International Journal of Computational Intelligence Systems
Vol. **12(2)**, 2019, *pp.* 643–648
DOI: https://doi.org/10.2991/ijcis.d.190521.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

A Novel Algorithm for Generating Pseudo-Random Number

Gangyi Hu[1], Jin Peng[2], Weili Kou[1,*]

[1]*College of Big Data and Intelligence Engineering, Southwest Forestry University, Panlong District, Kunming, 650024, China*
[2]*Yunnan Vocational College of Judicial, Guandu District, Kunming, 650050, China*

ARTICLE INFO

Article History
Received 04 Jan 2019
Accepted 17 May 2019

Keywords
Cellular neural networks
Chaotic system
Generate pseudo-random number

ABSTRACT

It proposes a pseudo-random number generation algorithm based on cellular neural networks. This method uses the hyper-chaos characteristics of the cellular neural networks and sets the appropriate parameters to generate the pseudo-random number. The experimental results show that, compared with other similar algorithms, this algorithm has the characteristics of simple operation, low complexity, large key space, good initial value sensitivity, good cross-correlation characteristics, and good randomness. It can meet the needs of secure communication and image encryption, which has good application prospects.

1. INTRODUCTION

The random number has an important effect on data encryption, network information security, image communication, and satellite navigation. Studying the algorithm which can generate a random number with high randomness is becoming an important topic of information security. At present, some common algorithms such as taking the middle number or the congruence method. Because of the generation circle of the pseudo-random number depends on the initial values, the statistical performance of these pseudo-random numbers is not perfect [1,2]. Some other methods such as shift registered sequence generator and compound prime number generator also have weak random performance [3,4]. Bo proposed a random sequence algorithm based on knight cruising, which can achieve good randomness, but the knight cruising path is complex [5]. Han proposed an algorithm to generate the pseudo-random number based on the discrete chaotic synchronization system, and Dong proposed an algorithm to generate the pseudo-random number based on the cellular neural networks (CNNs)[6,7]. Both of these two algorithms used multiple chaotic iterations to generate pseudo-random numbers. Although they can obtain high-performance pseudo-random sequences, they also have some problems such as computational complexity and low utilization because of multiple chaotic iterations. In addition, there are also some other algorithms to generate a pseudo-random number based on high dimensional chaotic. Wang [8] generated a pseudo-random sequence of good random performance by using a three-dimensional Lorenz system. Qi [9] designed a pseudo-random number generator using the discrete hyper-chaotic mapping system. Although these methods can increase the key space, the weakness is that their cycle is short.

In fact, a complex high-dimensional chaotic system can improve the security of the pseudo-random sequence and enhance the anti-decoding ability of the system. It is an effective solution to adopt a hyper-chaotic system with multiple positive Lyapunov value and sequence generation algorithm with the best random performance. The CNNs is a nonlinear dynamic chaotic system. It is an artificial neural network based on Hopfield Neural Network and Cellular Automata. It has complex chaotic dynamic characteristics. This system will have multiple positive Lyapunov values with reasonable parameters, which have complex chaotic characteristics and security. The application of complex chaotic systems to pseudo-random sequence generation is the major method to generate high-performance pseudo-random numbers.

With the purpose of generating pseudo-random sequences according to high random performance. We propose an algorithm to generate pseudo-random numbers based on CNNs. It used the hyper-chaos characteristics of the CNNs to produce six-dimensional chaotic random sequences in high performance. The pseudo-random sequences which are generated by this algorithm are fast and have nonrepetitive. The experimental results show that these pseudo-random sequences have the characters such as the strong sensitivity of the initial value, the key space is large, the speed is fast and can meet the requirement of the detection standard of the National Institute of Standards and Technology (NIST).

2. THE CNNS

The CNNs was proposed by L.O. Chua [10]. Its basic unit is cells as shown in Figure 1, which are arranged in a planar two-dimensional lattice. CNN's unique characteristic compare with other types of neural network is local connectivity; each cell only has connections

*Corresponding author. Email: kwl_eric@163.com

to cells within its neighborhood. Theoretically, it can define the CNN networks of any dimension.

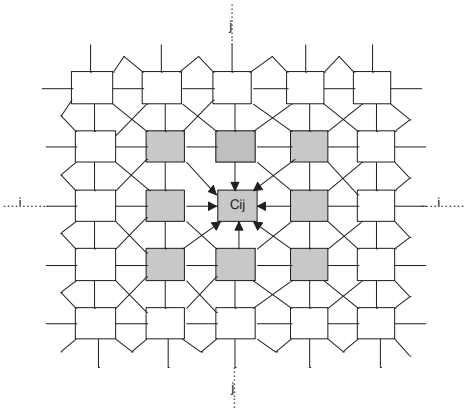

Figure 1 A standard cellular neursl network (CNN) model structure (two-dimensional).

Denoting the cell at row i and column j as C_{ij}, its neighborhood can be defined as

$$N_{ij}(r) = \{C_{ab} | \max(|a-i|, |b-j|) \le r, 1 \le a \le M, 1 \le b \le N\} \quad (1)$$

where $1 \le i \le M$, $1 \le i \le N$, r is the radius of the neighborhood of cell C_{ij}, and C_{ab} is the neighbor cell of cell C_{ij}.

Every cell in the CNN has an equivalent circuit, as shown in the Figure 2. The u, x, and y, respectively indicates the input, state, and output parameter of the cell. The node voltage x_{ij} is representing the state of the cell, and its initial amplitude value is not more than 1. The node voltage u_{ij} is representing the input of the cell with the requirement is constant amplitude and is less than 1.

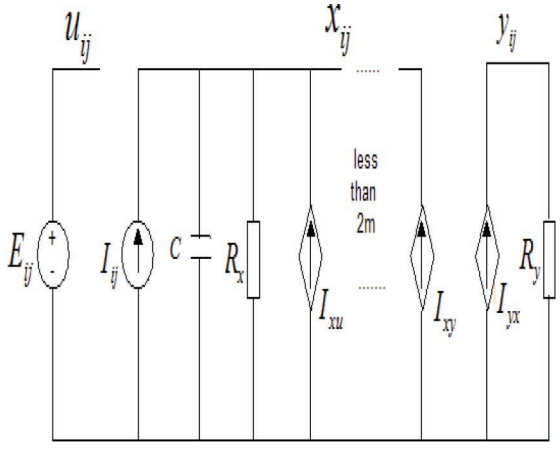

Figure 2 An equivalent circuit of a cell $C(i, j)$.

A cell is composed of a circuit which can be modeled by the first order nonlinear differential equation.

$$C\frac{dx_{ij}(t)}{dt} = -\frac{xij(t)}{R_x} + \sum_{k,l \in N_{ij}(r)} A_{kl} y_{kl}(t) + \sum_{k,l \in N_{ij}(r)} B_{kl} u_{kl} + I_{ij} \quad (2)$$

where x_{ij} is a state variable, y_{kl} is the outputs of cells, u_{kl} is the input of cells, C and R_x are system constants, I_{ij} is the threshold, A is the feedback parameter matrix, and B is the control parameter matrix. The subscripts after the matrices in the equation denote the matrix elements. The behavior of CNN is defined by these parameter matrices. Finally, the output equation of CNN is given by

$$y_{ij}(t) = \frac{1}{2}\left(|x_{ij}(t)+1| - |x_{ij}(t)-1|\right) = f(x) \quad (3)$$

In order to get the pseudo-random sequences to be used, we utilized six-units CNN. Since this is a small size, the neighborhood was defined to be the entire network. The challenge was how to discover the proper values for the parameter matrices A, B, and I that give rise to chaotic state evolution. In order to get these values, we set the system constants to $C = 1$ and $R_x = 1$, then performed a grid-based parameter search. One parameter set that we discovered that give rise to chaotic state evolution is shown in Equation (4)

$\mathbf{A} = \mathbf{0}$ except $a_{44} = 404$; $\mathbf{I} = \mathbf{0}$;

$$B = \begin{bmatrix} 0 & 0 & -1 & -1.2 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ 11 & -12 & 0 & 0 & 0 & 0 \\ 92 & 0 & 0 & -95 & 1 & -1 \\ 0 & 0 & 5 & 0 & -1 & 0 \\ 0 & 0 & 0 & 5 & 0 & -12 \end{bmatrix} \quad (4)$$

Substituting Equations (3) and (4) into Equation (2) and simplifying, we obtained the following state evolution equations for each of the six cells in the network. Note that we dropped the second subscript and simply use a single subscript to denote the different cells of the network.

$$\begin{cases} \frac{dx_1}{dt} = -x_3 - 1.2 * x_4 \\ \frac{dx_2}{dt} = 2 * x_2 + x_3 \\ \frac{dx_3}{dt} = 11 * x_1 - 12 * x_2 \\ \frac{dx_4}{dt} = 92 * x_1 - 95 * x_4 + x_5 - x_6 + 202 * \left(|x_4 + 1| - |x_4 - 1|\right) \\ \frac{dx_5}{dt} = 5 * x_3 - x_5 \\ \frac{dx_6}{dt} = 5 * x_4 - 12 * x_6 \end{cases} \quad (5)$$

Using Equation (5), the Lyapunov exponents of this system are −0.3824, 0.1283, 0.1596, −0.3995, −1.3580, −0.5473 respectively. There are two positive values in these Lyapunov exponents, which means that this system is hyper-chaotic system. The step-size parameter h can be chosen freely to a small value, which we set at 0.005. The initial value of x_i (where $i = 1, 2...6$) can be set to arbitrary values, each with any number of digits (up to machine precision). The initial state is the seed that starts the generation of chaotic sequences from the evolution of x_i. As long as the parameters given in Equation (4) is used. As an example, when the initial state is set as $x_1(0) = 0.1, x_2(0) = x_3(0) = x_4(0) = x_5(0) = x_6(0) = 0.2$ the CNN generated chaotic attractors as shown in Figure 3. In the

actual application, the above seven parameters (x_i ($i = 1, 2...6$) and h) can be set as any digit length value, it can greatly increases the key space.

The Figure 3 shows that the CNN system can generate chaotic system with the appropriate parameters.

In order to ensure that this chaos system generated by CNN is reliable, it judged the sensitivity of initial value for testing. Take the input initial value of the system x_4 and x_6 as an example, respectively, in this six-dimensional CNN system. Assuming that $x_4(0) = 0.2$ and $x_4(0) = 0.2 + 1 \times 10^{-16}$, $x_6(0) = 0.2$ and $x_6(0) = 0.2 + 1 \times 10^{-16}$, respectively, the evolution of x_4 and x_6 under these two initial conditions, as shown in Figure 4.

From Figure 4 can be seen that, although the two initial conditions of x_4 and x_6 only have the difference of 10^{-16}, after a finite time, although the initial state of these two chaotic signals is overlapped, and then there is an entirely different evolution process. It is taking the chaotic signal x_4 and x_6 as an example to analyze the chaotic characteristics of the initial value, in fact, for the chaotic signal x_1, x_2, x_3, and x_5, they also have a similar evolution process.

From the chaotic attractors, the two positive Lyapunov index, and the system is very sensitive to the initial input value can be determined that, the CNN parameters are set as specific parameters, it can generate chaotic system. The six-dimensional hyperchaotic sequences are generated based on the CNN system show that it has the chaos characteristics with strong randomness, and unpredictable.

3. THE METHOD OF GENERATING PSEUDO-RANDOM NUMBER

Because the CNNs have a good performance of chaotic characteristics, the pseudo-random number which is generated by the CNNs

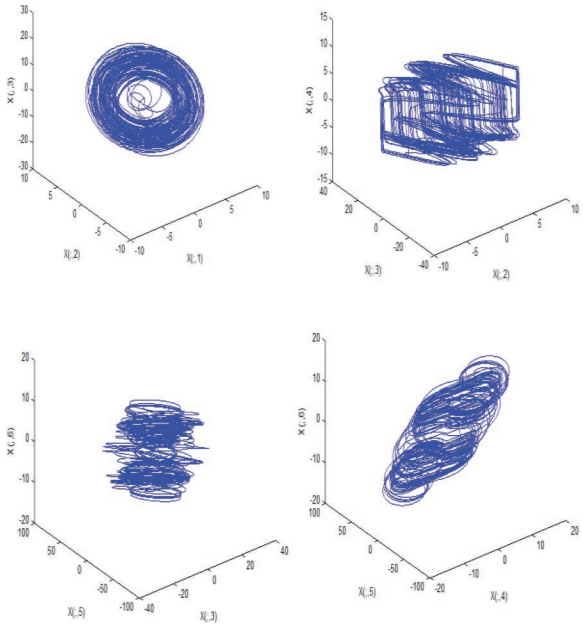

Figure 3 Some chaotic attractors generated by the six-dimensional cellular neural network (CNN).

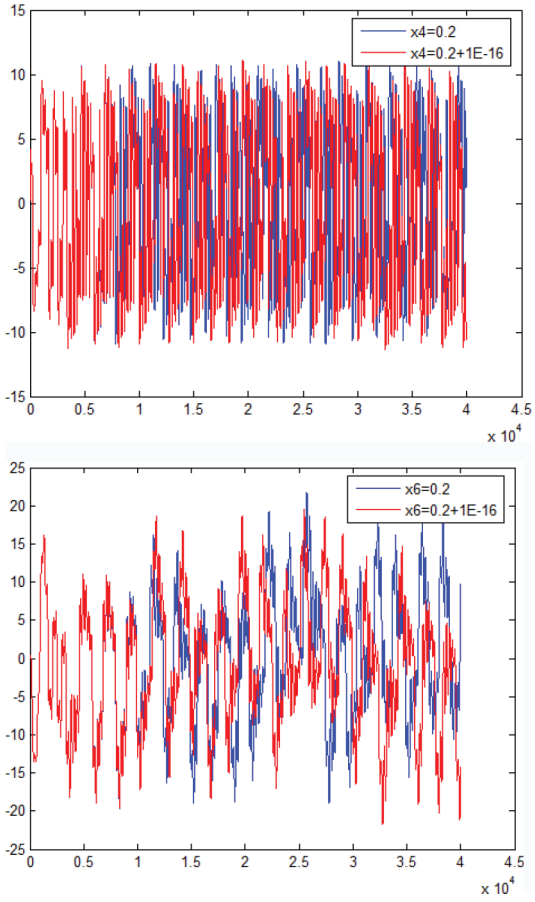

Figure 4 The sensitivity to initial value of the cellular neural network (CNN) six-dimensional systems.

depends on the key x_i ($i = 1, 2...6$) and step size h. In order to avoid the chaos degradation caused by the finite precision, the six output data from the iteration set as the feedback for the new input values each time to obtain the pseudo-random number with good performance. The main steps to generate pseudo- random numbers are as follows:

1. Set the constant coefficient value of the CNNs system, step size, and other initial parameters values (x_i ($i = 1, 2...6$)). These seven parameters (x_i and h) are also being as the key of the CNNs system, they can be set as any number of arbitrary digits.
2. After iterating the formula Equation (5) several times to eliminate the initial effect. The formula Equation (5) iterated once again, which can obtain six output values. These six values are taken as the values of the first sequences $\{x_1(k), x_2(k), x_3(k), x_4(k), x_5(k), x_6(k) | k = 0, 1, 2, ...\}$
3. Take the above six output values set as the new input value x_i' ($i = 1, 2...6$) for the CNNs system, and then iterate again.
4. According to the length of the pseudo-random sequences in practical application, repeat the above step three to get the final pseudo-random sequences.

According to the above steps, the system iterates 400,000 times, and the results are shown in Figure 5.

It can be seen that the iteration values are all over the whole interval, which show that this system has good pseudo-randomness.

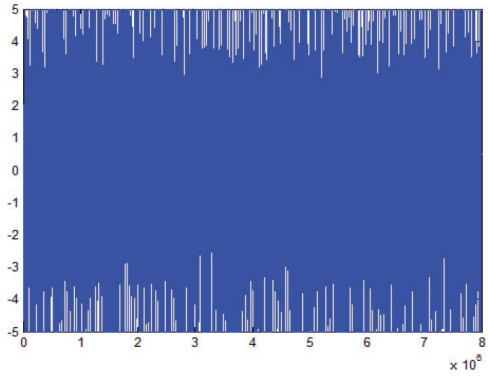

Figure 5 | The iteration result by the 6D CNN.

4. SECURITY ANALYSIS

4.1. The Key Space Analysis

With the purpose of resisting the enumerated attack, the key space for generating the pseudo-random sequences should be large enough. Our method used the different initial values (x_i and h) to obtain the different pseudo-random sequences. In this algorithm, the seven parameters (x_i and 2) can be set as any number of arbitrary digits. Its key space depends on the actual precision of the computer. Suppose that a 64-bit computer is used, the key space can be reached to 7×2^{64}. The key space is very large, which can effectively to resist exhaustive attack.

4.2. The Randomness Analysis

According to the randomness testing method which was proposed by the NIST800-22 [11]. The pseudo-random sequences generated by this algorithm is tested for a comprehensive way. Every test will get one *P_value*. When the test results to satisfy $P\_value \geq 0.01$, it is considered that the sequence is random in the test. When the test result to satisfy $P\_value < 0.01$, the sequence is considered as non-random in the test. The test results are shown in Table 1.

From the test results of Table 1, it shows that in each test result, the conditions are satisfied ($P\_value \geq 0.01$), the sequences generated by this algorithm can satisfy the NIST completely, which means that this sequence is randomness.

Table 1 | The test results from NIST800-22.

Test Type	Test values (group 1)	Test values (group 2)	Test values (group 3)	Test values (group 4)	Test values (group 5)	Test values (group 6)
Frequency	0.7058	0.6140	0.7973	0.5761	0.8713	0.4654
Block Frequency	0.8876	0.7723	0.9755	0.5001	0.8857	0.2108
Cumulative Sums	0.6782	0.5861	0.7132	0.6045	0.6878	0.3328
Runs	0.6023	0.6023	0.6023	0.6023	0.6023	0.6023
Rank	0.4019	0.2453	0.4019	0.2453	0.4019	0.2453
Discrete Fourier Transform	0.1742	0.1654	0.0565	0.2036	0.4132	0.1655
Overlapping Template Matching	0.3061	0.2101	0.2101	0.3061	0.2101	0.2101
Universal Statistical	0.5046	0.4578	0.3451	0.1979	0.2451	0.0824
Approximate Entropy	0.2804	0.2021	0.3328	0.1051	0.3670	0.3206
Random Excursions Variant	0.2825	0.3032	0.3216	0.2344	0.3542	0.2043
Serial	0.5088	0.4508	0.4960	0.3898	0.3445	0.3034
Linear Complexity	0.6125	0.5215	0.2074	0.4637	0.5032	0.2188

4.3. Analysis of the Effect of Image Encryption

The pseudo-random sequence is widely used in secure communication. We use the pseudo-random sequences which were generated by this algorithm for image encryption; the encrypted method used the image pixel XOR and position scrambling. The test 8-bit gray image is the Lena and Cameraman image. The cipher image and the histogram of the cipher image are shown in Figure 6.

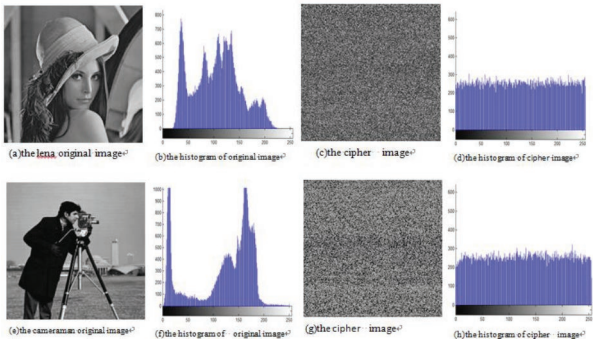

Figure 6 | The pseudo-random sequences generated by this algorithm are used for image encryption effect.

The pseudo-random sequences generated by this algorithm were applied to image encryption and compared with other algorithms. The number of pixel changed ratio (NPCR) and the information entropy (IE) is shown in Table 2.

Table 2 shows that using this algorithm for image encryption, the NPCR and IE is greater than other algorithms. This means that the pseudo-random sequences obtained from the CNNs have good applicability and can resist statistical attacks well.

4.4. Cross-Correlation Analysis

Cross-correlation is an important property of chaotic sequences used in cryptography. The cross-correlation function of ideal pseudo-random sequences is 0. For the pseudo-random sequences $\{x_1\}$ and $\{x_2\}$ which generated by system Equation (5) [12,13].

Table 2 | The image encryption effect by using different algorithms.

	Algorithms	NPCR	IE
Lena(256×256)	The Reference [14] method	0.9932	7.989
	The Reference [15] method	0.9953	7.989
	Our method	0.9960	7.991
Cameraman(256×256)	The Reference [14] method	0.9938	7.988
	The Reference [15] method	0.9959	7.989
	Our method	0.9961	7.992

The cross-correlation function is defined as

$$C(m) = \frac{1}{N}\sum_{i=0}^{N-m}\left(x_{1i} - \bar{x}_{1i}\right)\left(x_{2(i+m)} - \bar{x}_{2i}\right) \quad (6)$$

Among them, m is the correlation interval, $\bar{x}1i$ and $\bar{x}2i$ represent the mean of x_{1i} and x_{2i}, respectively. Figure 7 is the cross-correlation function of the two sequences. It can be seen that, the maximum deviation of the cross-correlation function is 0.01153. Therefore, the pseudo-random sequences have good cross-correlation characteristics.

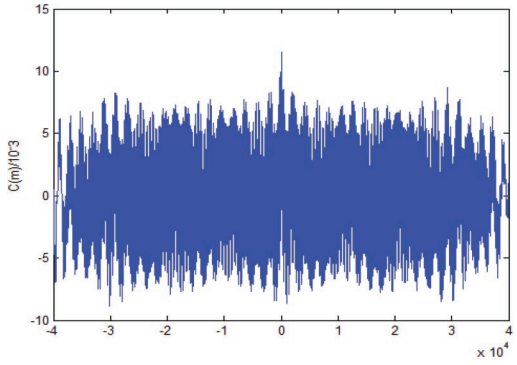

Figure 7 The cross-correlation function of sequences.

4.5. Comparison the Robustness to Noisy Cipher Pixel of Image Encryption

When a cipher image is transmitted through a wireless channel, it is unavoidably affected by noise. In order to achieve good recovery of the original image, an algorithm must have a certain ability to resist noise in the cipher image. In this experiment, we verify the effectiveness of our algorithm for noisy cipher image, which contains added Gaussian noise with zero means, and variance varying from 0.01 to 1. We compared our algorithm with the state of the art algorithm. The result of decryption from noisy cipher images is shown in Figure 8. In order to more easily interpret the result, we also calculated the root mean square error (RMSE) between the decryption image and the original image at various variances of the added Gaussian noise, shown in Figure 9. It can be seen that all algorithms perform similarly when the variance of the added noise is low. At higher variance value starting from 0.1 onward, our algorithm clearly outperforms the other three in term of robustness to noisy cipher image.

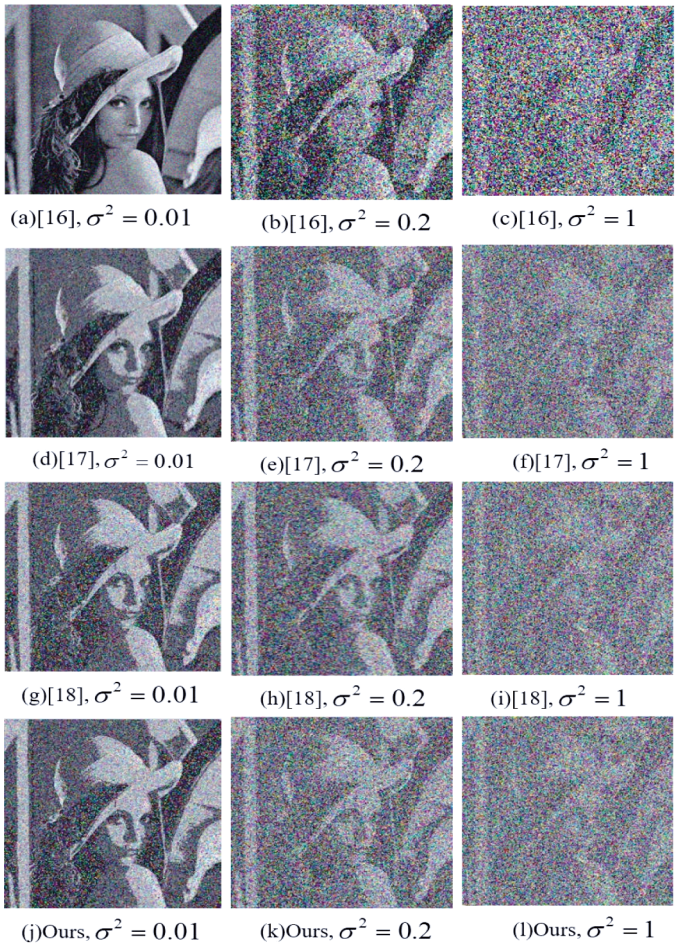

Figure 8 Decryption results when the cipher image is corrupted with noise.

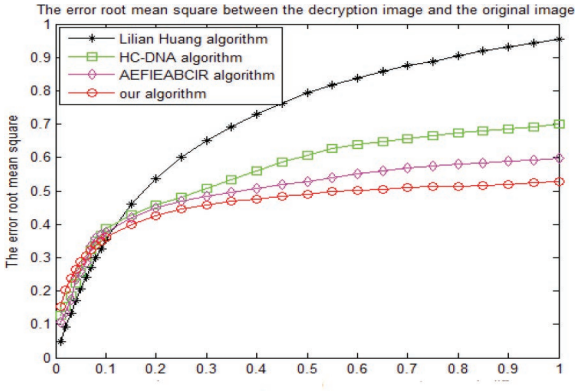

Figure 9 The root mean square error (RMSE) between the decrypted image and the original image at the various variance of the added Gaussian noise. For the plot legend: Lilian Huang is [16], HC-DNA is [17], and AEFIEABCIR is [18].

5. CONCLUSION

In this paper, we proposed a new pseudo-random number generation method which is designed by using the hyper-chaotic system of six-dimensional CNNs. It adjusted the initial input value of the CNNs system automatic iterated many times to obtain the pseudo-random number. Compared with other algorithms by using the CNN system and logistic mapping together to generate pseudo-random numbers, this algorithm is simple and has the character of low complexity. It also has good initial value sensitivity and cross-correlation characteristics. At the same time, these pseudo-random sequences generated by our algorithm show that it has a large key space and perfect randomness. The experiments show that this algorithm can be applied to secure communication very well, and it can meet the needs of network information security and expand the application of the chaotic system in cryptography. Which provide more nonlinear systems choices for generating independent, uniform, and complex random numbers.

REFERENCES

[1] J. Li, J. Zheng, P. Whitlock, Efficient deterministic and non-deterministic pseudo random number generation, Math. Comput. Simul. 143 (2018), 114–124.

[2] J.M. Bahi, X. Fang, C. Guyeux, An optimization technique on pseudo-random generators based on chaotic iterations, arXiv. 27 (2017), 1706–1713. https://arxiv.org/pdf/1706.08773.pdf

[3] R. Hamza, A novel pseudo random sequence generator for image-cryptographic applications, J. Info. Secur. Appl. 35 (2017), 119–127.

[4] T.T. Hue, T.M. Hoang, Complexity and properties of a multidimensional Cat-Hadamard map for pseudo random number generation, Eur. Phys. J. Spec. Top. 226 (2017), 2263–2280.

[5] S. Bai, L.-F. Zhou, H. Guo, B. Yan, Method to generate the pseudo random sequence based on the statistical properties, Chin. J. Netw. Info. Secur. 3 (2017), 31–38.

[6] L.-H. Dong, Method for generating pseudo random numbers based on cellular neural network, J. Commun. 37 (2016), 85–91.

[7] S.-S. Han, Lequan, T. Liu, Generalized Synchronization theorem based chaotic pseudo random number generator and performance analysis, Appl. Res. Comput. 30 (2013), 1512–1514.

[8] X. Wang, L. Liu, Cryptanalysis of a parallel sub-image encryption method with high-dimensional chaos, Nonlinear Dynamics. 73 (2013), 795–800.

[9] Y.B. Qi, K.H. Sun, H.H. Wang, The design and performance analysis of hyper-chaotic pseudo-random sequence generator, Comput. Eng. Appl. 53 (2015), 135–139.

[10] L.O. Chua, L. Yang, Cellular neural networks: theory, IEEE Trans. Circuits Syst. 35 (1988), 1257–1272.

[11] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, A statistical test suite for random and pseudo-random number generators for cryptographic applications NIST special publication 800-22. 59 (2010), 2289–2297. http://cs.sunysb.edu~algorith/implement/rng/distrib/SP800-22b.pdf

[12] Y.H. Wang, The design and applications of PRNG based on Henon map with parameter perturbation, J. Chin. Info. Process. 59 (2010), 2289–2297.

[13] M.B. Hossain, M.T. Rahman, B.M.S. Rahman, S. Islam, A new approach of image encryption using 3D chaotic map to enhance security of multimedia component, in International conference on Informatics, Electronics & Vision, Dhaka, Bangladesh.2014, pp. 1–6.

[14] O. Jallouli, S.E. Assad, M. Chetto, R. Lozi, Design and analysis of two stream ciphers based on chaotic coupling and multiplexing techniques, Multimedia Tools Appl. 77 (2018), 13391–13417.

[15] D. Han, L. Min, L. Hao, A chaos robustness criterion for 2d piecewise smooth map with applications in pseudorandom number generator and image encryption with avalanche effect, Math. Prob. Eng. 2016 (2016), 1–14.

[16] L. Huang, D. Shi, J. Gao, The design and its application in secure communication and image encryption of a new Lorenz-like system with varying parameter, Math. Prob. Eng. 2016 (2016), 1–11.

[17] K. Zhan, D. Wei, J. Shi, J. Yu, Cross-utilizing hyper-chaotic and DNA sequences for image encryption, J. Electr. Imaging. 26 (2017), 13–21.

[18] X. Wang, C. Liu, H. Zhang, An effective and fast image encryption algorithm based on chaos and interweaving of ranks, Nonlinear Dynamics. 84 (2016), 1595–1607.