Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **7**, Number **1**, February **2022**

A Novel Method to Generate Pseudo-Random Sequence based on GAN

Pujun Ji

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
jipujun@foxmail.com

Hongbin Ma*

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
*Correspondence: mahongbin@hlju.edu.cn

Qitao Ma

Faculty of Engineering
The Hong Kong Polytechnic University
11 Yucai Road, Hung Hom, Kowloon, Hong Kong, China
jack_coldsweat@163.com

Xuguang Chen

Electronic Engineering College
Heilongjiang University
No.74 Xuefu Road, Harbin, China
cxg190037571@163.com

Received November 2021; revised December 2021

Abstract. *Aiming at the problem of pseudo-random sequence generation, this paper innovatively introduces the generative confrontation network(GAN) into the pseudo-random sequence generation model. First, the GAN network model is improved to better adapt to one dimensional sequence learning and training. Then, GAN is used to learn and train chaotic sequences generated by chaotic system and a random sequence generation model is obtained by iterative training. Finally, the randomness of GAN sequence is tested. Te simulation results show that the generated sequence meets Golomb's three randomness postulate requirements and local randomness statistical test requirements and can be used as a key stream sequence to be applied to the encryption system.*
Keywords: Pseudo-random sequence, Generative adversarial network, Logistic mapping, Randomness detection

1. **Introduction.** Pseudo-random sequences are widely used in communication, cryptography, and coding. In the field of image encryption, traditional cryptographic technology is not suitable. The emergence of chaotic cryptography provides a new idea for the security protection of digital images. Meanwhile, the chaotic system with good performance has high implementation efficiency and low cost, and the software and hardware are convenient to implement. Therefore, chaotic cryptography is very suitable for the security protection of image files with large data volumes.

Chaotic system is highly sensitive to initial values and control parameters, ergodicity, pseudo-randomness and unpredictability, so it has been widely used in image encryption [1]. The nonlinear characteristics of chaotic systems can effectively offset the security risk of linear transformation in optical encryption. However, some cryptographic schemes based on chaos often have some shortcomings [2]. For example, the short period length caused by the limited accuracy of the computer is one of the important problems of the chaotic key stream generator [3]. In order to solve the randomness and security of keys, researchers have proposed many key generation schemes for chaotic systems or various other technologies [4-5].

The GAN has been concerned by many researchers since it was proposed. This network is mostly used to generate two-dimensional (2D) and three-dimensional (3D) data, [6-9] use this network to generate pictures, [10-11] generate videos and moving pictures. In the general application of GAN, the data to be processed is usually a picture in 2D. However, in our research, the data is generated by a chaotic model. We choose it due to the sensitivity of the chaos model to parameters, so the data we use is a 1D sequence. At present, there are few types of research on 1D GAN. Ruzicka et al. [12] so as to break through the problem of insufficient commercial data sets, a large number of client user movement patterns were generated by GAN, which provided sufficient data sets for subsequent research. In the work of [13], Cui et al. explored the application of GAN in the field of complex signals. They proved that an arbitrary signal generator (ASG) can be constructed by adjusting the parameters of 1D GAN and predicted the future research direction. Eskimez and Koishida [14] constructed 1D GAN based on convolution kernel. By comparing with the deep learning method, it was proved that this method has better effect and can restore more realistic high-frequency parts to low-resolution speech signals. Gao et al. [15] proposed a new data acquisition and fault detection method based on 1D GAN. This method can generate new fault features from natural faults and the fault detector can classify faults well. This model had the powerful ability of fault feature generation and detection.

To summarize the above, This paper starts from the perspective of theoretical research and practicability mainly studies the structure of the GAN and its implementation in sequence generation.

First, this paper improves the GAN network structure and loss function for one-dimensional data generation. Then, this paper constructs a new GAN sequence generation model based on the nonlinear and powerful fitting distribution ability of GAN. The periodicity problem of chaotic system is solved, and the performance and security of GAN sequence are guaranteed because of the one-time keys characteristic of GAN sequence generation model. Finally, the random character of GAN sequence is verified and the superiority of GAN sequence generation model is proved. This paper provides a new idea for generating key streams.

The second section of this paper mainly introduces the basic knowledge, including generative adversarial network, Logistic discrete chaotic system. The third section introduces the structure of the generator and discriminator. The fourth section analyzes characteristics of the sequence, including confidentiality, balance, run-length feature, and local random statistical. Finally, we summarize the paper in the fifth section.

2. **Basic Knowledge.**

2.1. **Generative Adversarial Network.**

Goodfellow et al. [16] first came up with a generative adversative network in 2014. Then a wave of deep learning research has been set off and more and more scholars have found the possibility of GAN. GAN's idea comes from

a zero-sum game, that is, the sum of the interests of both parties is a constant, and as one side's interests increase, the other side's interests decrease correspondingly. Figure 1 shows the basic structure of GAN that usually consists of a generator(G) and discriminator(D). The G model can be seen as a sample generator. It imitates the distribution of real data samples by inputting a noise z, so that the generated false samples have the probability distribution consistent with the real sample, but not exactly the same as the real sample. The D model can see that as a classifier, it can distinguish that the input sample is a real sample or fake sample [17].

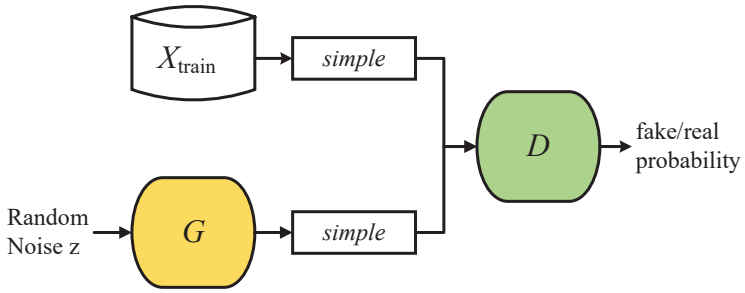


FIGURE 1. Structure of GAN

The training and optimization of generating and discriminating models is a binary minimax game problem. During training, fix one model, update the parameters of the other model and iterate alternately to maximize the errors of the other. The last, the generation model can learn the real sample distribution rules and generate enough samples to differentiate fake samples from real samples. The D model can accurately conclude that the input samples are real or fake.

The core principle formula of GAN is as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data(x)}}[\log D(x)] + E_{z\sim P_{z(z)}}[\log(1 - D(G(z)))] \tag{1}$$

where: x represents the real sample; z is random noise as input; $D(x)$ represents D determine the probability that the input is true; $G(z)$ represents the generated result when G receiving random noise; $P_{\text{data(x)}}$ represents the distribution of real data; $Pz(z)$ represents the distribution of generated data. The purpose of the D is to accurately judge the authenticity of the input sample, that is, make $D(x)$ infinitely close to 1; $D(G(z))$ infinitely close to 0. At this time, $V(D,G)$ becomes larger, that is, max D. The purpose of model generation is to generate more realistic samples, that is, to make $D(G(z))$ infinitely close to 1, at this time, $V(D,G)$ becomes smaller, that is, min G.

2.2. **Logistic Chaotic System.** In this paper, the Logistic map [18] is used to generate chaotic sequences as training sets for training GAN. The Logistic mapping equation is expressed as: $x_{n+1} = \mu x_n(1 - x_n)$, where the value $x_n \in (0,1)$ and the system parameter $\mu \in (0,4)$. The dynamic behavior of the Logistic chaotic map is closely related to the system parameter μ. Figure 2 shows the bifurcation characteristics of the Logistic chaotic map, which shows the 2D relationship between the system parameter μ and the numerical distribution of the iterative chaotic sequence. As can be seen from the figure, when $\mu < 3.5$, after a certain number of iterations, the generated value will converge to some specific values, which is unacceptable to us. As μ gets closer to 4, the complexity of its dynamic behavior increases. Previous studies have shown that the Logistic map is in the chaotic state when $3.56994568 \leq \mu \leq 4$, and only when the system parameter value

$\mu = 4$, the iterative value will be mapped in the whole $[0, 1]$ interval, which is called full mapping state [19].

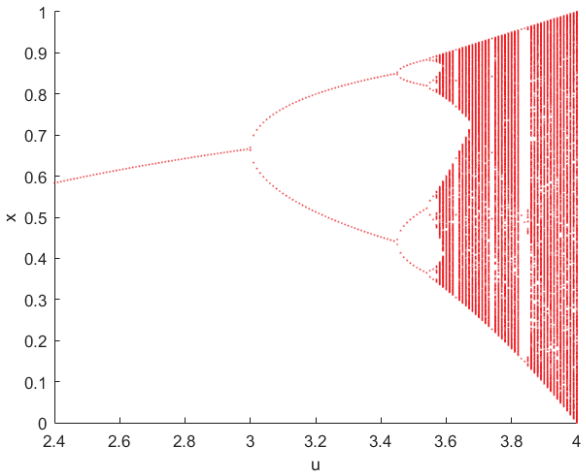


FIGURE 2. Bifurcation characteristics of Logistic chaotic map

3. **The Model Configuration.** Generative adversarial networks many used in the image domain, such as image super-resolution and image style migration, etc. The structure of GAN that process images is complex and is not suitable for processing 1D data. On the premise of ensuring the quality of the generated data, we are based on the WGAN-GP network model [20], improved the configuration of the generator and discriminator to make it more suitable for processing 1D data. It not only drops the complexity of the network but also increase the operating speed of the network.

This paper combines the convolutional neural network with GAN. The training data complete unsupervised training in the convolutional neural network and the G model. Thanks to the powerful feature extraction capabilities of convolutional networks, the learning effect of GAN has been greatly improved. The GAN training process is more stable and generates better samples. The format of data set adopted in this paper is $[N, L, 1]$ generated by the Logistic Chaotic System, where N is the number of data in a set of data, L is the length of a single sequence, and 1 is the dimension of the data set.

In Table 1 the configuration of G is described. In the G, which consists of a fully connected layer and four 1D convolution layers, the activation function is LeakyReLU[21], which can solve the dying ReLU problem and speed up convergence. The activation function of the last output layer adopts the tanh function, the tanh function overcomes the disadvantage of the sigmoid not being symmetric at the origin. The input of the G network is random noise z, and the output data is a 1D sequence with length L after processing by G.

The configuration of D is described in Table 2. The D consists of two 1D convolution layers, a Flatten, a Dropout, and two denses. The activation function of the convolution layer is LeakyReLU. This activation function was selected for the same reason as conv of the D network. The activation function of the last dense is tanh, for the same reason as the output layer of the G Network. The function of Flatten layer is to compress multidimensional data into 1D data, which is used between the convolution layer and dense layer. The dropout layer can mitigate the occurrence of overfitting.

The specific structure of G and D is shown in Figure 3. The convolution neural network is combined with GAN. G inputs a 1D random noise vector and outputs a sample with

TABLE 1. G network configuration

Layer	Activation	Filters	Filters.size	Strides
Dense	LeakyReLU			
Conv	LeakyReLU	32	3	1
Conv	LeakyReLU	32	3	1
Conv	LeakyReLU	32	3	1
Conv	tanh	1	5	1

TABLE 2. D network configuration

Layer	Activation	Filters	Filters.size	Strides
Conv	LeakyReLU	32	3	1
Conv	LeakyReLU	32	3	1
Flatten				
Dense		64		
Dropout	LeakyReLU			
Dense	tanh	1		

length L after processing by four convolution layers. The convolution kernel has a size of 3 and a step of 1. D is a convolutional network without a pooling layer. D receives real data and samples generated by G, extracts data features of the input sequence, and judges the authenticity of the input samples. The output of D is used to represent the probability that the input sample is true.

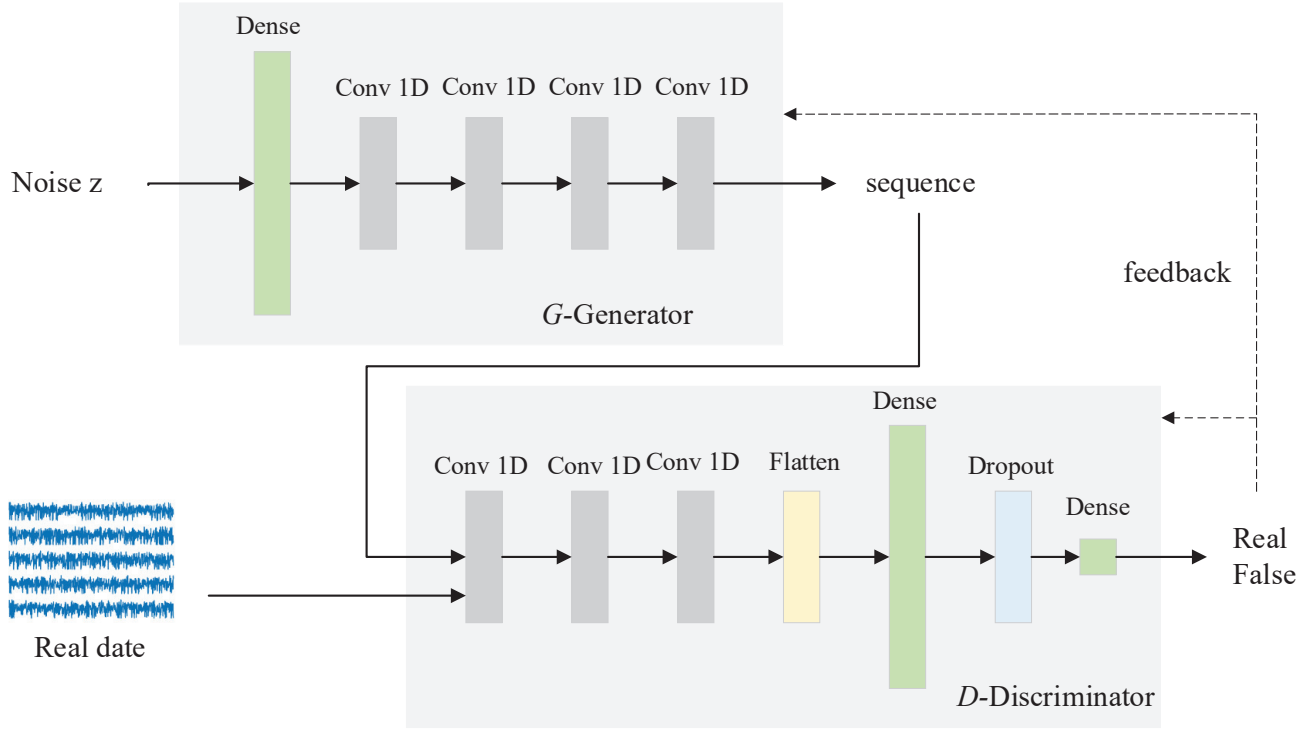


FIGURE 3. The structure of G and D

4. Sequence Characteristic Analysis.

4.1. **Confidentiality Analysis.** For a single chaotic map, the initial value of the system can be estimated by using a nonlinear inverse method, and some system parameters can be estimated by using the statistical properties of chaotic sequences. In addition, due to the finite precision effect of chaotic sequence implementation, the chaotic sequence of single mapping has hidden danger. The sequence generated by the generative adversarial model proposed in this paper overcomes this shortcoming because the sequence is dynamically generated in the game between generator and discriminator and is a dynamically variable one-time pad. It is not easy to get the initial value of the system by the inverse method. Even if the initial value is obtained, the sequence generated by the model next time cannot be predicted. Models trained with different training sets can generate sequences at the same time, which can be used at the same time in the process of image encryption and has better security than a single sequence.

4.2. **Balance.** The Logistic with the initial value $x_0 = 0.3711$ and $u = 3.99$ was used to obtain N chaotic sequences of length L. We use this data to train the model. After the training was completed, the number of 0 and 1 were found to be irregular, but the difference was small. The statistical results are shown in Table 3.

Table 3. The number of 0 and 1

	1000	10000	20000	40000
0	478	5018	10051	20172
1	522	4982	9949	19828

Theoretically, the 0-1 ratio of binary pseudo-random sequence can be calculated as follows:

$$r_{01} = \min_{L\to\infty} \frac{N_0(J)}{N_1(J)} = \frac{1 - \iint_{00}^{LL} T_n(x,y)P(x,y)\mathrm{d}x\mathrm{d}y}{\iint_{00}^{LL} T_n(x,y)P(x,y)\mathrm{d}x\mathrm{d}y} = 1 \tag{2}$$

where, $N_0(J)$ and $N_1(J)$ respectively represent the number of 0 and 1 in the binary pseudo-random sequence, from which the sequence has a balanced 0 - 1 ratio.

4.3. **Run-length Feature.** The results are shown in Table 4, the number of runs 0 and 1 in runs of different lengths is roughly equal, and the total number of runs of length L accounts for roughly $1/2^l$ of all runs of different lengths.

4.4. **Relevant Features.** The sequence autocorrelation function is defined as follows:

$$R(m) = \lim_{l\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} n = (S_i - S_{\mathrm{mean}})(S_{i+m} - S_{\mathrm{mean}}) \tag{3}$$

Cross-correlation function:

$$C(m) = \lim_{l\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} n = (S_i - S_{\mathrm{mean}})(S'_i - S_{\mathrm{mean}}) \tag{4}$$

Table 4. Run statistics for the output sequence

The length of run	0	1	0/1	Test values	Theoretical value
1	10002	10087	0.991573	0.501423	0.500000
2	5008	4993	1.003004	0.249626	0.250000
3	2483	2421	1.025609	0.122404	0.125000
4	1305	1296	1.006944	0.064921	0.062500
5	610	623	0.979133	0.030766	0.031250
6	305	315	0.968254	0.015475	0.015625
7	168	154	1.090909	0.008037	0.007813
8	65	69	0.942029	0.003345	0.003906

where, $\{S_i\}$ and $\{S_i'\}$ are two binary sequences with different initial values, and S_{mean} is the mean value of the sequence. The autocorrelation and cross-correlation properties of the Logistic sequence and generated data sequence are shown in Figure 4.

In Figure 4, abscissa m represents the step-size parameter. In Figure 4(a) and Figure 4(b), when the step size changes, the smaller the autocorrelation coefficient changes, the better the randomness of the corresponding sequence. Obviously, we can see that the sequence generated by GAN has better autocorrelation than the original Logistic sequence. In Figure 4(c) and Figure 4(d), if the cross-correlation function value is closer to 0, it indicates that the two sequences are more unrelated, and the different degree is greater. The sequence generated by GAN is a little closer to 0 than the original Logistic sequence. It can be seen from the figure that the sequence generated by GAN has sharp autocorrelation and good cross-correlation and has properties similar to δ-like.

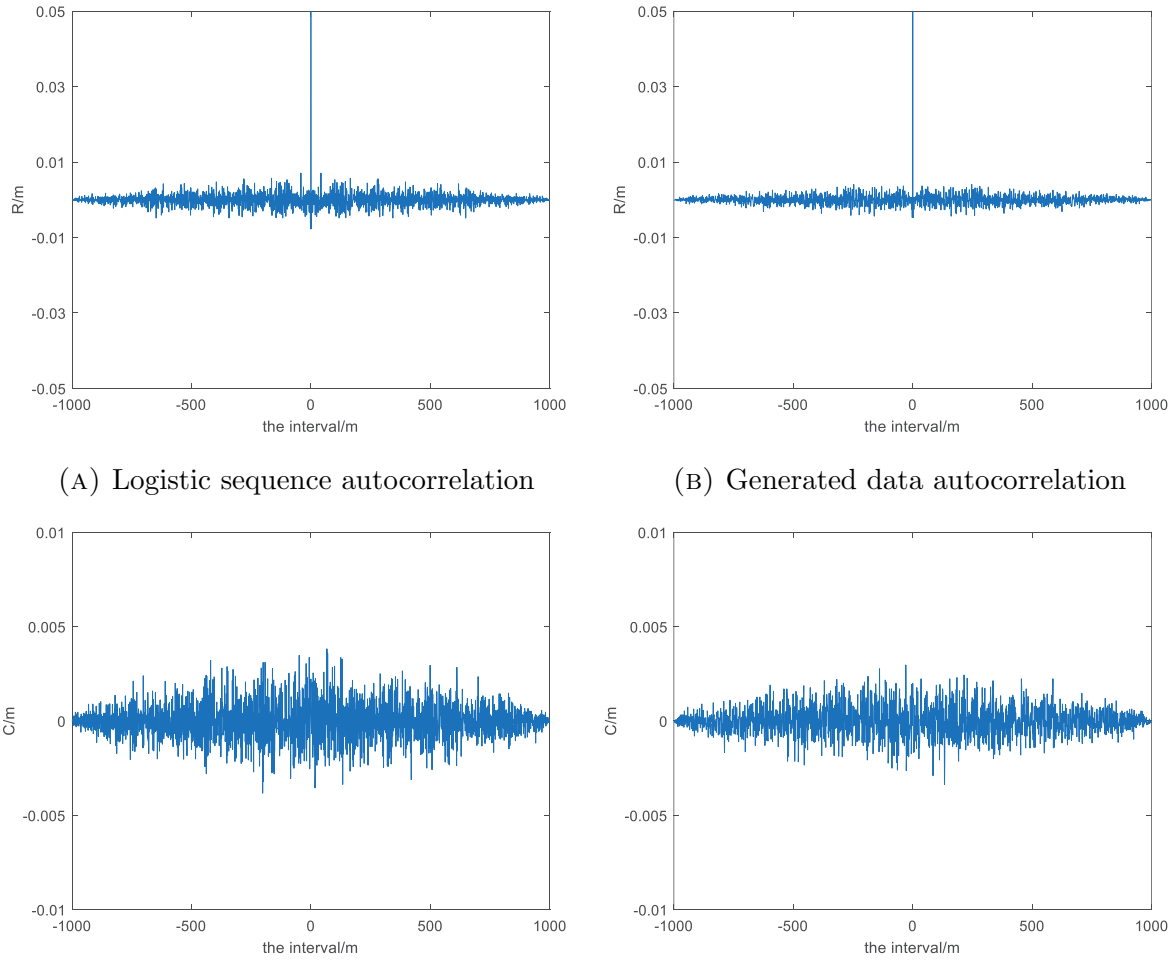
4.5. **Local Random Statistical Test.** If the key sequence is to be used for encryption in the encryption system, besides passing the three-point randomness postulates of Golomb [22], the local randomness test should be further carried out, and each paragraph of the sequence should be statistically tested.

We used the NIST suite for statistical tests, the significance level is set to 1%. If a P-value > 0.01, the binary sequence is considered random. It has a 99% confidence rate. The acceptance levels of several local randomness tests are briefly listed below, as shown in Table 5.

Table 5. Local randomness test

Test name	P-value	Results
Frequency	0.0861	success
Block-frequency	0.4318	success
Runs	0.5433	success
Long runs of ones	0.9187	success
Rank	0.2941	success
FFT	0.1132	success
Non-overlapping template	0.5387	success

Theoretical analysis and statistical results show that the generated sequence can pass the local random performance test, and is an excellent pseudo-random sequence, which is more random and unpredictable and can be used in secure communication and other fields.



(A) Logistic sequence autocorrelation

(B) Generated data autocorrelation

(C) Logistic sequence cross-correlation

(D) Generated data cross-correlation

FIGURE 4. The autocorrelation and cross-correlation properties of Logistic sequence and generated data sequence

5. **Conclusion.** In this paper, A novel method based on GAN to generate pseudo-random sequences is proposed. The improved scheme is based on GAN, chaos model, and convolutional neural network. GAN is used to learn chaotic random key, train and generate GAN key pool. GAN sequence not only has the advantages of chaotic random key, but also has the characteristics of non-repeatability, which greatly improves the key generation speed and increases the security of encryption system. Alternating GAN models with different initial values can greatly improve the randomness of the sequence and thus improve security. Therefore we can create a large number of new sequences using the methods presented in this article. The simulation results show that the generated sequence conforms to NIST standard and has strong randomness, which can be used in security authentication, image encryption, secure communication, and other fields. In the future, our work will focus on improving the generality of the GAN model and studying the influence of different training sets on GAN generation results.

Acknowledgment. This work is partially supported by the mentor. The authors also thanks to the reviewers for their helpful comments and suggestions, which improved the article.

REFERENCES

[1] Z. L. Man, J. Q. Li, X. Q. Di, O. Bai, An Image Segmentation Encryption Algorithm Based on Hybrid Chaotic System, *IEEE Access*, vol. 7, pp. 103047–103058, 2019.

[2] C. T. Li, C. L. Chen, C. C. Lee, C. Y. Weng, C. M. Chen, A Novel Three-party Password-based Authenticated Key Exchange Protocol with User Anonymity Based on Chaotic Maps, *Soft Computing*, vol. 22, Issue. 8, pp. 2495–2506, 2018.

[3] N. Tsafack, J.Kengne, B. A. Atty, A. M. Iliyasu, K. Hirota, Design and Implementation of A Simple Dynamical 4-D Chaotic Circuit with Applications in Image Encryption, *Information Sciences*, vol. 515, pp. 191–217, 2020.

[4] C. M. Chen, K. H. Wang, T. Y. Wu, E. K. Wang, On the Security of a Three-party Authenticated Key Agreement Protocol based on Chaotic Maps, *Data Science and Pattern Recognition*, vol. 1, no. 2, pp. 1–10, 2017.

[5] C. M. Chen, L. L. Xu, K. H. Wang, S. Liu, T. Y. Wu, Cryptanalysis and Improvements on Three-Party-Authenticated Key Agreement Protocols Based on Chaotic Maps, *Journal of Internet Technology*, vol. 19, no. 3, pp. 679–687, 2018.

[6] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint*, arXiv:1511.06434, 2015.

[7] Y. L. Wang, X. J. Li, H. B. Ma, Q. Ding, Matin Pirouz, Image super-resoluton reconstruction based on improved generative adversarial network, *Journal of Network Intelligence*, vol. 6, No.2, pp. 154–163, 2021.

[8] T. Y. Wu, X. Fan, K. H. Wang, J. S. Pan, and C. M. Chen, Security Analysis and Improvement on an Image Encryption Algorithm Using Chebyshev Generator, *Journal of Internet Technology*, vol. 20, no. 1, pp. 13–23, 2019.

[9] H. B. Ma, X. G. Chen, Y. L. Wang, P. J. Ji, Efficient face attribute editing method based on GAN, *Journal of Network Intelligence*, vol. 6, No.3, pp. 646–655, 2021.

[10] A. Trockman, J. Z. Kolter, Orthogonalizing convolutional layers with the cayley transform, *arXiv preprint*, arXiv:2104.07167, 2021.

[11] A. Gupta, J. Johnson, F. F. Li, S. Savarese, A. Alahi, Social GAN: socially acceptable trajectories with generative adversarial networks, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[12] M. Ruzicka, M. Volosin, J. Gazda, T. Maksymyuk, The extension of existing end-user mobility dataset based on generative adversarial networks, *International Conference Radioelektronika*, 2020, https://doi.org/10.1109/RADIOELEKTRONIKA49387.2020.9092404

[13] L. Cui, P. Zhao, K. Wang, J. Yang, X. Bu, A kind of arbitrary signal generator based on 1D generative adversarial networks, *8th Data Driven Control and Learning Systems Conference*, pp. 1324–1328, 2019.

[14] S. E. Eskimez, K. Koishida, Speech super resolution generative adversarial network, *44th IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3717–3721, 2019.

[15] S. Gao, X. Wang, X. Miao, C. Su, Y. Li, ASM1D-GAN: an intelligent fault diagnosis method based on assembled 1D convolutional neural network and generative adversarial networks, *Journal of Signal Processing Systems*, vol. 91, pp. 1237–1247, 2019.

[16] I. J. Goodfellow, Generative adversarial nets, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 2672–2680, 2014.

[17] Y. Jo, J. Park, SC-FEGAN: face editing generative adversarial network with user's sketch and color, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp.1745–1753, 2019.

[18] J. Zhang, Y. Zhu, H. Zhu and J. Cheng, Some improvements to logistic map for chaotic signal generator, *3rd IEEE International Conference on Computer and Communications*, pp. 1090–1093, 2017.

[19] C. Liu and Q. Ding, A modified algorithm for the logistic sequence based on PCA, *IEEE Access*, vol. 8, pp. 45254–45262, 2020.

[20] S. Liu, D. Li, T. Cao, GAN-based face attribute editing, *IEEE Access*, vol. 8, pp. 34854–34867, 2020.

[21] Z. M. Chan, C. Y. Lau, K. F. Thang, Visual Speech Recognition of Lips Images Using Convolutional Neural Network in VGG-M Model, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 3, pp. 116–125, 2020.

[22] Z. He, W. Zuo, M. Kan, S. Shan, X. Chen, AttGAN: Facial Attribute Editing by Only Changing What You Want, *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5464–5478, 2019.