Cryptanalysis of a Chaotic Neural Network Based Multimedia Encryption Scheme[⋆]

Chengqing Li[1], Shujun Li[2a⋆⋆], Dan Zhang[3], and Guanrong Chen[2b]

[1] Department of Mathematics, Zhejiang University, Hangzhou 310027, Zhejiang, China, e-mail: swiftsheep@hotmail.com
[2] Department of Electronic Engineering, City University of Hong Kong, Kowloon Toon, Hong Kong, China, e-mails: hooklee@mail.com[2a], eegchen@cityu.edu.hk[2b]
[3] College of Computer Science, Zhejiang University, Hangzhou 310027, Zhejiang, China, e-mail: zhangdan@etang.com

Abstract. Recently, Yen and Guo proposed a chaotic neural network (CNN) for signal encryption, which was suggested as a solution for protection of digital images and videos. The present paper evaluates the security of this CNN-based encryption scheme, and points out that it is not secure from the cryptographical point of view: 1) it can be easily broken by known/chosen-plaintext attacks; 2) its security against the brute-force attack was much over-estimated. Some experiments are shown to support the results given in this paper. It is also discussed how to improve the encryption scheme.

1 Introduction

In the digital world today, the security of multimedia data (such as digital speeches, images, and videos) becomes more and more important since the communications of such digital signals over open networks occur more and more frequently. Also, special and reliable security in storage and transmission of multimedia products is needed in many real applications, such as pay-TV, medical imaging systems, military image/database communications and confidential video conferences, etc. To fulfill such a need, many encryption schemes have been proposed as possible solutions [1, Sec. 4.3], among which some are based on chaotic systems [1, Sec. 4.4]. Meanwhile, cryptanalysis work has also been developed, which reveal that some proposed multimedia encryption schemes have been known to be insecure.

From 1998, Yen et al. proposed a number of chaos-based multimedia encryption schemes [1, Sec. 4.4.3], but some of them have been successfully broken by Li et al. [2–6]. This paper analyzes the security of a class of encryption schemes proposed by Yen et al. in [7–9], which have not yet been cryptanalyzed before.

[⋆] This paper has been published in *Advances in Multimedia Information Processing - PCM 2004 Proceedings, Part III*, volume 3333 of *Lecture Notes in Computer Science*, pp. 418-425, 2004, Springer-Verlag Berlin Heidelberg.

[⋆⋆] The corresponding author, web site: http://www.hooklee.com.

In a recent paper [10], this class of encryption schemes were simply extended to arbitrary block size without influencing the security and then applied for JPEG2000 image encryption.

The studied encryption scheme here is a stream cipher based on a chaotic neural network (CNN), which is designed to encrypt 1-D signals and is simply extended to encrypt 2-D digital images and 3-D videos. This paper evaluates the security of the CNN-based scheme and points out two security problems: 1) it can be easily broken by the known/chosen-plaintext attacks with only one known/chosen plaintext; 2) its security against the brute-force attack was much over-estimated.

The rest of the present paper is organized as follows. In Sec. 2, a brief introduction of the CNN-based encryption scheme is given. The cryptanalytic studies and some experimental results are given in Sec. 3. Section 4 briefly discusses how to improve the security of the studied encryption scheme, and the last section concludes the paper.

2 The CNN-Based Scheme for Signal Encryption

In the following, the concerned encryption scheme is simply referred to as CNN.

Assuming that $\{f(n)\}_{n=0}^{M-1}$ is a 1-D signal for encryption, the encryption procedure of CNN can be briefly depicted as follows:

- The *chaotic Logistic map* $f(x) = \mu x(1-x)$ is used, where μ is the control parameter [11].
- *The secret key* is the control parameter μ and the initial point $x(0)$ of the Logistic map, which are all L-bit binary decimals.
- *The initialization procedure*: under L-bit finite computing precision, run the Logistic map from $x(0)$ to get a chaotic sequence $\{x(i)\}_{i=0}^{\lceil 8M/K\rceil-1}$, and extract K bits below the decimal dot of each chaotic state[1] to generate a chaotic bit sequences $\{b(i)\}_{i=0}^{8M-1}$, where $x(i) = 0.b(Ki+0)\cdots b(Ki+K-1)\cdots$.
- *The encryption procedure*: For the n-th plain-element $f(n) = \sum_{i=0}^{7} d_i(n)\times 2^i$, the corresponding cipher-element $f'(n) = \sum_{i=0}^{7} d_i'(n)\times 2^i$ is determined by the following process:
    - for $i = 0 \sim 7$ and $j = 0 \sim 7$, 64 weights w_{ji} are calculated as follows: if $i = j$, $w_{ji} = 0$; else $w_{ji} = 1 - 2b(8n+i) = \begin{cases} 1, & b(8n+i) = 0, \\ -1, & b(8n+i) = 1; \end{cases}$
    - for $i = 0 \sim 7$, 8 biases θ_i are calculated as follows:

$$\theta_i = \frac{2b(8n+i)-1}{2} = \begin{cases} -1/2, & b(8n+i) = 0, \\ 1/2, & b(8n+i) = 1; \end{cases}$$

[1] In real implementations of CNN, the K bits can be extracted from the direct multiplication result $\mu x(i-1)(1-x(i-1))$, before $x(i)$ is obtained by quantizing the value. As a result, it is possible that $K > L$. For example, in [9], $K = 32 > L = 17$.

- the i-th cipher-bit $d_i'(n)$ is calculated as follows:

$$d_i'(n) = \operatorname{sign}\left(\sum\nolimits_{j=0}^{7} w_{ji} \times d_i(n) + \theta_i\right), \tag{1}$$

  where $\operatorname{sign}(\cdot)$ denotes the sign function, i.e., $\operatorname{sign}(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$

– *The decryption procedure* is the same as the above one.

The above encryption procedure looks very complicated, however, actually it can be simplified to be a much more precise form. Observing the proofs of Proposition 1 in [7, 8] and Lemma 1 in [9], one can see the following fact:

$$d_i'(n) = \begin{cases} 0, & \text{if } d_i(n) = 0 \text{ and } b(8n+i) = 0, \\ 1, & \text{if } d_i(n) = 1 \text{ and } b(8n+i) = 0, \\ 1, & \text{if } d_i(n) = 0 \text{ and } b(8n+i) = 1, \\ 0, & \text{if } d_i(n) = 1 \text{ and } b(8n+i) = 1, \end{cases} \tag{2}$$

which means that

$$d_i'(n) = d_i(n) \oplus b(8n+i), \tag{3}$$

where $\oplus$ denotes the XOR operation.

Obviously, CNN is a stream cipher encrypting the plain-signal bit by bit, where the key stream for masking is the chaotic bit sequence $\{b(i)\}$.

3 Cryptanalysis of the CNN-Based Encryption Scheme

3.1 Brute-Force Attacks

In [7–9], it was claimed that the computing complexity of a brute-force attack to CNN is $O\left(2^{8M}\right)$, since there are $8M$ bits in $\{b(i)\}_{i=0}^{8M-1}$ (which is unknown to the attacker). However, this statement is not true due to the following fact: the $8M$ bits are uniquely determined by the secret key, i.e., the control parameter μ and the initial condition $x(0)$, which have only $2L$ secret bits. This means that there are only 2^{2L} different chaotic bit sequences.

Now, let us see what is the real complexity of a brute-force attack. For each guessed value of $x(0)$ and μ, about $8M/K$ chaotic iterations and $8M$ XOR operations are needed for verification. Assuming that each L-bit digital multiplication needs L times of additions, then each chaotic iteration needs $2L+1$ times of additions. Therefore, the complexity of a brute-force attack to CNN will be $O\left(2^{2L} \times \left(\frac{8M(2L+1)}{K} + 8M\right)\right) = O\left(2^{2L}M\right)$, which is much smaller than 2^{8M} when M is not too small. What's more, considering the fact that the Logistic map can exhibit strong chaotic behavior only when μ is close to 4 [11], the complexity should be even smaller than $O\left(2^{2L}M\right)$.

The above analysis shows that the security of CNN was much over-estimated by the authors, even under the simplest attack. Because of the rapid progress of digital computer and distributed computing techniques, the complexity not lower than $O\left(2^{128}\right)$ is required for a cryptographically strong cipher [12]. To achieve such a security level, $L \geq 64$ is required. As a comparison, $L = 8$ in [8] and $L = 17$ in [9], which are both too small[2].

3.2 Known/Chosen-Plaintext Attacks

In known-plaintext or chosen-plaintext attacking scenarios, CNN can be broken with only one known/chosen plaintext $\{f(n)\}_{n=0}^{M-1}$ and its corresponding ciphertext $\{f'(n)\}_{n=0}^{M-1}$, with a complexity that is smaller than the complexity of a brute-force attack.

From Eq. (3), one can get $b(8n+i) = g_i(n) \oplus g'_i(n)$. That is, an attacker can successfully reconstruct the chaotic bit sequence $\{b(i)\}_{i=0}^{8M-1}$ by simply XORing $\{f(n)\}_{n=0}^{M-1}$ and $\{f'(n)\}_{n=0}^{M-1}$ bit by bit. Assuming $\{f_m(n) = f(n) \oplus f'(n)\}_{n=0}^{M-1}$, one has $f_m(n) = 0.b(8n+0) \cdots b(8n+7)$. Without deriving the secret key $(\mu, x(0))$, given any ciphertext g' encrypted with the same secret key, the attacker can use f_m to decrypt the M leading bytes of the corresponding plaintext g: $n = 0 \sim M-1$, $g(n) = g'(n) \oplus f_m(n)$. Here, we call f_m the *mask signal* (or the *mask image* when CNN is used to encrypt digital images), since the plaintext can be decrypted by using f_m to "mask" (i.e., XOR) the ciphertext[3].

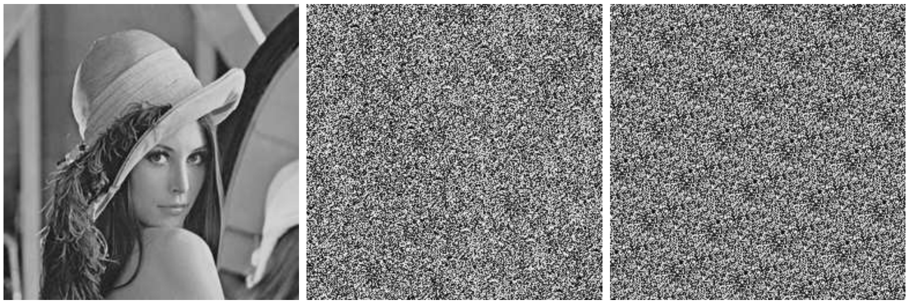
To demonstrate the above attack, with the parameters $L = 17, K = 32$ [9] and the secret key $\mu = 3.946869$, $x(0) = 0.256966$, some experiments are given for the encryption of digital images. In Fig. 1, a 256×256 known/chosen plain-image "Lenna", its corresponding cipher-image, and the mask image $f_m = f \oplus f'$ are shown. If another plain-image "Babarra" (of size 256×256) is encrypted with the same key, it can be broken with the mask image f_m derived from "Lenna" as shown in Fig. 2. For a larger plain-image "Peppers" (of size 384×384), the 256×256 leading pixels can be successfully broken with f_m as shown in Fig. 3.

From the above experiments, one can see that the breaking performance of known/chosen-plaintext attacks based on f_m is limited. Fortunately, from the reconstructed bit sequence $\{b(i)\}_{i=0}^{8M-1}$, it is easy for an attacker to derive the values of μ and $x(0)$, and then to completely break CNN. Even when only part of a plaintext $f(n_1) \sim f(n_2)$ is known to the attacker, he can still derive the values of μ and a chaotic state $x(i)$, which can be used to calculate all following chaotic states, i.e., all following chaotic bits $\{b(i)\}_{i=8n_2}^{\infty}$. In this case, all plain-pixels after the n_1-th position can be broken. In the following, let us discuss how to derive chaotic states and the value of μ.

Firstly, let us see how a chaotic state $x(i)$ is derived. Recall the generation procedure of $\{b(i)\}_{i=0}^{8M-1}$. It is easy to reconstruct a K-bit approximate of the chaotic sequence by dividing $\{b(i)\}_{i=0}^{8M-1}$ into K-bit segments: $\{\widetilde{x}(i)\}_{i=0}^{\lceil 8M/K \rceil - 1}$,

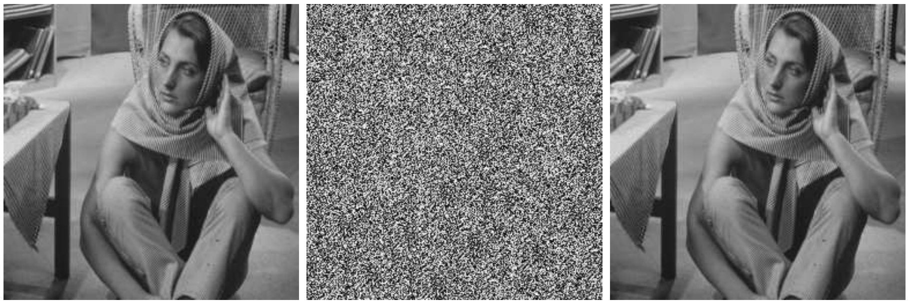
[2] In [7], the value of L is not explicitly mentioned. Since [7] is an initial version of [8], it is reasonable to assume $L = 8$.

[3] In fact, it is a common defect of most stream ciphers [12].



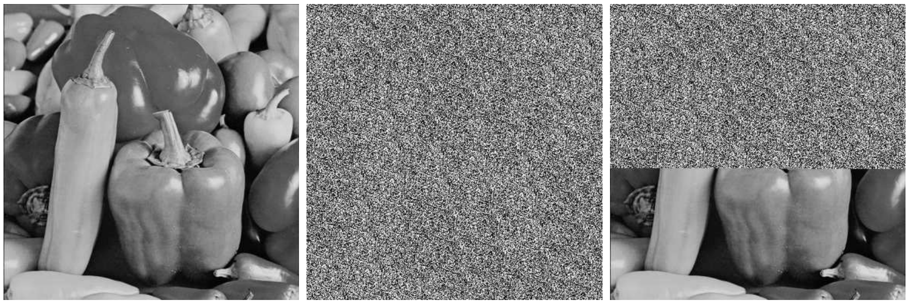
a) The plain-image f b) The cipher-image f' c) The mask image f_m

Fig. 1. One known/chosen plain-image "Lenna" (256×256), its corresponding cipher-image, and the mask image $f_m = f \oplus f'$



a) The plain-image "Babarra" b) The encrypted "Babarra" c) The recovered "Babarra" with f_m

Fig. 2. Decrypt a plain-image "Babarra" (256×256) with f_m shown in Fig. 1c



a) The plain-image "Peppers" b) The encrypted "Peppers" c) The recovered "Peppers" with f_m

Fig. 3. Decrypt a plain-image "Peppers" (384×384) with f_m shown in Fig. 1c

where $\widetilde{x}(i) = 0.b(Ki+0)\cdots b(Ki+K-1)$ and

$$|\Delta x(i)| = |\widetilde{x}(i) - x(i)| \le 0.\overbrace{0\cdots0}^{K}\overbrace{1\cdots1}^{L-K} = \sum_{j=K+1}^{L} 2^{-j} < 2^{-K}. \tag{4}$$

Apparently, when $L \leq K$, $\widetilde{x}(i) = x(i)$; when $L > K$, the exact value of each chaotic state $x(i)$ can be derived by exhaustively guessing the $L - K$ unknown bits, and the guess complexity is $O\left(2^{L-K}\right)$.

Once two consecutive chaotic states $x(i)$ and $x(i+1)$ are derived, the estimated value of μ can be calculated to be $\tilde{\mu} = \frac{x(i+1)}{x(i)\cdot(1-x(i))}$. Due to the influence of quantization errors existing in forward chaotic iterations, in general $\tilde{\mu} \neq \mu$. When the difference between $\tilde{\mu}$ and μ is sufficiently small, it is possible to exhaustively search the neighborhood of $\tilde{\mu}$ to find the accurate value of μ with a sufficiently small complexity. In the following, we will show how to get a $\tilde{\mu}$ close enough to μ, and estimate the search complexity of the accurate value of μ.

Apparently, the estimation error $\Delta\mu = \tilde{\mu} - \mu$ is caused by the quantization error $\Delta x(i+1)$ generated in the forward chaotic iteration $x(i+1) = \mu \cdot x(i) \cdot (1 - x(i))$. In one L-bit digital multiplication, the quantization error does not exceed 2^{-L} for the floor or ceiling quantization function, and does not exceed $2^{-(L+1)}$ for the round quantization function. Considering there are two L-bit digital multiplications in each forward chaotic iteration, one has

$$\begin{aligned}\bar{x}(i+1) &= (\mu \cdot x(i) + \Delta_1 x(i+1)) \cdot (1 - x(i)) + \Delta_2 x(i+1)\\ &= \mu \cdot x(i) \cdot (1 - x(i)) + \Delta_1 x(i+1) \cdot (1 - x(i)) + \Delta_2 x(i+1)\\ &= x(i+1) + \Delta x(i+1),\end{aligned}$$

where $\bar{x}(i+1)$ denotes the real value of $x(i+1)$ and $\Delta x(i+1) = \Delta_1 x(i+1) \cdot (1 - x(i)) + \Delta_2 x(i+1)$. Then, one can get $|\Delta x(i+1)| \leq |\Delta_1 x(i+1)| + |\Delta_2 x(i+1)| < 2^{-L} + 2^{-L} = 2^{-(L-1)}$, and get the quantization error $|\Delta\mu|$ as follows:

$$\begin{aligned}|\Delta\mu| &= \left|\frac{\Delta x(i+1)}{x(i)\cdot(1-x(i))}\right| = \left|\frac{\Delta x(i+1)}{x(i+1)} \cdot \frac{x(i+1)}{x(i)\cdot(1-x(i))}\right|\\ &= \frac{|\Delta x(i+1)|}{x(i+1)} \cdot \mu < \frac{2^{-(L-1)}}{x(i+1)} \cdot 4 = \frac{1}{2^{L-3}\cdot x(i+1)}.\end{aligned} \tag{5}$$

When $x(i+1) \geq 2^{-n}$ ($n = 1 \sim L$), $|\Delta\mu| < \frac{1}{2^{L-3}\cdot x(i+1)} \leq \frac{1}{2^{L-3}\cdot 2^{-n}} = 2^{n+3} \times 2^{-L}$, which means the size of the neighborhood of $\tilde{\mu}$ for exhaustive search is 2^{n+3}. To minimize the search complexity in real attacks, $x(i+1) \geq 0.5$ is suggested to derive μ, which occurs with a probability of 0.5. In this case, $n = 1$ and the size of the searched neighborhood is only $2^{3+1} = 16$.

With the mask image f_m derived from the known plain-image "Lenna" (of size 256×256) shown in Fig. 1a, the values of $x(0)$ and μ are calculated following the above procedure to completely decrypt the larger plain-image "Peppers" (of size 384×384). The decryption result is given in Fig. 4.

Finally, it deserves being mentioned that even without deriving the secret key there is another way based on a mask signal f_m to decrypt any plaintext of arbitrary size. It is due to the following fact: for a digital chaotic system implemented in L-bit finite computing precision, each chaotic orbit will lead to a cycle whose length is smaller than 2^L (and generally much smaller than 2^L, see [4, Sec. 2.5]). For the implementation of CNN in [9], $L = 17$, $K = 32$. Thus,

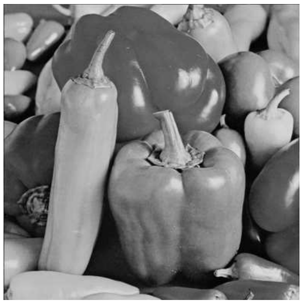
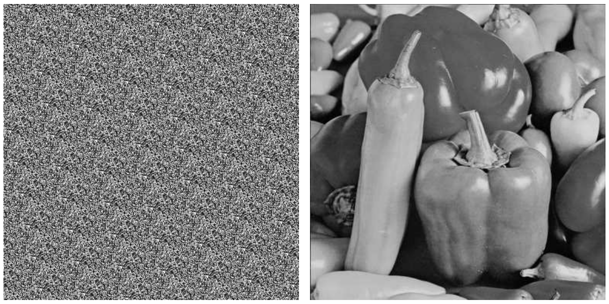


Fig. 4. The decrypted "Peppers" (384×384) with the secret key derived from f_m shown in Fig. 1c



a) The extended mask image f_m^*

b) The recovered "Peppers" with f_m^*

Fig. 5. Decrypt "Peppers" (384×384) with f_m^* extended from f_m shown in Fig. 1c

the cycle length of each chaotic orbit will be much smaller than 2^{17} in most cases. Such a length is not sufficiently large in comparison with the size of many plaintexts, especially for digital images and videos. For example, a 256×256 image corresponds to a chaotic orbit $\{x(i)\}$ whose length is $8 \times 256 \times 256/32 = 2^{14}$. For almost every value of μ and $x(0)$, the cycle length of $\{x(i)\}$ is even much smaller than 2^{14}, which means that there exists an visible repeated pattern in $\{x(i)\}$. Carefully observing the mask image f_m shown in Fig. 1c, one can easily find such a repeated pattern. Then, it is easy to get the cycle of f_m, and to extend it to arbitrary sizes by appending more cycles at the end of the original mask signal. This means that any ciphertext can be decrypted with a mask signal f_m^* extended from the mask image f_m. Using such a method, the larger plain-image "Peppers" is completely decrypted as shown in Fig. 5.

4 Improving the CNN-Based Encryption Scheme

The simplest way to improve the original CNN is to make L sufficiently large so as to ensure the complexity of the brute-force attack cryptographically large. In addition, to make the complexity of guessing the $L-K$ unknown bits of each chaotic state cryptographically large, $L-K$ should also be sufficiently large. To be practical, $(L,K) = (64,8)$ is suggested. In this case, the complexity to get the value of $x(0)$ is $O\left(2^{L-K}\right) = O\left(2^{56}\right)$, and the complexity to get the value of μ (i.e., to get two consecutive chaotic states) is $O\left(2^{2(L-K)}\right) = O\left(2^{112}\right)$. Such a complexity is sufficiently large to make both the brute-force attack and the attack of deriving the secret key from f_m impossible in practice.

However, because CNN is a stream cipher, making $L-K$ sufficiently large cannot enhance the security against the known/chosen-plaintext attacks based on the mask signal f_m. To resist such attacks, a substitution encryption part should be used to make CNN a product cipher. Note that the security of the modified CNN is ensured by the new substitution part, not the CNN itself.

So, essentially speaking, the CNN cannot be enhanced to resist known/chosen-plaintext attacks.

5 Conclusion

In this paper, the security of a chaotic signal encryption scheme called CNN [7–10] has been investigated and it is found that the encryption scheme is not secure from the cryptographical point of view. Both theoretical and experimental analyses show the feasibility of the proposed known/chosen-plaintext attacks of breaking CNN. Also, it is pointed out that the security of CNN against brute-force attacks was much over-estimated. Some possible methods to enhance the security of CNN are also discussed, but its insecurity against the known/chosen-plaintext attacks cannot be essentially improved. As a result, CNN is not suggested in applications requiring a high level of security.

Acknowledgement

This research was supported by the National Natural Science Foundation, China, under grant no. 60202002, and by the Applied R&D Centers of the City University of Hong Kong under grants no. 9410011 and no. 9620004.

References

1. Li, S., Chen, G., Zheng, X.: Chaos-based encryption for digital images and videos. In Furht, B., Kirovski, D., eds.: Multimedia Security Handbook. CRC Press (2004) preprint available at http://www.hooklee.com/pub.html.
2. Li, S., Zheng, X.: Cryptanalysis of a chaotic image encryption method. In: Proc. IEEE Int. Symposium on Circuits and Systems. Volume II. (2002) 708–711
3. Li, S., Zheng, X.: On the security of an image encryption method. In: Proc. IEEE Int. Conference on Image Processing. Volume 2. (2002) 925–928
4. Li, S.: Analyses and New Designs of Digital Chaotic Ciphers. PhD thesis, School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China (2003) Available online at http://www.hooklee.com/pub.html.
5. Li, S., Li, C., Chen, G., Mou, X.: Cryptanalysis of the RCES/RSES image encryption scheme. Cryptology ePrint Archive: Report 2004/376, available online at http://eprint.iacr.org/2004/376 (2004)
6. Li, S., Li, C., Chen, G., Zhang, D., Bourbakis, N.G.: A general cryptanalysis of permutation-only multimedia encryption algorithms. Cryptology ePrint Archive: Report 2004/374, available online at http://eprint.iacr.org/2004/374 (2004)
7. Yen, J.C., Guo, J.I.: A chaotic neural network for signal encryption/decryption and its VLSI architecture. In: Proc. 10th VLSI Design/CAD Symposium. (1999) 319–322
8. Su, S., Lin, A., Yen, J.C.: Design and realization of a new chaotic neural encryption/decryption network. In: Proc. IEEE Asia-Pacific Conference on Circuits and Systems. (2000) 335–338

9. Yen, J.C., Guo, J.I.: The design and realization of a chaotic neural signal security system. Pattern Recognition and Image Analysis (Advances in Mathematical Theory and Applications) **12** (2002) 70–79
10. Lian, S., Chen, G., Cheung, A., Wang, Z.: A chaotic-neural-network-based encryption algorithm for JPEG2000 encoded images. In: Advances in Neural Networks - ISNN 2004: International Symposium on Neural Networks Proceedings, Part II. Volume 3174 of Lecture Notes in Computer Science. (2004) 627–632
11. Hao Bai-Lin: Starting with Parabolas: An Introduction to Chaotic Dynamics. Shanghai Scientific and Technological Education Publishing House, Shanghai, China (1993) (in Chinese).
12. Schneier, B.: Applied Cryptography – Protocols, Algorithms, and Souce Code in C. Second edn. John Wiley & Sons, Inc., New York (1996)