Computer Algorithms for Direct Encryption and Decryption of Digital Images for Secure Communication

A. N. PISARCHIK
Photonics Department
Centro de Investigaciones en Optica
Loma del Bosque 115, Lomas del Campestre, 37150 Leon, Guanajuato
MEXICO
http://www.cio.mx
N. J. FLORES-CARMONA
Computer Department
Institute of Technology of Leon
Av. Tecnologico, Julian de Obregon, 37290 Leon, Guanajuato
MEXICO

Abstract: - In this paper we design a new practical computer algorithm based on chaotic map lattices which allows direct encryption and decryption of color digital images. The basic idea is to convert, pixel by pixel, the image color to logistic chaotic maps one-way coupled by initial conditions. After small numbers of iterations and cycles, the image becomes indistinguishable due to inherent properties of chaotic systems. The image can be completely recovered by the corresponding decryption algorithm in the case if all secure keys, such as map parameters, number of iterations, number of cycles, and the image size are exactly known. We test our algorithm with a real color image and prove that our cryptosystem incorporates necessary properties inherent to a good cryptosystem. These are (i) high sensitivity to any changes in the image, (ii) high sensitivity to secret keys, (iii) absence of any patterns in the encoded image, and (iv) robustness against cryptographic attacks. We also discuss possibilities for improving our algorithm with further developments in computer techniques and new coupled schemes.

Key-Words: - Cryptography, chaos, logistic map, image encryption, secure communication

1 Introduction

In the last decade, the chaos theory has attracted much attention due to its applications to cryptography and secure computer communications [1-6]. Cryptography is the art of the secret writing. The main purpose of cryptography is to develop a cryptosystem which converts an original message into a non-readable message and then it recovers the message back in its original form. This involves information transformation to apparent understandable garbage in order to non-authorized people cannot understand the message [7]. High sensitivity of chaotic systems to initial conditions and parameters implies strong cryptographic properties of a chaotic cryptosystem that makes it robust against any statistical attacks. Therefore, the use of chaos in cryptography is of great interest to many areas, including a data base, internet transaction banking, software, protection of communication channels, in order to preserve confidential data from indiscretion attacks of enemies, spies, interceptors, opponent, cryptanalysts, etc. [8,9].

Chaotic communication schemes are based either on discrete or continuous systems. Here we consider discrete systems due to their simplicity and rapidness. Most of the discrete chaotic cryptographic algorithms explore one or more chaotic maps as pseudorandom number generators producing a binary stream which is used for encryption of a plain text to produce a cipher text [5,10-14]. The initial conditions or parameters or both have been used as secret keys. The existing algorithms utilize a block encryption technique which allows the encryption of plaintext files (blocks of bits); however they do not allow the direct encryption of images. The latter requires the use of spatially extended dynamical systems, e.g. 2D map lattices. The chaotic map lattice (CML) has been introduced by Kaneko [15] as a simple model capturing essential features of spatiotemporal dynamics of extended nonlinear systems. Later CML has been applied for modeling complex spatial phenomena in diverse areas of science and engineering. Recently, Wang *et al*. [16] have shown that the communication with the CMLs is more secure that the communication with a single map.

Due to finite precision of computer calculations, chaotic discrete systems always generate periodic time series, however, the period of the periodic orbits increases exponentially with the number of coupled maps.

In this paper, we suggest a new approach to secure computer communications based on CMLs. Our cryptosystem is different from all known cryptosystems because (i) it does not utilize a block encryption technique and (ii) it does not require synchronization of a receiver with a transmitter. Our computer algorithm utilizes only the essence of chaos: high sensitivity of a chaotic trajectory to initial conditions and to system parameters, confinement of the motion to a finite region of the phase space, and recurrence properties of a chaotic trajectory (i.e., any trajectory originating inside the attractor always remains within it and visits all points of the attractor in infinite time). This work is the first attempt, to our knowledge, of exploring the CML for direct encryption and decryption of digital images.

The paper is organized as follows. In Section 2 we describe the chaotic map lattice and our encryption/decryption algorithm. In Section 3 we prove our method with a computer experiment by encoding a real color image and study the sensitivity of our cryptosystem to secret keys. Finally, the main results are summarized in Section 4.

2 Chaotic Map Lattices

The logistic map is one of the simplest nonlinear chaotic discrete systems known as

$$x_{n+1} = a x_n (1 - x_n), \qquad (1)$$

where x_n and a are the system variable and parameter, respectively, and n is the number of iterations. For $3.57 < a < 4$ the map (1) is chaotic. The main idea of our cryptosystem is that any image can be represented as lattices of pixels, each of which has a particular color. The pixel's color is the combination of three components: red, green and blue, each of which takes an integer value $C = (C_r, C_g, C_b)$ between 0 and 255. Thus, we can create three parallel CMLs by converting each of these three color components to the corresponding values of the map variable, $x_c = x_c^r, x_c^g, x_c^b$, and use these values as the initial conditions, $x_c = x_0$. Starting from different initial conditions, each chaotic map in the CMLs, after a small number of iterations, yields a very different value from the initial conditions, and the image becomes indistinguishable because of an exponential divergence of chaotic trajectories. In order to be able to recover the image, the maps are coupled by the initial conditions, i.e. the initial condition x_0^i of the map i depends on the final variable of the previous map i-1 after n iterations, x_n^{i-1}, and contains information about the pixel's color.

The process of developing the chaos-based cipher can be summarized as follows. Let a = 3.9; for this parameter the chaotic attractor occupies the phase space between x_{min} = 0.0950626 and x_{max} = 0.975. To convert the color components C of each pixel to the variable x_c of the corresponding map in the lattice, we use the following transformation:

$$x_c = x_{min} + \delta x (C/255), \qquad (2)$$

where $\delta x = x_{max} - x_{min}$. To extract the value of the color component, we apply the inverse function:

$$C = round\,[(x_n - x_{min})255/\delta x]. \qquad (3)$$

Equation (3) allows us to transform any state x_n of the logistic map Equation (1) into a value between 0 and 255 and thus to visualize the pixel's color.

2.1 Encryption algorithm

The encryption algorithm includes the following steps.

Step 1. Let an image contains $N \times M = m$ pixels (i = 1,2,...,m) as shown in Figure 1(a). Three color components of the pixel i are converted to three values of the variable x_c with Equation (2). For example, if the color values of the the pixel i are C^i = 64, 121, and 176 for red, green and blue components, respectively, we obtain x_c^i = 0.315909654, 0.512601554, and 0.702392.

Step 2. The color value x_c^m of the last map m in the lattice is used as the initial condition for the first map i=1, i.e. $x_0 = x_c^m$.

Step 3. After n iterations of the first map, we obtain the map variable x_n^1 and add to this value the color value of the pixel, x_c^1. The sum value is used as the initial condition for the subsequent map, $x_0^2 = x_n^1 + x_c^1$.

Step 4. We iterate all maps subsequently starting from the first map and going through all image pixels, pixel by pixel, towards the last map, as shown in Figure 1(a). In order to obtain always a stable solution and exclude transients, the trajectory should be initiated inside the chaotic attractor, i.e. $x_0 \in [x_{min}, x_{max}]$. Therefore, if the sum $x_n^i + x_c^i > x_{max}$, we subtract δx, i.e. $x_0^{i+1} = x_n^i + x_c^i - \delta x$. After one cycle, going from the first map to the last one, we obtain the map lattice

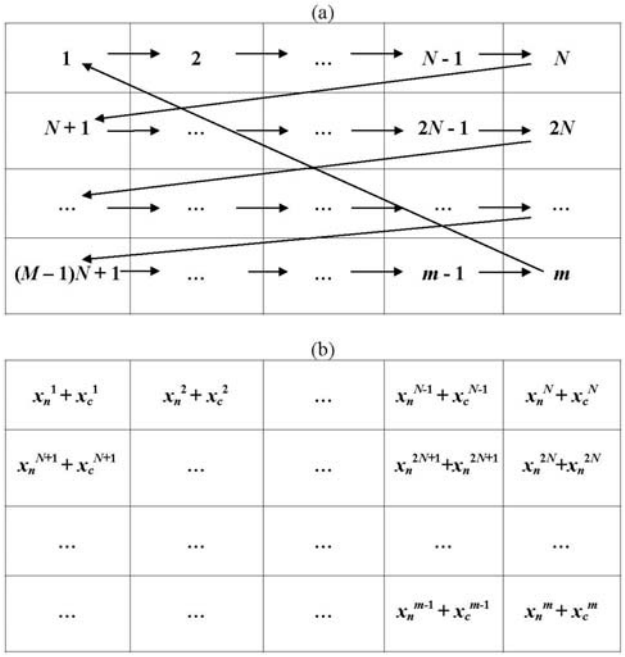

Fig. 1. (a) Indices for image pixels and (b) encoded variables. The arrows indicate coupling directions.

shown in Figure 1(b) which we can be visualized by converting the new map variables $x_n^i+x_c^i$ to the corresponding color values by using Equation (3).

Step 5. We repeat steps 3 and 4 and make several cycles. For the next cycle, the new color value of the last map, $x_c^m(j) = x_n^m(j)+x_c^m(j)$ (j being the number of the cycle), is used as the initial condition for the first map to initiate the next cycle, i.e. $x_0^1(j+1) = x_c^m(j)$. After j cycles we obtain the map lattice similar to that shown in Figure 1(b) and can visualize the encoded image using Equation (3).

Step 6. We repeat all steps for each color component (red, green, and blue) and superimpose the three images.

Thus, the encryption algorithm is

$$x_0^i(j) = x_c^m(j-1), \quad \text{if } i = 1, \tag{4}$$
$$x_0^{i+1}(j) = x_c^i(j), \quad \text{if } i > 1, \tag{5}$$
$$x_c^i(j) = x_n^i(j\text{-}1)+x_c^i(j\text{-}1), \quad \text{if } x_n^i(j\text{-}1)+x_c^i(j\text{-}1) \le x_{max}, \tag{6}$$
$$x_c^i(j) = x_n^i(j\text{-}1)+x_c^i(j\text{-}1)-\delta x, \text{ if } x_n^i(j\text{-}1)+x_c^i(j\text{-}1) > x_{max}. \tag{7}$$

2.2 Decryption algorithm

The encoded image is converted with Equation 3 to the map lattice $x_c^i(j)$. For decryption, we need to recover the original image cycle by cycle in the reverse direction starting from the last map m and going to the first map i=1 by making the same number of iterations for each map as for the encryption process. The decryption algorithm includes the following steps.

Step 1. First, we need to recover the image of the j-1 cycle. We start from the last map. The encoded value of the penultimate map in the j cycle is the initial condition for the last map in the j-1 cycle, i.e. $x_0^m(j\text{-}1) = x_c^{m-1}(j)$. Starting from this initial condition, we iterate the last map n times and obtain the value $x_n^m(j\text{-}1)$. The number of the iterations should be the same as for the encryption process. The color value of the last map in the j cycle is $x_n^m(j\text{-}1)+x_c^m(j\text{-}1)$. Subtracting $x_n^m(j\text{-}1)$, we get the color value of the last map m in the j-1 cycle, i.e. $x_c^m(j\text{-}1)$.

Step 2. Taking the encoded value of the map m-2 as the initial condition for the map m-1, we find its color value in the cycle j-1 and so on.

Step 3. We repeat step 2 for each map in the reverse direction from the last map to the first map and reconstruct the image of the cycle j-1. Note, this it is not the original image.

Step 4. To reconstruct the color value of the first map in the j-1 cycle, we use the color value of the last map m in the j-1 cycle, $x_c^m(j\text{-}1)$, as the initial condition for the fist map i = 1.

Step 5. We repeat all previous steps j times to obtain the map lattice $x_c^i(0)$ and convert it to the color values C^i by Equation (3).

Step 6. We repeat all steps for each color component (red, green, and blue) and superimpose the three images to obtain the original image.

The decryption algorithm is

$$x_0^i(j\text{-}1) = x_c^{i-1}(j), \quad \text{if } i > 1, \tag{8}$$
$$x_0^i(j\text{-}1) = x_c^m(j\text{-}1), \quad \text{if } i = 1, \tag{9}$$
$$x_c^i(j\text{-}1) = x_c^i(j) - x_n^i(j\text{-}1), \quad \text{if } x_c^i(j) - x_n^i(j\text{-}1), \tag{10}$$
$$x_c^i(\text{j-}1) = x_c^i(j) - x_n^i(j\text{-}1)+\delta x, \text{ if } x_c^i(j) - x_n^i(j\text{-}1) < 0. \tag{11}$$

Thus, our cryptographic algorithm has four secrete keys: system parameters, number of iterations, number of cycles, and the image size. For higher security each map can have differing parameter and differing number of iterations.

3 Computer Experiment

Conventional cryptography deals with binary streams and utilizes the terms of *plaintext* (the original text to be encoded) and *ciphertext* (the encoded text). A good cryptosystem should incorporate the following features: (i) be sensitive with respect to a plaintext (slight modification in the plaintext creates completely different ciphertext); (ii) be sensitive with respect to keys (change in a secret key produces a completely different ciphertext); and (iii) map a plaintext to a random ciphertext (no any patterns in the ciphertext).

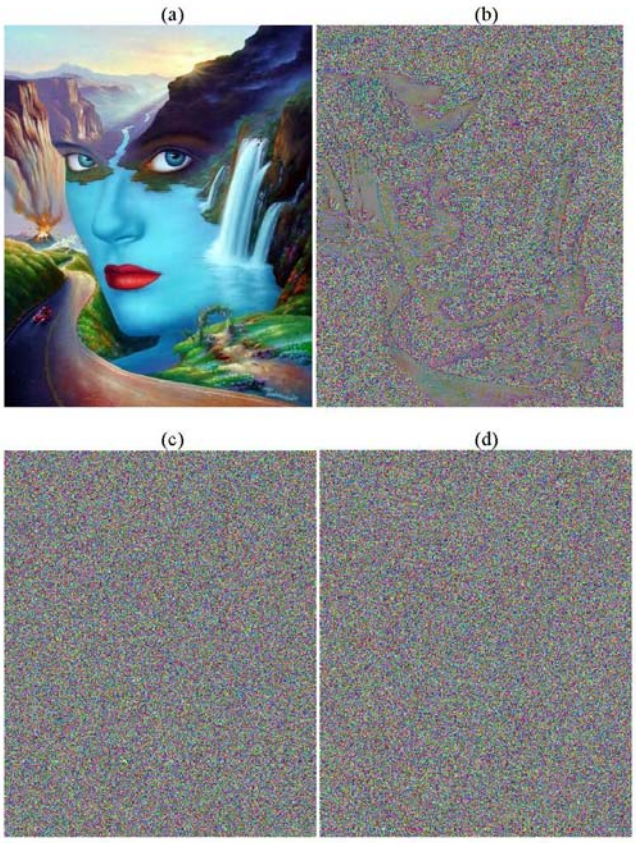

Fig. 2. Sensitivity to number of iterations. (a) Original image, (b) image encoded with n = 1, (c) n = 30, and (d) n = 75. a = 3.9 and j = 3.

Since our cryptosystem does not deal with binary streams and since both the original image and encoded image are digital, we will use the terms *original image* and *encoded image* instead of "plaintext" and "ciphertext". In the following we will demonstrate that our cryptosystem compiles all cryptographic properties inherent with a good cryptosystem.

As we mentioned above, our cryptosystem has four secret keys: the system parameter a, the number of iterations n, the number of cycles j, and the image size $m = N \times M$. Here, we consider the sensitivity of our cryptosystem to the secrete keys. The original image ($N \times M$ = 455 pixels) is shown in Figure 2(a). The sensitivity of our encryption algorithm to the number of iterations n is demonstrated in Figs. 2(b)-2(d) where we display the images encoded with n = 1, n = 30, and n = 75, respectively.

For visualization, the values of the map variables $x_c^i(j=1)$ and $x_c^i(j=3)$ are converted to the color numbers C^i by Equation (3). No decryption algorithm Equations (8-11) are used. These figures illustrate a crucial dependence of chaotic trajectories on initial conditions. One can see that using only 1 iteration for each map, the image can be still distinguished even after 3 cycles [Figure 2(b)]. The use of 30 iterations makes the image almost indistinguishable, but the colors are not uniformly distributed [Figure 2(c)]. However, for 75 iterations all colors are completely lost in the encoded image [Figure 2(d)].

The number of iterations n is not so critical for the encryption/decryption time (EDT), as the number of cycles j. For example, EDT for the images shown in Figures 2(b)-2(d) are varied between 165 and 170 seconds.

The sensitivity of our encryption algorithm to the number of cycles j is demonstrated in Figures 3(a) and 3(b), in which we display the images encoded with 1 and 2 cycles, respectively. As seen from the figures, only 1 or 2 cycles are not sufficient for secure encryption of the image because the original outline still can be distinguished in the encoded image. Therefore, we need to make at least 3 cycles to get an indistinguishable image, as shown in Figure 2(d).

The larger n and j are, the better security is. From the other hand, EDT also becomes longer. To have reasonable EDT, we should balance between these two factors (security and time) to select adequate values for n and j. Moreover, EDT increases significantly with increasing m, as one can see from Table 1, where we show EDT (in seconds) for three different image sizes and three different j. The calculations have been performed with Pentium IV 3.0 GHz PC, 1.0 GB RAM and visualized with Microsoft Visual C#.NET 2005. From Table 1 one can see that for higher resolution and security, we need to sacrifice to time.

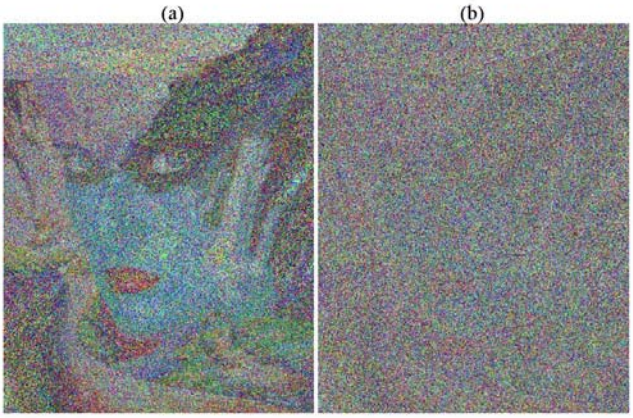

Fig. 3. Sensitivity to number of cycles. (a) Image encoded with j = 1, and (b) j = 2. a = 3.9 and n = 75.

Table 1. Details of encryption/decryption time.

Image size ($N \times M$ pixels)	**Time (j = 1)**	**Time (j = 2)**	**Time (j = 3)**
300×200	13.6	26.7	39.1
455×569	58.0	113.0	169.3
2400×1200	626.6	1255.9	1866.2

To prove the first property of a good cryptosystem, we slightly modify the original image with a black square at the right inferior corner [Figure 4(a)]. The modified image encoded with 75 iterations and 3 cycles is shown in Figure 4(b) and its histogram is plotted in Figure 4(c). The histogram of the encoded original image is shown in Figure 4(d). Although the both encoded images [in Figures 2(d) and 4(b)] are generated by the same keys, the distribution of their colors is completely different [compare Figures 4(c) and 4(d)].

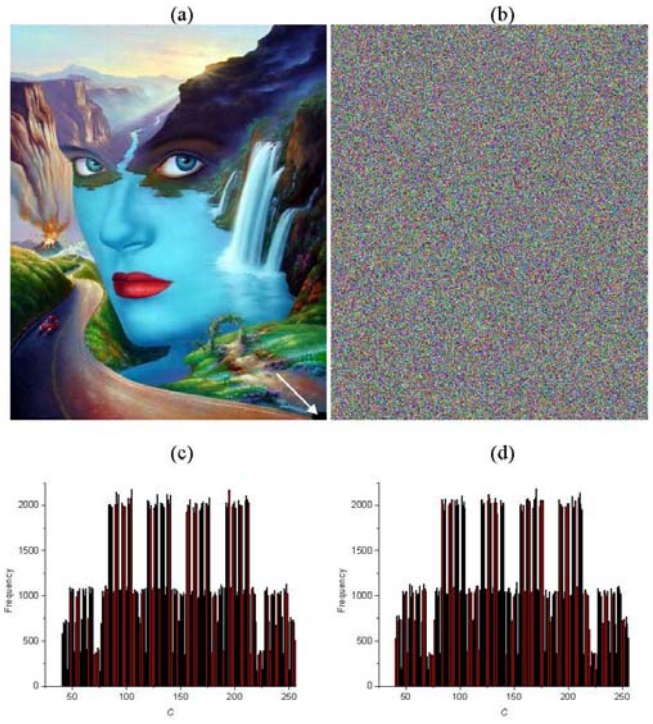

Fig. 4. Sensitivity with respect to plaintext. (a) Slightly modified image with black square indicated by white arrow, (b) encoded modified image, (c) histogram of encoded modified image, and (d) histogram of encoded original image shown in Figure 2(d).

Finally, our cryptosystem complains the third property of a good cryptosystem. The image encoded with a relatively small number of iterations and cycles does not display any patterns, as seen in Figure 2(d).

4 Conclusion

In this paper, we have described a new secure compute algorithm based on chaotic map lattices for encoding and decoding digital images. Our algorithm explores the important properties of chaos: localization of a chaotic attractor in a particular region of the phase space, recurrence of a chaotic trajectory (it visits all points of the chaotic attractor in infinite time) and its high sensitivity to initial conditions and map parameters. The first two properties allow decoding an infinite number of colors, and the second one provides a very high security of the chaotic cryptosystem, because after a small number of iterations the trajectory occurs far away from the initial state. We have demonstrated how a color image can be directly converted to lattices of chaotic logistic maps one-way coupled by initial conditions and how the image can be completely recovered if all secret keys are exactly known. In our encryption algorithm the coupling is necessary for both further decoding of the image and conserving information about the pixel's colors. Our encryption/decryption algorithm complies with all essential properties for a good cryptosystem and can be easily adapted to other chaotic maps including two-dimensional maps, like the Hénon map or Baker map.

The important problem in computer communications and telecommunications is the possibility of communicating in real time. Unfortunately, the speed of modern personal computers is still insufficient for communicating with our algorithm in real time. However, we believe that progressing development of computer technology and further improvement of the algorithm will allow decreasing the encryption/decryption time. The future possible research in this direction we suppose will be the use of mutual (spatial) coupling instead of one-way coupling. This will make an image indistinguishable after smaller numbers of iterations and cycles. Other developments of cryptosystems based on CMLs might be the use of transmitter-receiver coupling, i.e. each map of a transmitter could be coupled with the corresponding map of a receiver and then, to recover an image we may apply a conventional chaotic communication technique based on complete synchronization. The latter approach does not require reverse calculations and therefore will safe time and probably will allow computer communications and telecommunications in real time.

Finally, our encryption/decryption algorithm can be extended to a 3D chaotic map system, i.e. to a volume of maps. This will allow direct encoding 3D images, like holograms, which we believe will be widely used in future communications.

The work was supported by the Mexican Council of Science and Technology (CONACYT), Project No. 46973. A. N. P. acknowledges support from the Spanish Ministry of Education and Science, Project No. SAB2004-0038. We thank Jim Warren for the permission to explore his picture in our work.

References:

[1] L. M. Pecora and T. L. Carroll, Synchronization in Chaotic Systems, *Phys. Rev. Lett.*, Vol. 64, No. 9, 1990, pp. 821-824.

[2] K. M. Cuomo and A. V. Oppenheim, Circuit Implementation of Synchronized Chaos with Applications to Communications, *Phys. Rev. Lett.*, Vol. 71, No. 1, 1993, pp. 65-68.

[3] L. Kocarev and U. Parlitz, General Approach for Chaotic Synchronization with Applications to Communication, *Phys. Rev. Lett*., Vol. 74, No. 25, 1995, pp. 5028-5031.

[4] D. G. Van Wiggeren and R. Roy, Communication with Chaotic Lasers, *Science*, Vol. 279, No. 5354, 1998, pp. 1198-1200.

[5] N. K. Pareek, V. Patidar, and K. K. Sud, Discrete Chaotic Cryptography Using External Key, *Phys. Lett. A*, Vol. 309, Nos. 1-2, 2003, pp. 75-82.

[6] L. Kocarev, M. Sterjev, A. Fekete, and G. Vattay, Public-Key Encryption with Chaos, *Chaos*, Vol. 14, No. 4, 2004, pp. 1078-1082.

[7] A. J. Mendezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[8] Li Shujun, *Analyses and New Designs of Digital Chaotic Cipher*, Xi'an Jiaotong University, 2003.

[9] D. Bishop, *Introduction to Crypography with Java Applets*, Jones and Bartlett Publishers, 2003.

[10] Z. Kotulski and J. Szczepanski, Discrete Chaotic Cryptography, *Ann. Phys*., Vol. 6, No. 5, 1997, pp. 381-394.

[11] M. S. Baptista, Cryptography with Chaos, *Phys. Lett. A*, Vol. 240, 1998, pp. 50-54.

[12] E. Alvarez, A. Fernandez, P. Garcia, J. Jimenez, and A. Marcano, New Approach to Chaotic Encryption, *Phys. Lett. A*, Vol. 263, Nos. 4-6, 1999, pp. 373-375.

[13] K. W. Wong, S. W. Ho, and C. K. Yung, A Chaotic Cryptography Scheme for Generating Short Ciphertext, *Phys. Lett. A*, Vol. 310, No. 1, 2003, pp. 67-73.

[14] N. K. Pareek, V. Patidar, and K. K. Sud, Cryptography Using Multiple One-Dimensional Chaotic Maps, *Communications in Nonlinear Science and Numerical Simulations*, Vol. 10, 2005, pp. 715-723.

[15] K. Kaneko and I. Tsuda, *Complex Systems: Chaos and Beyond: A Constructive Approach with Applications in Life Sciences*, Springer-Verlag, 2001.

[16] S. Wang, W. Liu, H. Lu, J. Kuang, and G. Hu, Periodicity of Chaotic Trajectories in Realizations of Finite Computer Precisions and Its Implication in Chaos Communications, *Int. J. Mod. Phys. B*, Vol. 18, Nos. 17-19, 2004, pp. 2617-2622.