A Chaotic Encryption System Using PCA Neural Networks

Xiao Fei , Guisong Liu, Bochuan Zheng
Computational Intelligence Laboratory, School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 610054, P. R. China
Email: fxiao.uestc@gmail.com

***Abstract*—This paper introduces a chaotic encryption system using a principal component analysis (PCA) neural network. The PCA neural network can produce the chaotic behaviors under certain conditions so that it serves as a pseudo-random number generator to generate random private keys. In this encryption system, the one-time pad encryption method is used, which is regarded as the most secure encryption method. The proposed system can encrypt any kind of data. The security and high performance of encryption are illustrated via some simulations.**

***Index Terms*—Principal Component Analysis, Neural Networks, Chaotic Encryption, Information Security.**

I. Introduction

With modern technology and network developing, the digital information has been applied to many areas in the world. However, because of its easy replicability, information security problem has become more and more important. Cryptography is one of basic methodologies for information security. In recent years, chaos theory consistently plays an active role in cryptography [1][2]. The attractiveness of using chaos as the basis for developing cryptosystem is mainly due to its random behavior and sensitivity to initial conditions and parameter setting that fulfill the classic Shannon requirements of confusion and diffusion [3] [4]. In those classic encryption methods, one-time pad is thought as the most secure way in encryption. In this paper, an encryption algorithm is presented by using chaos theory to implement one-time pad encryption. By this method, each data stream will correspond to a pseudo-random private key. Therefore, even by brute-force attacks, it is impossible to decode in a short time.

Chaos is a ruleless and random phenomenon and usually occurs in a deterministic system. This phenomenon is very common in the natural life, such as paths of light, changes of weather and fluctuations of economic. These disorderly and unsystematic phenomena have some amazing features. The most famous case is the Butterfly Effect: while a butterfly fans its wings at Brazil, it may cause a tornado at Texas of United States. The characteristic of Chaos, especially its sensitivity to initial value, is shown in this phenomenon. Chaos system has some geometric and statistical characteristics which deterministic system doesn't have. Different from other complex phenomena, chaos has the following characteristics:

(1) Ergodicity: All trajectories of a chaos is confined to a fixed region, a chaotic domain of attraction. The trajectories transit each output state in this domain.

(2) Randomness: If the selected parameters of a chaos is in the chaotic domain, the output sequence generated by the chaotic system is chaotic and pseudo-random.

(3) Certainty: Chaos is generated by a deterministic system. When the system parameters and the initial values are invariant, the output of the random sequence is invariant and repeatable.

(4) Extreme sensitivity to initial conditions: Even any minor changes on the initial conditions will have a big impact.

The relationships between the basic characteristics of chaos and the common cryptography have been discussed by Shannon in his classic article [3], where he proposed two basic principles to guide the cryptography design: diffusion and confusion. The chaotic mixing characteristic in an orbit is corresponding to the traditional characteristics of diffusion in encryption system. Chaotic signals with random characteristic and the sensitivity to system parameters are related to the conventional characteristics of confusion in encryption system. Therefore, these two characteristics of chaos are able to guarantee the confusion and diffusion of a chaotic encryption system, like other traditional encryption methods.

The rest of the paper is organized as follows. Section II gives a detailed description of the chaos encryption algorithm. The designing method is also discussed. Security analysis and conclusions are given in Secions III and IV, respectively.

II. The Chaos Encryption Algorithm and System Design

The chaos system introduced in this paper is implemented by PCA neural networks algorithm proposed by Xu[7]. In [5], some chaotic characteristics of the algorithm is presented. The PCA algorithm can be described by

$$\omega(k+1) = \omega(k) + 2\eta[\omega(k) - \omega^3(k)], 0 < \eta < 1, \quad (1)$$

where η is a parameter and $\omega(k)$ is system output. If the initial value is set $\omega(0) = 0.5$, the system output corresponding to η is shown in Fig.1. In this figure, we can see that the system output will eventually converge to one point if parameter η is between 0 and 0.5; the system will converge to two points when η is between 0.5 and about 0.65. Obviously, if we increase η, the system will converge to more and more points, from 4, 8 to 2^n. When η is between 0.8 and 1, the system no longer comes to convergence and has a chaotic state.

978–1–4244–1674–5/08/$25.00 ©

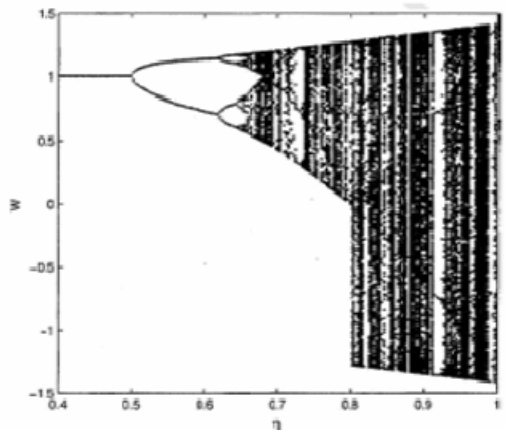

Fig. 1. System chaos phenomena map when $\omega(0) = 0.5$

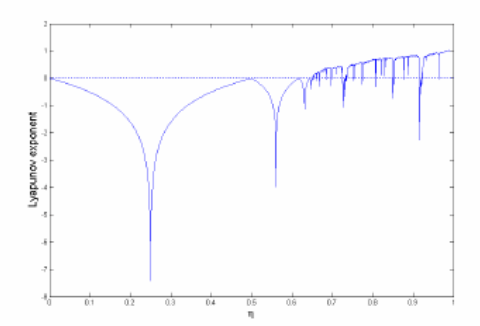

Fig. 2. The Lyapunov's exponents λ with different η when $\omega(0) = 0.5$

For a chaos dynamical system, the size of Lyapunov's exponents λ with the system is related to the degree of chaos. By analyzing Lyapunov's exponents λ, if η is selected as different values, the system shows different degree of chaos. Fig. 2 shows the Lyapunov's exponents λ with different η when $\omega(0) = 0.5$.

Based on the above analysis, the algorithm is designed as:

$$\begin{cases} \omega(k+1) = \omega(k) + 2\eta[\omega(k) - \omega^3(k)] \\ \omega(0) = 0.5, \eta = 0.95, \end{cases}$$

where the Lyapunov's exponent value $\lambda = 0.88006$.

A. The System Design

Chaotic systems can generate an unpredictable random orbit. It can be used as pseudo-random number generator to generate encryption keys and then make use of these keys encrypting. The flowchart of the Key Generator is shown in Fig.3.

One-time pad method is considered as the most secure encryption method. In this method, each data stream in the smallest units is corresponding to a random key. So that the encryption method is quite safe even under exhaustive attack. It is because that the key space is large enough to decode in a short time.

The system has been deployed by C++ Language. The interface of the chaotic encryption system is shown in Fig.4.

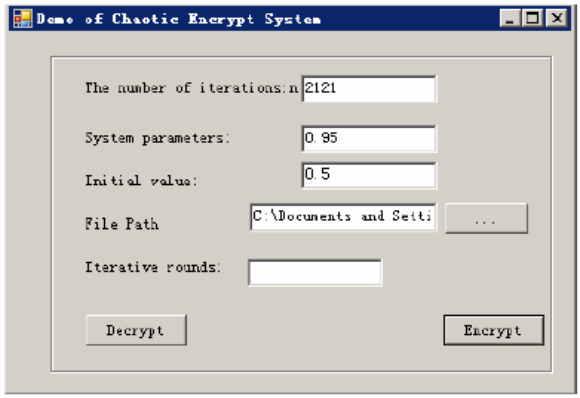

Fig. 4. The chaotic encryption system

B. The Encryption Algorithm

The description of the encryption algorithm is stated as follows,

(1) N_i is the byte array of the original data. The value of A is:

$$A = N_i(XOR)N_2(XOR)N_3...(XOR)N_i...(XOR)N_n \quad (2)$$

The value of n equals to the bytes number of the original data.

(2) Then computing by

$$N_i = N_i(XOR)A, i = 1 - n \quad (3)$$

(3) Setting parameter η, the initial ω, and the number of iteration ν to generate random private keys serial $S_i, i = 1-n$ by the chaotic system.

(4) Mapping the value of S_i to a range between 0 and 255, then encrypting data by calculating the bytes array of the encrypted data as follows

$$D_i = N_1(XOR)S_i, i = 1 - n$$

(5) In order to make the encrypted information more adaptable with actual requirements, we repeat step 3 to step 4 for M times. The repeat times M is decided by the encryption requirements. A larger M will achieve a better chaotic of original information, but it is more time-consuming.

C. The Decryption Algorithm

Decryption is an inverse operation to encryption. In decrypting, there exists four parameters to determine which are needed and dispensable.

The decryption algorithm is stated by,

(1) N_i is the byte array of the encrypted data. We calculate the value of A by

$$A = N_1(XOR)N_2(XOR)N_3...(XOR)N_i...(XOR)N_n \quad (4)$$

The size of n is equate to the numbers of bytes of the encrypted data.

(2) Computing:

$$N_i = N_i(XOR)A, i = 1 - n \quad (5)$$

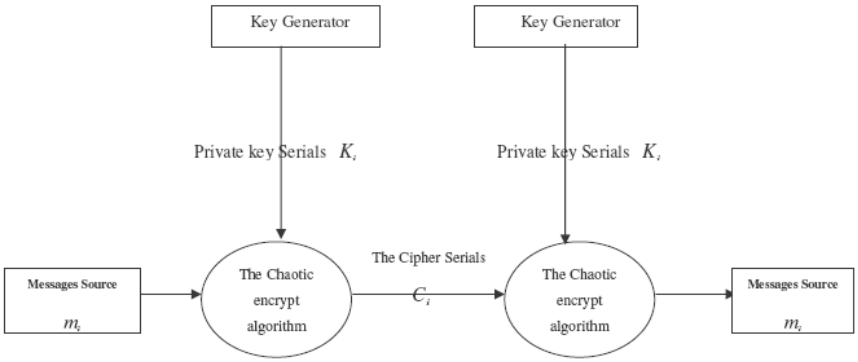

Fig. 3. The work flow of the encryption system

(3) Input correct system parameter η, the initial ω , the number of iteration ν and the repeat time M to make the chaotic system generate the same random private keys serial $S_i, i = 1 - n$ as the encryption procedure.

(4) Mapping the value of S_i to a range between 0 and 255, then decrypt data by calculating

$$D_i = N_1(XOR)S_i, i = 1 - n$$

Then, we get the decrypted data. D_i is the bytes array of the encrypted data.

In the following experiments, we can see that even the initial value or system parameters has very little changes, the decryption will fail.

III. Security analysis

A good encryption algorithm should not only have well confusing information, but also have certain ability to against all kinds of cryptanalytic attacks. In general, the attack methods includes statistical analysis attack, differential cryptanalysis, brute-force attacks etc. In this section we will analyze the security of the proposed encryption algorithm with three attacks, namely, statistical analysis, key space analysis and differential cryptanalysis. We use lina-image (Fig.5(a)) as the experimental data.

A. Statistical Analysis

In cryptography analysis, statistical analysis is a very common attack method. Therefore, an ideal encrypt algorithm should be robust against any statistical analysis. To prove the robustness of the proposed encrypt algorithm, we have performed cryptography analysis by calculating the histograms of the encrypt image.

An image-histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level[8]. Fig. 5(a) was the original data, Fig.5 (b) is the encrypt data. The two images have the same size of 512×512. Fig.5(c) and Fig.5 (d) are the histograms of Fig.5 (a) and Fig.5 (b). In Fig.5 (d), the maximum value is 1092, and the minimum is 969; the difference is so close that an attacker cannot get any information from the encrypt-image. Thus, the algorithm can prevent certain statistical analysis attack.

B. Key Space Analysis

For a good encrypt algorithm, the key space should be large enough to against the brute-force attack. It should be very sensitive to the private key. If we use slightly different secret keys in the proposed algorithm, it will give a quite different results. In order to test the sensitivity of the keys, we have done the following steps:

(1) We still take the standard Lina map (Fig.5(a)) as the plain-image.

(2) Encryption: We encrypt the original data (Fig.5(a)) with the parameters in Table I. Then we get the encrypted data (Fig.5(b)).

TABLE I
THE PARAMETERS OF SYSTEM FOR ENCRYPTING

items	value	description
n	2121	the number of iterations
ω_0	0.5	the initial value
M	2	repeated rounds
η	0.95	system parameter

(3) Decryption: We change the initial value $\omega(0)$ just for 10^{-7} . The other parameters of this system are unchanged. So we take the parameters in Table II into the system to decrypt Fig.5(b).

TABLE II
THE PARAMETERS OF SYSTEM FOR DECRYPTING

items	value	description
n	2121	the number of iterations
ω_0	$0.5 + 10^{-7}$	the initial value
M	2	repeated rounds
η	0.95	system parameter

Simulations show that even if its initial values are only changed for 10^{-7} level, the decryption will fail to decrypt correctly. Moreover, Fig.7 is the histogram of Fig.6(c) while Fig.5(c) is of original image. We can conclude that it cannot get any information about the original image from Fig.6(c).

Meanwhile, the algorithm is quite sensitive to its parameters and initial values so that it can generate enough large key spaces to resist all kinds of brute-force attacks.

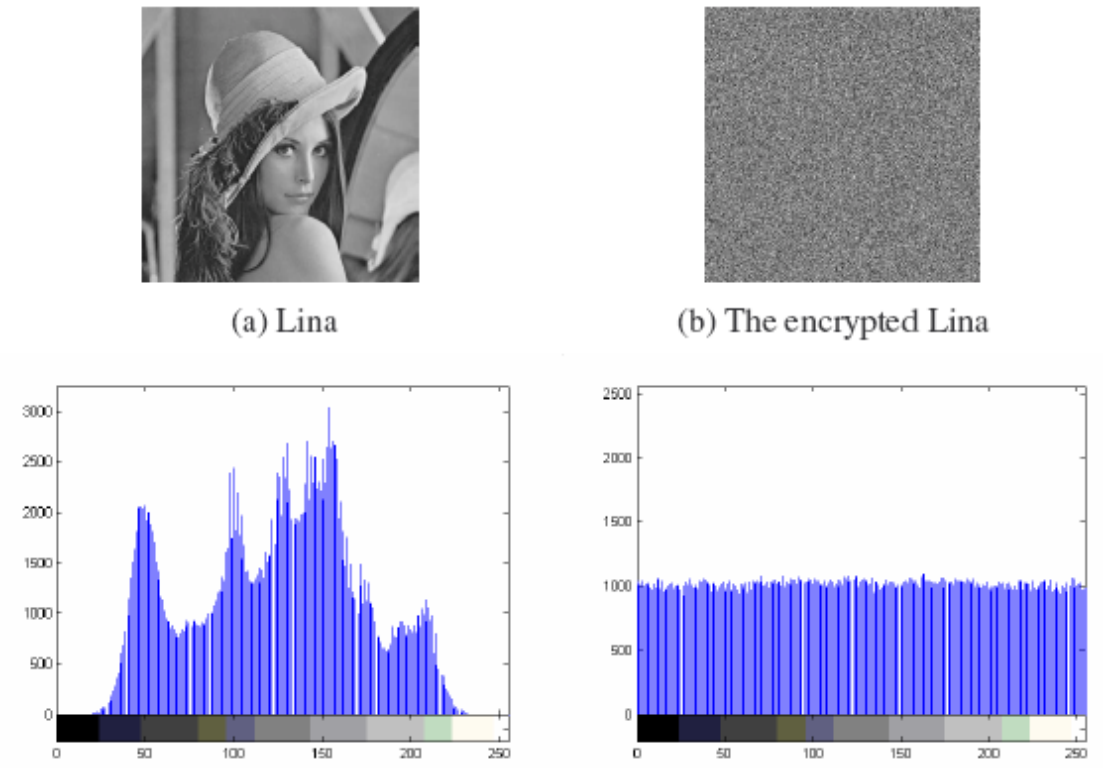

Fig. 5. Statistical analysis:(a)The original data. (b)The encrypted data using chaotic system. (c)The histogram of plain-image(a). (d)The histogram of encrypted data(b).

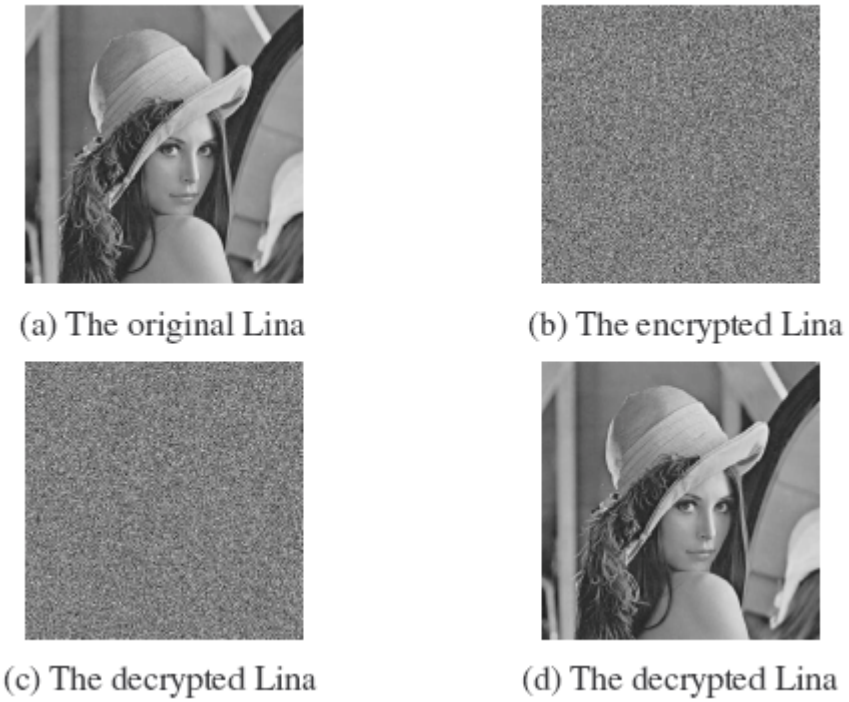

Fig. 6. Key Space analysis: (a)The original data. (b) The encrypted data using parameters $\omega(0) = 0.5$. (c)The decrypted data using parameters $\omega(0) = 0.5 + 10^{-7} = 0.5000001$. (d) The decrypted data using parameters $\omega(0) = 0.5$

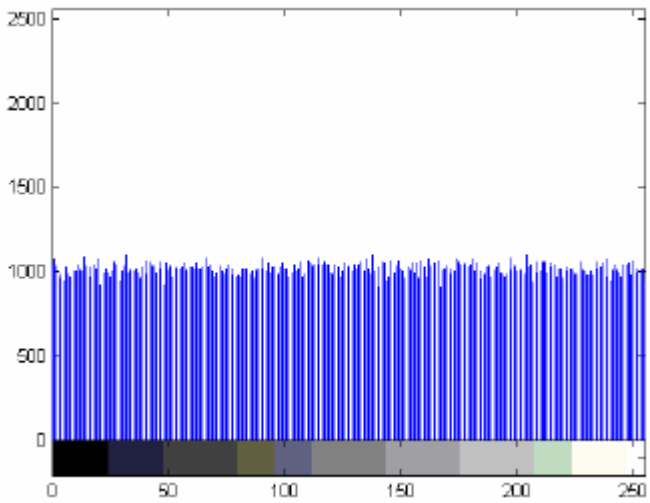

Fig. 7. Histogram of the decrypt image

C. Differential Cryptanalysis

We can measure the number of pixels change rate to test the influence of one-pixel change on the whole image encrypted by the proposed algorithm. The NPCR measures the percentage of different pixel numbers between the two images. We take two encrypted images C_1, C_2, whose corresponding the original have only one-pixel difference. Define a two-dimensional array D,having the same size as the image of C_1/C_2 ,the $D(i,j)$ is defined from the $C_1(i,j)$ and $C_2(i,j)$. If $C_1(i,j) = C_2(i,j)$,then $D(i,j) = 1$, otherwise $D(i,j) = 0$.NPCR is defined by the following formulas[6]:

$$NPCR = \frac{\sum_{ij} D(i,j)}{W \times H} \tag{6}$$

where the W and H are the width and height of C_1 or C_2 .We have take many tests about the one-pixel changed influence on a 256 gray-scale image(Fig.6)(a)of size 512×512 . And we obtained NPCR by using the encryption scheme and found that the NPCR is so small to limit to 0 values. The result shows that the encryption scheme is very sensitivity to small changes on the plaintext and it proved that the algorithm proposed has a good ability to anti differential attack.

IV. Conclusions

In this paper, we proposed an encryption algorithm by using PCA neural networks. We applied it to a chaotic encryption system. The effect of encryption and the security of the algorithm are also analyzed. The experiments show that the chaotic encryption system using PCA neural networks is able to resist many common attacks.

Acknowledgment

This work was supported by Chinese 863 High-Tech Program Under Grant 2007AA01Z321.

References

[1] Liuz P.L,de O.,M.Sobottka,Cryptography with chaotic mixing, *Chaos, Solitons & Fractals*, vol. 35, pp. 466-471, 2008
[2] S.Behnia,A.Akhshani,H.Mahmodi,A novel algorithm for image encryption based on mixture of chaotic maps, *Chaos, Solitons & Fractals*, vol. 35, pp. 408-419, 2008
[3] C.E.Shannon,Bell Syst.Tech.J.28(1949) 656.
[4] C.Y Chee,Daolin Xu,Chaotic encryption using discrete-time synchronous chaos. Physics Letters A 348 (2006) 284-292
[5] J.C Lv , Z. Yi, Stability and Chaos of LMSER PCA learning algorithm. *Chaos, Solitons & Fractals*, vol. 32, pp. 1440-1447, 2007
[6] G. Chen H,Y.Mao,C.K.Chui, A symmetric image encryption based on 3D chaotic maps,*Chaos, Solitons & Fractals*,21 ,(2004),749-761
[7] L. Xu, Least Mean Square Error Reconstruction Principle for for Self-Organizing Neural-Nets, neural Networks, Vol. 6, pp. 627-648, 1993.
[8] N.K.Pareek,Vinod Patidar and K.K,Sud,Image encryption using chaotic logistic map,Image and Vision Computing ,24 (2006) 926-934.