points started after the decimal point of real numbers. For example, assuming that the value of X_n is 0.8080204084200282. After converting this value

to integer form will yield 80802040, and then it will be modulo 256. The result value will be stored in array $U[i]$ where $i = 0, 1, \dots, 255$. This process will be repeated until $U[255]$ is filled.

Step (3). RC4 streams cipher and encryption/decryption process for RGB channel.

The output of Step (2) is an array $U[i]$ which also called "key streams" and consists of 256 pseudo

random numbers. Array S is created (as shown in Figure 2a.) where the content of array S are set equal to the values from 0 through 255 in ascending order; which is

$S[0] = 0, S[1] = 1, \dots, S[245] = 254, S[255] = 255$. Next step, array U is used to produce the initial

permutation of array S (Figure 2b.). For each $S[i]$, swap $S[i]$ with another byte in S according to the content of $U[i]$ and this will cause the content of S still contains all the numbers from 0 through 255. In Figure 2a., streams generation is done by swapping $S[i]$ with another byte in S according to a scheme dictated by the current configuration of S . The encryption/decryption process for each RGB channel is done by XORed each pixel's of RGB component of plain-image with the bytes of array S (or XORed the bytes of cipher-image with the bytes of array S when do a decryption process). The decryption algorithm is identical to the encryption algorithm discussed above except that the order of the basic operations is reversed. The result of encryption process can be seen in Figure 3.a and decryption process in Figure 3.b.

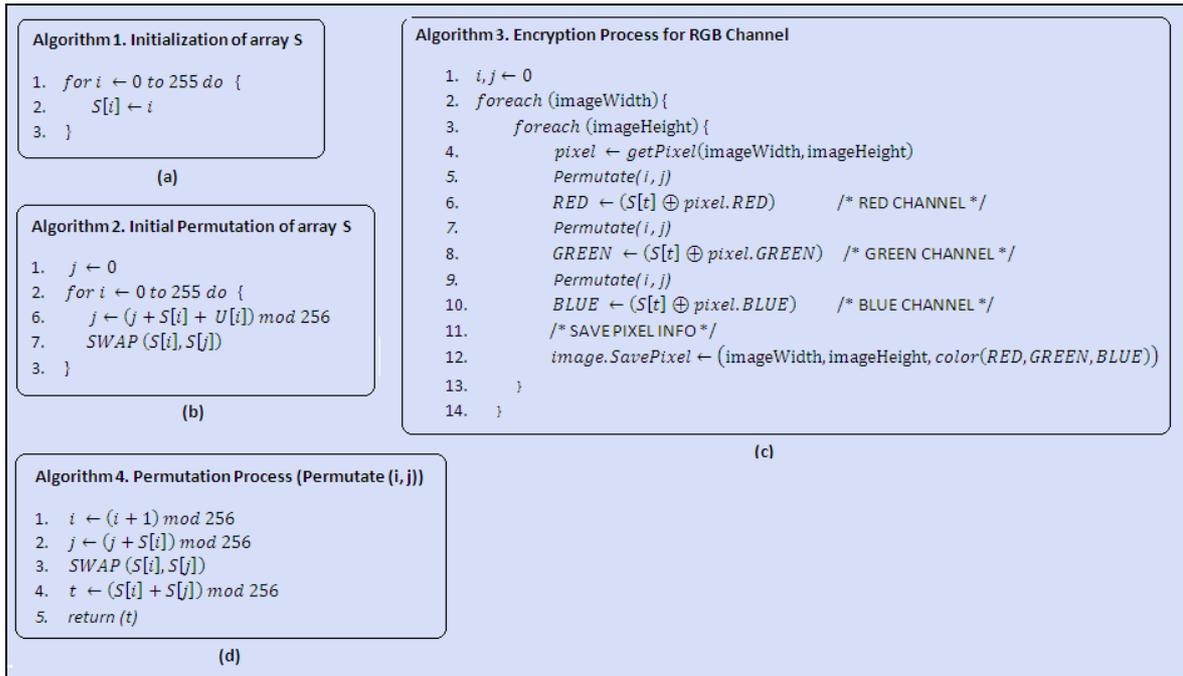


Figure 2. The algorithm of RC4 streams cipher. (a) Initialization of array S , (b) Initial permutation of array S , (c) Encryption/Decryption process for each RGB channel, (d) Permutation process.

3. RESULTS AND ANALYSIS

To empirically assess the performance of our proposed method, we have carried out a number of experiments. For this purpose, a program was developed for C# (2005) by the authors. These experiments include: (i) encryption and decryption process, (ii) histogram analysis of plain-image and cipher-image, and (iii) key sensitivity analysis.

3.1. Encryption and Decryption Process

We use a 24-bit color image of size 270 x 272 and $k = 4$ for CLM function. This image is encrypted using secret key "abcdefghijklmnopq". The visual inspection of Figure 3.a shows that the encrypted image (cipher-image) region is totally invisible for human and Figure 3.b shows that by using the same secret key, cipher-image could be converted back to plain-image. This result shows that our approach works well in both encryption and decryption process.

3.2. Security Analysis

In this section, we discuss the security analysis of the proposed image encryption algorithm such as histogram analysis and key sensitivity analysis with respect to the plain-image and key to prove that the proposed method is robust against the statistical and brute force attack.

3.2.1. Histogram Analysis

Ideally, the histogram of plain-image and cipher-image should not have statistical relationship between each other. Based on the histogram analysis in Figure 4 and Figure 5, we can see that the histogram

of each RGB channel is uniform. The uniform distribution of cipher-image histogram is an good indication that cipher is robust against statistical and brute force attack [7]. The results of histogram analysis also shows that there is no statistical relationship between plain-image in Figure 4 and cipher-image in Figure 5.

3.2.2. Key Sensitivity Analysis

We have carried out a key sensitivity test using a key that is one digit different from the original key to decrypt the encrypted image. We have encrypted plain-image using key "abcdefghijklmnopq" and then decrypted the cipher-image using: (i) wrong key "abcdefghijklmnop", and (ii) correct key "abcdefghijklmnopq". The resulting image is totally different from the original image as shown in Figure 6. This test demonstrates that the proposed algorithm is very sensitive to any change in the secret key value.

4. CONCLUSIONS

We have proposed a color image encryption method which based on RC4 stream cipher and chaotic logistic map. Experimental results show that our method can be used as an alternative method to encrypt digital images because this method: (i) can encrypt image in such way so that cipher-image can not be visually identified by human, (ii) eliminates statistical relation between plain-image and cipher-image (histogram of cipher-image has a uniform distribution), (iii) is very sensitive to any changes in key value.

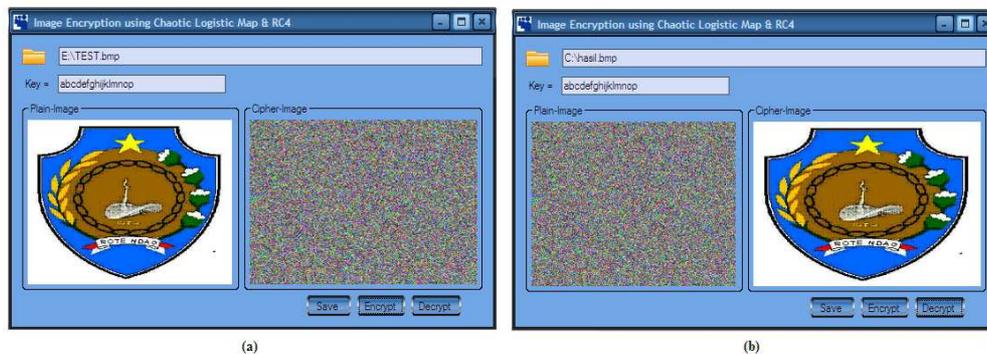


Figure 3. Image encryption/decryption results. (a) shows image encryption result where plain-image (left side) and cipher-image (right side), using key "abcdefghijklmnopq". (b) shows image decryption result, cipher-image (left side) and plain-image (right side), using key "abcdefghijklmnopq".

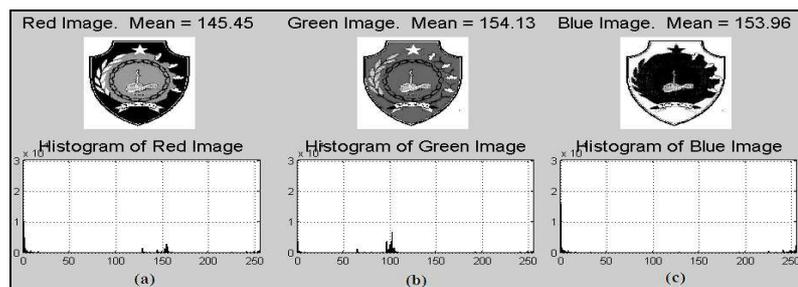


Figure 4. Histogram analysis of plain-image. (a) histogram of red channel, (b) histogram of green channel, (c) histogram of blue channel.

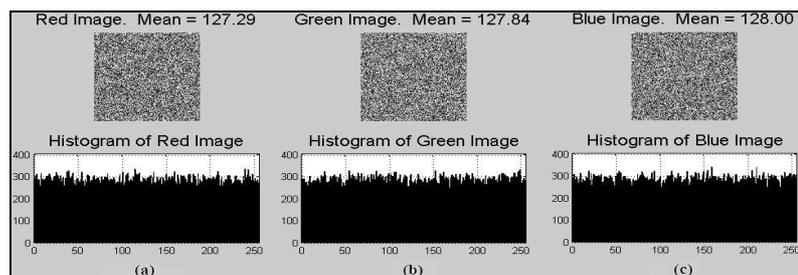


Figure 5. Histogram analysis of cipher-image using key "abcdefghijklmnopq". (a) histogram of red channel, (b) histogram of green channel, (c) histogram of blue channel.



Figure 6. Key sensitivity analysis. (a) encrypted image (cipher-image) using key "abcdefghijklmnopq", (b) decrypted image using correct key "abcdefghijklmnopq", (c) decrypted image using wrong key "abcdefghijklmnopr".

REFERENCES

- [1] N.K. Pareek, *et al.*, "Image Encryption using Chaotic Logistic Map," in *Image and Vision Computing*, Volume 24, pp. 926–934, 2006.
- [2] Y. Mao and G. Chen, "Chaos-based image encryption," in Eduardo Bayro-Corrochano, editor, *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neural Computing and Robotics*. Springer-Verlag, Heidelberg, April 2004.
- [3] B. Schneier, "*Applied Cryptography: Protocols, Algorithms, and Source Code in C*," (2nd edn.) Wiley, New York, 1996.
- [4] M. Ahmad and M.S. Alam, "A New Algorithm of Encryption and Decryption of Images Using Chaotic Mapping," in *Prosiding of International Journal on Computer Science and Engineering*, IJCSE 2009, Volume 2(1), pp. 46-50, 2009.
- [5] E.S. El-Alfy and K. Al-Utaibi, "An Encryption Scheme for Color Images Based on Chaotic Maps and Genetic Operators," *The Seventh International Conference on Networking and Services*, ICNS 2011, pp. 92-97, 2011.
- [6] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proceedings of the second ACM conference on Wireless network security*, Zurich, pp. 79-86, 2009.
- [7] A. Jolfaei and A.R. Mirghadri, "An Image Encryption Approach Using Chaos and Stream Cipher," in *Journal of Theoretical and Applied Information Technology*, Volume 12 No 2, pp 117-125, 2010.

BIBLIOGRAPHY OF AUTHORS

	<p>Ni G.A.P. Harry Saptarini received S.Kom (Informatics Management) from STIKOM Surabaya and M.Cs (Computer Science) from Gadjah Mada University. She is currently a lecture in Informatics Management, Department of Electrical Engineering, Politeknik Negeri Bali. Her research interests include cryptography, text mining, and artificial intelligent.</p>
	<p>Yosua Alberth Sir received the ST (Electrical) from University of Indonesia and M.Cs (Computer Science) from Gadjah Mada University. He is currently a lecturer in Computer Science Department, University of Nusa Cendana, Kupang. His research interests include cryptography, text mining, and plagiarism detection.</p>