Chaotic Encryption Technique for Color Images by Coupling Two Chaotic Maps

Asmita, Shrikant Lade

Asmita, M-Tech IV Semester, IT, RKDF, Bhopal
Shrikant Lade, H.O.D. (I.T.), RKDF, Bhopal

Abstract
With the increasing demand of providing security for images/videos with private information, chaos-based cryptosystems have played an important role in image encryption because of their excellent random properties and encryption performance. However, existing chaos-based systems have the security defect due to small key space or other security weakness. This paper introduces an efficient chaos-based stream cipher, composing two chaotic maps (Logistic map and Standard map) and a large enough external secret key for image encryption. The external secret key is used to derive the initial conditions for the chaotic maps, and is employed with the two chaotic maps to confuse the relationship between the cipher image and the plain image. The proposed chaotic system shows excellent chaotic behaviours encryption performance, high sensitivity to the security keys, and a sufficiently large key space to resist the brute attack.

Keywords
Chaos, Chaotic Sequence, Encryption, Key Space, Logistic map, Standard map

1. INTRODUCTION

With the fast development of image transmission through computer networks especially the Internet, medical imaging and military message communication, the security of digital images has become a most important concern. Image encryption, is urgently needed but it is a challenging task because it is quite different from text encryption due to some intrinsic properties of images such as huge data capacity and high redundancy, which are generally difficult to handle by using conventional techniques. Nevertheless, many new image encryption schemes have been suggested in current years, among which the chaos-based approach appears to be a hopeful direction.

2. EXISTING MODELS

In general, the confidentiality of multimedia data such as digital image and video can be safeguarded by means of private-key cryptography. Some encryption techniques particularly dedicated to image indeed form the basis for video encryption.

A. Selective Bitplane Encryption

To achieve a fast encryption, image encryption schemes are often designed not to encrypt the entire images completely, but a portion only. In this way, the amount of computation is reduced and this approach is regarded as selective image encryption [1]. Gray level images are usually composed of eight bitplanes. The higher-order bitplanes contain the majority of visually significant and strong correlation data of the plain image, whereas the remaining contributes to more subtle details in the image. Based on this observation, a selective bitplane encryption scheme is proposed [2]. AES is selected as the functional encryption in this scheme. Undoubtedly, the underlying security is subject to the portions of bitplane to be encrypted. However, there seems to be no convincing method to determine the portions of bitplane encryption for encryption

B. SCAN-based Image Encryption

A formal language (SCAN) is intended to describe and generate multiple of two-dimensional (2D) spatial accessing order from a short set of simple ones [3]. It is first employed for image encryption in [4]. The plain image is initially serialized to one dimensional data stream which is then described by the SCAN language. The SCAN string is served as an encryption key bound to a given 2D image array. The encryption procedure is to rearrange image into a final sequential representation. Each assembled secret image in process of SCAN string is combined by the insertion of additive noises at particular image points. Since no one except the intended user can obtain the correct SCAN combinations, the original image is therefore considered confidential.

C. Embedding Image Compression into Encryption

The abovementioned schemes are devoted to the uncompressed image data. For compressed images, some

Manuscript received October 5, 2015
Manuscript revised October 20, 2015

special measures are required before strictly combining encryption and compression directly. In, [5] a framework is proposed for fast encryption by entropy encoders such as Huffman coder. In entropy coding, the statistical model is used to decode the compressed bit stream. It is therefore suggested that multiple statistical models are used alternately in certain secret order to encode the input symbol stream. Through security analyses, the proposed scheme is proved to be applied effectively on both multiple Huffman coding tables of Huffman coder and multiple state indices of QM coder. There is also a concern about codec dependence of such kind of scheme [6]. Nevertheless, the potential for integrating encryption with multimedia compression at a low computation is promised.

D. Chaotic Image Encryption

Recently, a widely studied example of image encryption is based on chaos theory which is well established, simple but with complicated dynamics. In, [7] a symmetric encryption scheme based on two-dimensional chaotic maps is proposed. A two or higher dimensional discretized chaotic map is adopted for pixel permutation together with another one-dimensional (1-D) map for diffusion.

3. Proposed Methodology

The following proposed cryptosystem is a symmetric key steam cipher algorithm, which utilizes the essence of chaos, i.e., sensitivity on the initial condition as well as on system parameter. This cryptosystem does not use explicitly the system parameter or initial condition of the chaotic map (logistic map) as a secret key. However, these parameters are generated by an external secret key. The cryptosystem is further made robust against any reasonable attack by using the feedback technique, i.e., encryption of each pixel of the plain image is also made dependent upon the encryption properties of the previous pixel of the plain image [17]. Further, new features of the proposed stream cipher include the heavy use of data-dependent inputs, data-dependent iterations (variable number of iterations for each map depending on the current value of the key, the value of the previous cipher pixel and the output of the logistic map). The proposed procedure consists of three steps: choosing chaotic maps, permutation and diffusion.

A. Choosing Chaotic Maps

We have chosen three different chaotic maps for each color plane of the color image (i.e. red, green and blue).

Logistic map: The Logistic map defines one of the simplest forms of a chaotic process [9]. It is defined by the following equation:

$$x(n)= \lambda*x(n-1)*[1-x(n-1)]$$

Where x(n) is a state variable, which lies in the interval [0,1] and λ is the control parameter and belongs to interval(0; 4]. λ should be greater than the accumulation point 3.569945672 in order to maintain the highly chaotic state.

2D Standard Map: It adopts an invertible discretized 2D standard map [8, 10] with the introduction of random scan couple (rx, ry) for corner pixel confusion,

$$x_{k+1} = (x_k + y_k + r_x + r_y) \bmod N,$$

$$y_{k+1} = \left(y_k + r_y + K_C \sin \frac{2\pi x_{k+1}}{N} \right) \bmod N,$$

where (xk, yk) and (xk+1, yk+1) is the original and the permuted pixel position of an N × N image, respectively. The standard map parameter KC is a positive integer.

B. Permutation:

After generating random sequence with above mentioned chaotic maps, a random sequence is generated for each color plane. On the basis of this sequence pixels are shuffled row wise and column wise.

- Two sequences are obtained for each color plane, one for shuffling row having length equal to numbers of rows, and one for shuffling column having length equal to number of columns.
- For a particular row sum of all the elements is calculated and then its mode with 2 is calculated if it is 1, row shifted left, otherwise row is shifted right as in [11].
- With respect to value present in the row sequence the corresponding row pixels are circularly shifted.
- Same process is repeated for columns.

The image obtained after various rounds of the above process is a permuted image with shuffled pixels. But histogram of this permuted image will remain same as that of original image, so some mechanism to change pixel values must be implemented to make it more secure.

C. Diffusion:

After obtaining a permuted image the diffusion stage, each pixel of the 2D permuted image is scanned in sequential manner, which usually starts from the upper left corner. The diffusion effect in this stage is realized by XOR-and-then-shift method:

$$bs \leftarrow Ci_{-1} \;\&\; 7 \quad \ldots(1)$$

$$Pi \leftarrow Pi \wedge Ci_{-1} \quad \ldots.(2)$$

$$Ci \leftarrow (Pi >> bs) \mid (Pi << (8 - bs)) \quad \ldots(3)$$

where Pi is the value of the ith pixel of the permuted image, Ci-1 and Ci is the value of the (i-1)th and the ith pixel of the diffused image, respectively. The seed of the diffusion function is C-1 which is obtained from the diffusion key KD. The new pixel value is obtained by Exclusive-OR (XOR) the current pixel value Pi of the permuted image with the previously diffuse pixel Ci-1. Then it is bitwise rotated circularly with value bs obtained by bitwise anding Ci-1 with 7. As the previous diffused pixel will influence the current one, a tiny change in the original image is reflected in more than one pixel in the cipher image and hence the diffusion effect is introduced in this stage.

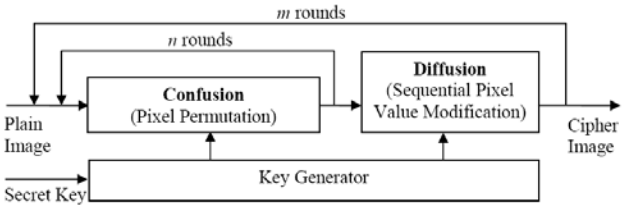


Fig (1) Flow chart of encryption process

4. Encryption Algorithm

An M×N color image P is taken as input along with a 12 digit decimal encryption key K. Output of this procedure will an encrypted image of size m×n

Step1: Get the parameter settings and initial values of logistic map from the encryption key K.

$\mathbf{K = [K_{10}\ K_9\ K_8\ K_7\ K_6\ K_5\ K_4\ K_3\ K_2\ K_1]}$.

$\mathbf{r(1) = K_4\ K_3\ K_2\ K_1}$.

$\mathbf{\lambda = K_8\ K_7\ K_6\ K_5}$.

$\mathbf{T_p = K_9}$.

$\mathbf{T_d = K_{10}}$.

Step2: Generate a random sequence of numbers with logistic map

$$\mathbf{r(i) = \lambda * r(n-1) * [1 - r * n - 1)] \quad where\ i = 1\ to\ 15}$$

Step3: Generate another chaotic sequence using standard map

$$x_{k+1} = (x_k + y_k + r_x + r_y) \bmod N,$$

$$y_{k+1} = \left(y_k + r_y + K_C \sin \frac{2\pi x_{k+1}}{N} \right) \bmod N,$$

Where x(1) = r(1), y(1) = r(2), Kc = r(3), rx = r(4), ry = r(5) and N=256.

Step4: Separate the color image into 3 planes of each color:

R = P(: , :, 1), G = P(: , :, 2), B = P(: , :, 3).

Step5: Shuffle the red plane R of the image P with respect to chaotic sequence generated by standard map:

(1) Generate randomly two vectors Xr and Yr of length m and n, respectively. Note that both sequence must not have constant values.

(2) Determine the number of iterations, ITERmax, and initialize the counter ITER at 0.

(3) Increment the counter by one: ITER = ITER + 1.

(4) For each row i of image R,

(a) Compute the sum of all elements in the row i, this sum is denoted by α(i)

$\alpha(i) = \sum_{nj=1} R(i,j)$ i=1 to m

(b) computemodulo 2 of α(i), denoted by Mα(i),

(c) row i is left, or right, circular-shifted by Xr(i) positions (image pixels are moved Xr(i) positions to the left or right direction, and the first pixel moves in last pixel.), according to the following:

if Mα(i) = 0 −→ right circular shift

else −→ left circular shift.

(5) For each column j of image R,

(a) compute the sum of all elements in the column j, this sum is denoted by β(j),

$\beta(j) = \sum_{mi=1} R(i,j)$ ……. j=1 to n

(b) computemodulo 2 of β(j), denoted by Mβ(j).

(c) column j is down, or up, circular-shifted by Yr(i) positions, according to the following:

if Mβ(j) = 0 −→ up circular shift

else −→ down circular shift.

Steps 4 and 5 above will create a scrambled image, denoted by Sr.

Step:6 The above procedure is carried for green and blue planes also with different chaotic sequence. To generate them standard map is used along with initial conditions being specified in sequence generated by logistic map. Sg and Sb planes are generated for green and blue planes.

Step7: Diffusion: After scrambling the image pixels, they scaned from top left corner and pixel values are changed. For scanning a different seed value is taken for three different color planes. Seed value= Ks

Ks = Xr(1) ⊕ Yr(1).

And then a XOR and SHIFT operation is performed on each pixel.

$$bs \leftarrow Cr(i-1)\ \&\ 7$$

$$Sr(i) \leftarrow Sr(i) \oplus Cr(i-1)$$

$$Cr(i) \leftarrow (\ Sr(i) >> bs\ ) \mid (Sr(i) << (\ 8 - bs\ )\ )$$

Where Sr(i) is the ith pixel of scrambled red plane, Cr(i) is the ith pixel after scanning the scrambled image Sr. For the first pixel C(0) = Ks is taken.

Same scanning process is repeated for green and blue planes:

Ks = Xg(1) ⊕ Yg(1).

And then a XOR and SHIFT operation is performed on each pixel.

$$bs \leftarrow Cg(i-1)\ \&\ 7$$

$$Sg(i) \leftarrow Sg(i) \oplus Cg(i-1)$$

$$Cg(i) \leftarrow (\ Sg(i) >> bs\ ) \mid (gr(i) << (\ 8 - bs\ )\ )$$

Ks = Xb(1) ⊕ Yb(1).

And then a XOR and SHIFT operation is performed on each pixel.

bs ← Cb(i-1) & 7

Sb(i) ← Sb(i) ⊕ Cb(i-1)

Cb(i) ← (Sb(i) >> bs) | (Sb(i) << (8 - bs))

Step8: Permutation along with diffusion is repeated ITER_T times, for every color planes separately. Then the color planes are combined

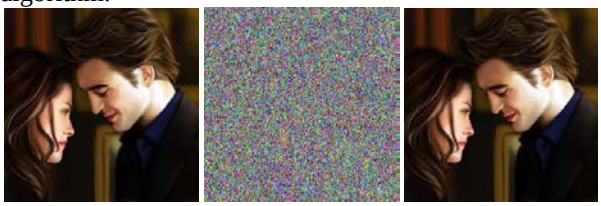
C = Con(Cr, Cg, Cb).

Image C obtained after several rounds of permutation and diffusion is the encrypted image.

5. Performance Analysis

A. Simulation results

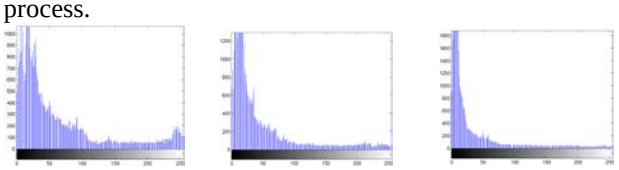
The encryption and decryption algorithms are implemented in MATLAB. The simulation results demonstrate that the proposed new algorithm shows good performances in image encryption. The encrypted image in Figure 2(b) is completely different from the original image and cannot be recognized. The decrypted image in Figure 2(c), getting from the decryption process, is the same as the original image in Figure 2(a). This shows the success of the encryption and decryption algorithm.



2(a) 2(b) 2(c)

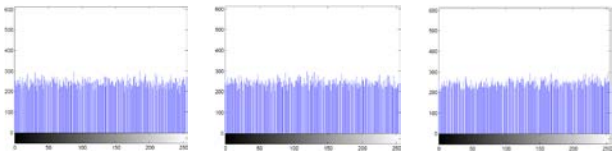
Fig 2(a) shows the original image, 2(b) shows encrypted image, 2(c) shows decrypted image with the same key.

We also observe the histograms of the original, encrypted and decrypted images. Histogram of the encrypted image has been equalized after the encryption process.



3(a) 3 (b) 3 (c)

Fig 3(a), (b) and (c) shows the histogram of original image for red, green, and blue planes respt.



4(a) 4(b) 4(c)

Fig 4(a), (b) and (c) shows the histogram of encrypted image for red, green, and blue planes respt.

B. Correlation test

Statistical analysis on large numbers of images shows that averagely adjacent 8 to 16 pixels are correlated. To test the correlation between horizontally, adjacent pixels from the image, we calculate the correlation coefficient of a sequence of adjacent pixels by using formulas

$$E(x) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} P_1(i,j)$$

$$D(P_1) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N}[P_1(i,j) - E(P_1(i,j))]^2$$

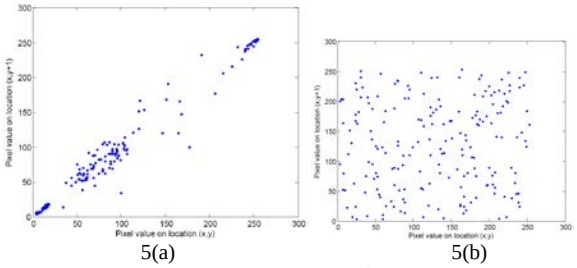
$$\mathrm{cov}(P_1,C_1) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N}[P_1(i,j) - E(P_1(i,j))][C_1(i,j) - E(C_1(i,j))]$$

$$r_{P_1C_1} = \frac{\mathrm{cov}(P_1,C_1)}{\sqrt{D(P_1)}\sqrt{D(C_1)}}$$

where P1(i, j) and C1(i, j) are the grayscale values of the original pixel and the encrypted one. According to cryptographic requirements, a good encryption system should have a high level of security to resist many well-known attacks. Following the principle in [12, 13], the security level of an encryption system depends on its security keys. If the key space is small or the key is not well-designed, the encryption system is not able to resist the different attacks, even though the encryption algorithm is much complicated. Here, the security key analysis is performed to test the security level of the proposed image encryption algorithm.

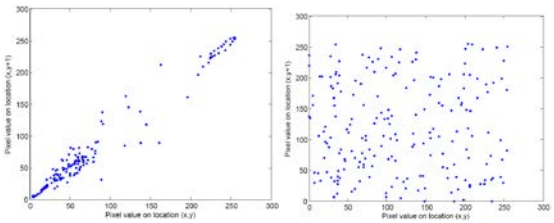
TABLE I correlation coefficient values

Defect	Original Image	Encrypted Image
Red plane	0.3069	0.0045
Green plane	0.2983	0.0037
Blue plane	0.3218	0.0045



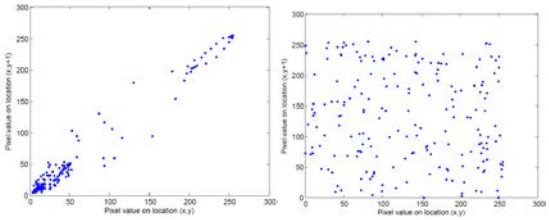
5(a) 5(b)

Fig 5(a) and (b) shows the correlation of two adjacent pixels in red plane of original and encrypted images respt.



6(a) 6(b)

Fig 6(a) and (b) shows the correlation of two adjacent pixels in green plane of original and encrypted images respt.



7(a) 7(b)

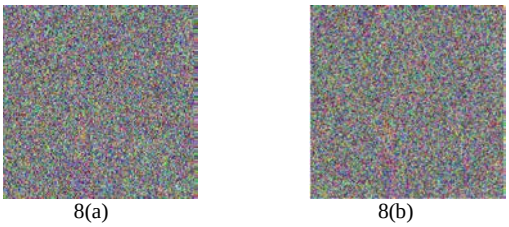
Fig 7(a) and (b) shows the correlation of two adjacent pixels in blue plane of original and encrypted images respt.

A. Key space

One of the requirements for the security key is that the key Space should be large enough to resist the brute-force attack. Given today's computer speed, the system has a certain level of security only if the key space is of size k > 2100 [15, 16]. The proposed encryption algorithm is a 128-bit encryption scheme with the key space size of 2240≈1.7668∗1072. Compared with the space requirement (2100), the key space of this proposed encryption algorithm is more than twice large. It means that the algorithm is enough to withstand the brute-force attack.

B. Key sensitivity

A large key space does not mean using more bits to represent the same parameters and initial values. The encryption key in the proposed encryption algorithm is composed of eight parts which determine the parameters and initial values. Every part consists of 24 bits which means that a single difference in the key will result in a value change (at least 5.9605∗10−8). A well-known test has been performed in this paper. The original image is encrypted with the key1 and key2 to obtain two encrypted images. The absolute value of the difference between these two encrypted images is then obtained.
Key1="012345678909876543210123"
Key2="012345678909876543210124"



8(a) 8(b)

Fig 8(a) and (b) shows the encrypted image for key1 and key2 respt.

The difference between key1 and key2 is value of the last bit. It is easy to calculate that the encrypted image by key1 has 95.02% difference from the encrypted image by key2. The key sensitivity analysis should also be tested in the decryption process. As shown in Figure 8, the same test performed in the decryption process using two different decryption keys (denoted as Dekey1 and Dekey2). Dekey1 is generated by the encryption process with the encryption key. Dekey2 is generated by making a slight change in the value of the first bit in Dekey1.
Dekey1="012345678909876543210123"
Dekey2="012345678909876543210124"
From the results shown in Figure 9(a), a slight change in the decryption key will lead to the failure of the image decryption.
As shown in Figure 9(b), the decrypted image using dekey2 has not been recognized.

9(a) 9(b)

Fig 9(a) and (b) shows the decrypted image for key1 and key2 respt.

C. Information entropy

Histogram analysis just shows the result of the encryption algorithm in a qualitative way. To get the quantitative analysis, information entropy is utilized [14]. Information entropy, as a measure of disorder, can quantify the uniformity of histogram.
The function of the information entropy is defined as Eqn. (16).

$$E = -sum(p.*log2(p))$$

TABLE IIi entropy values

Entropy	Original Image	Encrypted Image
Red plane	7.1141	7.9954
Green plane	6.6579	7.9959
Blue plane	5.8835	7.9959

Information entropy has been increased with the encryption system and the information entropy of the encrypted image is almost near to 8. This confirms that the pixel values after encryption process seems random, which is sufficient secure for information leakage.

6. CONCLUSION

We have introduced a novel image encryption algorithm using the proposed new chaotic system. Meantime, histogram analysis shows the encryption performance of the proposed algorithm. The security key analysis shows that algorithm has the sufficiently large key space to resist the brute-force attack and high sensitivity to the key changes for encryption and decryption. All these show that the proposed encryption algorithm has a high level of security.

References

[1] B. Furht, D. Socek, A.M. Eskicioglu, Fundamentals of Multimedia Encryption Techniques, in B. Furht and D. Kirovski (Eds.), Multimedia Security Handbook, Ch. 3, CRC Press, 2005.

[2] M. Podesser, H.-P. Schmidt, A. Uhl, Selective Bitplane Encryption Scheme for Secure Transmission of Image Data in Mobile Environments, Proc. of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02), Trondheim, Norway, October 2002.

[3] C. Alexopoulos, SCAN, A Language for 2-D Sequential data accessing, PhD. thesis, University of Patras, Greece, 1989. 106

[4] N. Bourbakis, C. Alexopoulos, Picture Data Encryption using SCAN Patterns, Pattern Recognition 25(6), pp. 567-581, 1992.

[5] C.P. Wu, C.C. Kuo, Design of Integrated Multimedia Compression and Encryption Systems, IEEE Trans. Multimedia 7(5), pp. 828-839, 2005.

[6] T. Xiang, K.W. Wong, X. Liao, Selective Image Encryption using Spatiotemporal Chaotic System, Chaos 17(2), paper 023115, June 2007.

[7] J. Fridrich, Symmetric Ciphers Based on Two-dimensional Chaotic Maps, Int. J. Bifurcat. Chaos 8(6), pp. 1259-1284, 1998.

[8] S. Neil-Rasband, Chaotic Dynamics of Nonlinear Systems, John Wiley & Sons Inc., 1990.

[9] A. B. Campbell, Applied Chaos Theory: A paradigm for complexity, Academic Press Inc., pp. 81-125, 1993.

[10] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, 1993. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems.

[11] Khaled Loukhaoukha, Jean-Yves Chouinard, and Abdellah Berdai, A Secure Image Encryption Algorithm Based on Rubik's Cube Principle, Journal of Electrical and Computer Engineering, Volume 2012.

[12] G. Chen, Y.B. Mao, C.K. Chui, A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps, Chaos, Solitons and Fractals 12, pp. 749- 761, 2004. 105

[13] Y.B. Mao, G. Chen, S.G. Lian, A Novel Fast Image Encryption Scheme Based on the 3D Chaotic Baker Map, Int. J. Bifurcat. Chaos 14(10), pp. 3613-3624, 2004.

[14] Sodeif Ahadpour, Mahdiyeh Majidpour, Yaser Sadra, Public key Steganography Using Discrete Cross- Coupled Chaotic Maps, page 5.

[15] C. K. Huang and H. H. Nien, "Multi chaotic systems based pixel shuffle for image encryption," Optics Communications, vol. 282, no. 11, pp. 2123–2127, 2009.

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

[17] K.W. Wong, S.H. Kwok, An Efficient Diffusion Approach for Chaos-based Image Encryption, Proceedings of the Third International IEEE Scientific Conference on Physics and Control (PhysCon 2007), Potsdam, Germany, September 3-7, 2007.