Neurocomputing 72 (2009) 1296–1301

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Letters

A block cipher based on chaotic neural networks

Shiguo Lian

France Telecom R&D (Orange Labs) Beijing, 2 Science Institute South Road, Haidian District, Beijing 100080, PR China

ARTICLE INFO

Article history:
Received 6 November 2007
Received in revised form
7 November 2008
Accepted 7 November 2008
Communicated by W.L. Dunin-Barkowski
Available online 18 November 2008

Keywords:
Neural network
Chaos
Security
Block cipher
Image encryption

ABSTRACT

In this paper, the neural network composed of a chaotic neuron layer and a linear neuron layer is used to construct a block cipher that transforms the data from the plaintext form into the unintelligible form under the control of the key. Among them, the chaotic neuron layer realizes data diffusion, the linear neuron layer realizes data confusion, and the two layers are repeated for several times to strengthen the cipher. The decryption process is symmetric to the encryption process. Theoretical analysis and experimental results show that the block cipher has good computing security and is more suitable for image encryption. It is expected to attract more researchers in this field.

1. Introduction

Neural networks can be used to design data protection schemes because of its complicated and time-varying structures [1]. For the property of initial-value sensitivity, ergodicity or random similarity, chaos is also used in data protection [2]. As a combination of neural networks and chaos, chaotic neural networks (CNN) are expected to be more suitable for data encryption. For example, it is reported [3] that faster synchronization can be obtained by jointing neural network's synchronization and chaos' synchronization.

Till now, various CNN-based encryption algorithms have been reported, which can be classified into two classes, i.e., stream cipher and block cipher. The first one uses neural networks to produce pseudorandom sequences [4,5]. These stream ciphers' security depends on the sequences' randomness. However, the real random sequence is still difficult to generate in practice, and its random property should be analyzed in detail, which restricts their applications. The second one makes use of CNN properties to encrypt plaintext block by block [6,7]. Generally, block ciphers' security depends on their computing security [8] (it is not practical from the view of computing complexity for attackers to break the cryptosystem in condition of not knowing the key) that is determined by the confusion and diffusion criteria [9]. According to the criteria, some of the CNN-based ciphers are not secure enough. For example, the block ciphers [6,7] have only good diffusion property but no confusion property since the encryption or decryption process consists of only linear functions. This property makes it vulnerable to general attacks [10], which restrict its practical applications.

In this paper, we construct a CNN-based block cipher with good computing security. Intuitively, CNN with multiple-input and multiple-output is more suitable for block cipher construction. Firstly, the multiple-input or multiple-output composed of many bits act as the plaintext or ciphertext, respectively. Secondly, the neural network realizes confusion and diffusion functionalities. Using the chaotic map as the transfer function of the neural layer, the CNN is designed, which confuses and diffuses large number of adjacent image pixels in the same encryption round, which is presented in Section 2. Compared with existing ciphers, the proposed cipher aims to reduce the redundancy among the adjacent image pixels and obtain higher perceptual security. The analyses and experiments are done to show its competences in Section 3.

2. The proposed block cipher

The proposed block cipher is composed of diffusion process and confusion process. As shown in Fig. 1, the diffusion process and confusion process is repeated for t times in order to improve the encryption strength. Among them, the diffusion process is implemented by a chaotic neuron layer, and the confusion process is implemented by a linear neuron layer. As shown in Fig. 2, in encryption, the chaotic neuron layer and linear neuron layer are repeated for t times in order to enhance the cryptosystem's security. From the key, the parameters are generated and used to

E-mail addresses: shiguo.lian@orange-ftgroup.com, sg_lian@163.com (S. Lian).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.11.005

control the neuron layers. The decryption process is symmetric to the encryption process. Both of them are defined as

$$\begin{cases} C = E(P,K), \\ P = D(C,K). \end{cases} \quad (1)$$

Here, P, K, C, $E()$ and $D()$ are the plaintext, key, ciphertext, encryption function and decryption function, respectively.

2.1. The encryption process

The encryption process is composed of t times of iteration of a chaotic neuron layer and a linear neuron layer, which realize data diffusion and confusion, respectively. As shown in Fig. 2(a), the encryption process is described as

$$\begin{cases} C_i = g(W_{ci}M_i + B_{ci}) = g(W_{ci}f(W_{di}P_i + B_{di}, A_i)) \\ P_i = C_{i-1} \end{cases}, \quad (2)$$

where P_i, K_i and C_i are the t-th plaintext, key and ciphertext in the i-th iteration, respectively. W_{di}, B_{di}, $f()$ and A_i are the chaotic neuron layer's weight, bias, transfer function and chaotic parameter, respectively, and W_{ci}, B_{ci} and $g()$ are the linear neuron layer's weight, bias and transfer function, respectively.

Firstly, the chaotic neuron layer is defined as

$$\begin{aligned} M_i &= f(W_{di}P_i + B_{di}, A_i) \\ &= f\left(\begin{bmatrix} w^{di}_{0,0} & w^{di}_{0,1} & \cdots & w^{di}_{0,n-1} \\ w^{di}_{1,0} & w^{di}_{1,1} & \cdots & w^{di}_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ w^{di}_{n-1,0} & w^{di}_{n-1,1} & \cdots & w^{di}_{n-1,n-1} \end{bmatrix} \begin{bmatrix} p_{i,0} \\ p_{i,1} \\ \vdots \\ p_{i,n-1} \end{bmatrix} + \begin{bmatrix} b^{d}_{i,0} \\ b^{d}_{i,1} \\ \vdots \\ b^{d}_{i,n-1} \end{bmatrix}, A_i \right) \\ &= f\left(\begin{bmatrix} m'_{i,0} \\ m'_{i,1} \\ \vdots \\ m'_{i,n-1} \end{bmatrix}, A_i \right) = \begin{bmatrix} f^z_{Tent}(a_{i,0}, m'_{i,0}) \\ f^z_{Tent}(a_{i,1}, m'_{i,1}) \\ \vdots \\ f^z_{Tent}(a_{i,n-1}, m'_{i,n-1}) \end{bmatrix} = \begin{bmatrix} m_{i,0} \\ m_{i,1} \\ \vdots \\ m_{i,n-1} \end{bmatrix}, \end{aligned} \quad (3)$$

where $P_i = [p_{i,0}, p_{i,1}, p_{i,2}, \ldots p_{i,n-1}]^T$ $(i = 0,1,\ldots,t-1)$ is the plaintext, $0 \leqslant p_{i,j} < S$ (S is a positive integer, and $j = 0,1,\ldots,n-1$), n is the plaintext's length, W_{di} and $B_{di} = [b_{i,0}, b_{i,1}, b_{i,2},\ldots,b_{i,n-1}]^T$ $(0 \leqslant b_{i,j} < S$, $j = 0,1,\ldots,n-1)$ are the weight and bias of the neuron layer, $f()$ is an reversible chaotic dynamic function, and $A_i = [a_{i,0}, a_{i,1},\ldots,a_{i,n-1}]$ $(1 \leqslant a_{i,j} \leqslant S$, $j = 0,1,\ldots,n-1)$ is the control parameter of the chaotic map $f()$. Among them, W_{di} is a reversible integer mapping [11] that will be described in Section 2.3. The $f()$ function is composed of z $(z \geqslant 10)$ times of iteration of discrete tent map [12]. The discrete tent map is

$$f_{Tent}(a,x) = \begin{cases} \left\lceil \frac{S}{a}x \right\rceil, & 1 \leqslant x \leqslant a \\ \left\lfloor \frac{S}{S-a}(S-x) \right\rfloor + 1, & a < x \leqslant S \end{cases}, \quad (4)$$

where a $(a \in [1,S])$ is an integer determined by user key K, and, $\lfloor x \rfloor$ and $\lceil x \rceil$ denote floor and ceiling of x, respectively. This discrete tent map is a one-to-one mapping.

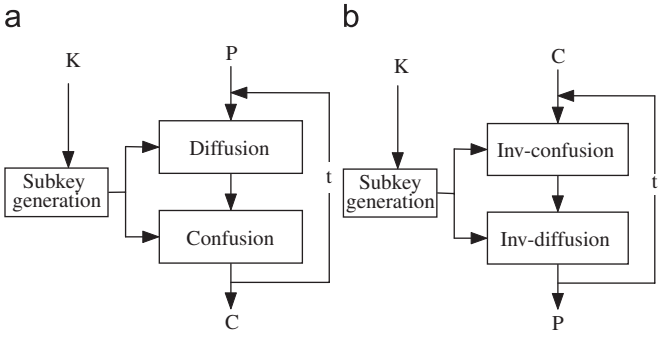

Fig. 1. General architecture of the proposed block cipher. (a) Encryption and (b) decryption.

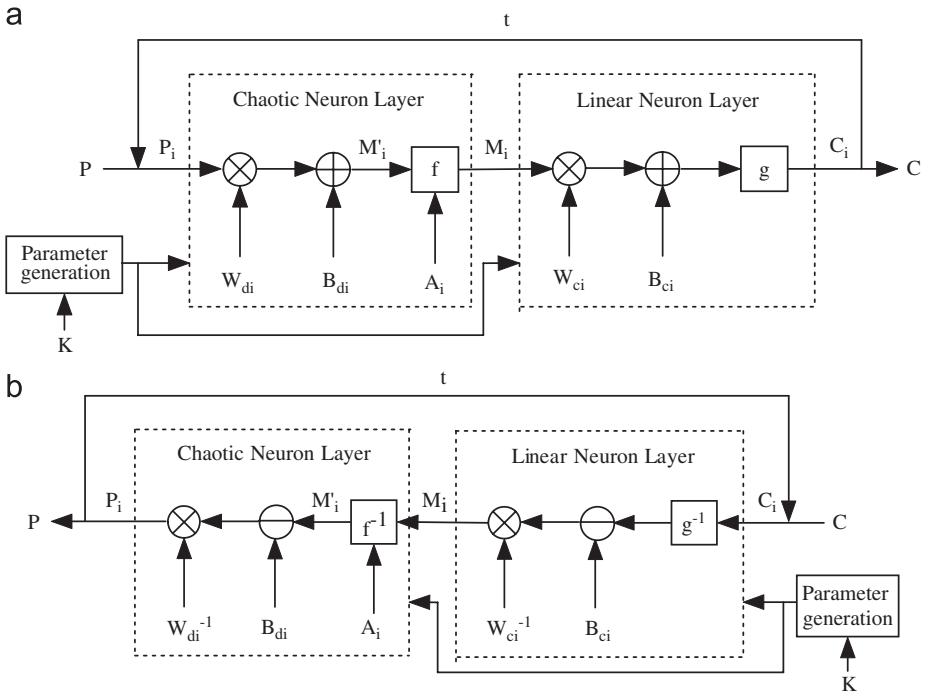

Fig. 2. The neuron layers in the proposed block cipher. (a) The encryption process and (b) the decryption process.