Based on GAN Generating Chaotic Sequence

Xuguang Chen¹, Hongbin Ma¹(✉), Pujun Ji¹, Haiting Liu¹, and Yan Liu²

¹ Electronic Engineering College, Heilongjiang University, No. 74 Xuefu Road, Harbin, China
mahongbin@hlju.edu.cn

² National Computer Network Emergency Response Technical Team/Coordination Center of
China, Beijing, China
liuyan@cert.org.cn

Abstract. In this paper, an adversarial encryption algorithm based on generating chaotic sequence by GAN is proposed. Starting from the poor leakage resistance of the basic adversarial encryption communication model based on GAN, the network structure was improved. Secondly, this paper used the generated adversarial network to generate chaotic-like sequences as the key K and entered the improved adversarial encryption model. The addition of the chaotic model further improved the security of the key. In the subsequent training process, the encryption and decryption party and the attacker confront each other and optimize, and then obtain a more secure encryption model. Finally, this paper analyzes the security of the proposed encryption scheme through the key and overall model security. After subsequent experimental tests, this encryption method can eliminate the chaotic periodicity to a certain extent and the model's anti-attack ability has also been greatly improved. After leaking part of the key to the attacker, the secure communication can still be maintained.

Keywords: GAN · Chaos model · Key generation · Data protection

1 Introduction

With the development of science and technology, the importance of data is becoming more and more obvious, and data protection is also highly valued. Information security includes a wide range of encryption technology is one of the important technologies to ensure information security.

In 2016, Abadi et al. proposed an adversarial encryption algorithm based on neural network, which consists of the communication party Alice and Bob and the attacker Eve. Eve tried to decipher the communication model between Alice and Bob. Alice and Bob tried to learn how to prevent Eve's attack. The three used this confrontation training to increase their performance. However, this model has no way to show what the two communication parties and the attacker learned in the end, nor can they judge whether the password structure is safe. Therefore, this paper first analyzed the security of the scheme in detail and statistics the security of the leaked part of the key and then improved the structure of the existing network according to the remaining problems in the system. In addition, a key generation model based on GAN was constructed. It takes Logistic

mapping as input, and generated chaotic sequence as encryption key with the help of confrontation training between networks, and inputs into the subsequent encryption model to obtain a more secure encryption algorithm. To get a more secure encryption algorithm. Finally, this paper analyzed the security of the proposed encryption scheme through the key and the overall model security.

2 Basic Adversarial Encryption Algorithm and Chaotic Map

2.1 Basic Adversarial Encryption Communication Model

In 2016, Abadi M, etc. first proposed the use of GAN to implement encrypted communication [1]. Through this technology, encrypted and secure communication during enemy monitoring is realized. Its work is based on traditional cryptography scenarios, and its workflow is shown in Table 1.

Table 1. Training of basic adversarial encryption communication model based on GAN.

The training steps of the basic adversarial encryption communication model based on GAN

1. Alice encrypts plaintext P with key K to generate ciphertext C
 2. Bob gets ciphertext C and decrypts it when he knows the key K to get message P_{Bob}
 3. Eve obtains ciphertext C and decrypts P_{Eve} without key K
 4. The communication model and attack model are optimized in the training
 5. Finally, P_{Bob} is the same as P , and the gap between P_{Eve} and P is as wide as possible
-

In terms of internal model construction, Alice and Bob have the same model. The Eve network adds a fully connected layer to simulate the key generation process. Eve is trained to improve his decryption ability to make $P_{Eve} = P$. Figure 1 shows the network structure model of Alice, Bob, and Eve.

Fig. 1. The network structure of Alice, Bob and Eve.

The L1 distance is used to calculate the distance between the plaintext and its estimated value. The L1 distance is defined as:

$$d(P, P') = \frac{1}{N} \sum |P_i - P'_i| \quad (1)$$

Eve's loss function is defined by an expected value:

$$L_E(\theta_A, \theta_E) = E_{P,K}[d(P, D_E(\theta_E, E_A(\theta_A, P, K)))] \quad (2)$$

Where θ_A and θ_E are the neural network parameters of Alice and Bob respectively. P is the encryption output $E_A(\theta_A, P, K)$ when the plaintext K is the key and $D_E(\theta_E, C)$ is the decryption output when Eve inputs ciphertext C . Similarly, Bob's loss function is defined as follows:

$$L_B(\theta_A, \theta_B) = E_{P,K}[d(P, D_B(\theta_B, E_A(\theta_A, P, K), K))] \quad (3)$$

Where $D_B(\theta_B, C, K)$ is the decryption output when Bob inputs ciphertext C and key K .

Alice and Bob should exchange data accurately while defending against attacks. Therefore, the joint loss function of communication parties is defined by combining L_B and L_E , which is defined as follows:

$$L_{A,B}(\theta_A, \theta_B) = L_B(\theta_A, \theta_B) - L_E(\theta_A, O_E(\theta_A)) \quad (4)$$

Where $O_E(\theta_A)$ is the optimal Eve found by minimizing the loss.

However, according to continuous research, people gradually found the problems exposed by this basic model. The main problem is that when part of the key and plaintext are leaked, it is no longer an acceptable encrypted communication scheme [2]. This will cause losses that are difficult to assess for the data protection of both parties. Experiments showed that neither Bob nor Eve could easily converge when the amount of plaintext leaked to Eve was small. When the number of leaked bits in plaintext exceeded 9 bits, the two could gradually converge, but at this time Eve's decryption ability also increased. When the number of leaks is 16 bits, both could get good performance. The experimental results are shown in Fig. 2, where the ordinate indicates the decryption error rate, and the abscissa indicates the number of training rounds.

Fig. 2. The decryption of Bob and Eve when the amount of leaked plaintext is 8, 10, and 16 bits respectively.

2.2 Logistic Chaotic Map

Up to now, many classic chaotic models have been widely used, such as Logistic mapping, Tent mapping, Lorenz chaotic model, etc. In this paper, Logistic mapping is selected as the encryption method, which has a simple mathematical form and its complex dynamic behavior [3]. Its system equation is as follows:

$$x_{n+1} = \mu x_n (1 - x_n) \quad (5)$$

The range of μ in the formula is 0 to 4, which is called the logistic parameter. Studies have shown that when x is between 0 and 1, the Logistic map is in a chaotic state. Outside this range, the sequence generated by the model must converge to a certain value [4]. And when the value of μ is 3.5699456 to 4, the value generated by the iteration presents a pseudo-random state. In other ranges, convergence will occur after a certain number of iterations, which is unacceptable to us.

In this paper, GAN can automatically learn the characteristics of data distribution of the real sample set, to automatically learn the data distribution of Logistic mapping and generate variable sequences as keys for subsequent encryption models. Besides, the mapping equations of $\mu = 3.5699456$ and $\mu = 4$ are selected as the input of the GAN model. The powerful learning ability of the model is used to generate the distribution between the two input data distributions and try to fit the original data distribution.

3 Adversarial Encryption Algorithm Based on Logistic Mapping

3.1 Key Generator Based on GAN

In this section, the GAN was used to simulate the chaotic model, and then a random sequence similar to the chaos was generated as an encryption key. The GAN model contains two systems, namely the discriminating system and the generating system. The results of the two systems will be opposed to each other and will be updated in turn [5]. The principle of the binary game is used to achieve the optimal state.

Improvement of Key Generator Network Structure. GAN has high requirements for hardware devices. If the network structure is too complex, GAN may not only lead to the collapse of the platform but also reduce the efficiency of the key generation to some extent. Because of this situation, this paper proposed a form of self-encoder and GAN fusion to generate data.

The self-encoder can traverse the input information, convert it into efficient potential representations, and then output something that it wants to look very close to the input. The model includes generative networks and discriminant networks, except that part of the generative network uses self-encoders as the basic structure. Besides, in the overall framework, the information source should not only be used as the input source for the generator to extract the generation factor, but also need to be mixed with the real samples from the training set, because the generated fake samples contain key information features. The self-encoder consists of two parts: the encoder extracts potential features

Fig. 3. Basic structure of self-encoder.

from the input source, and the decoder replaces these potential features with the output source. Its structure is shown in Fig. 3.

The key generation model is combined with the GAN network and the self-encoder, in which the self-encoder is applied to the generating network part and the rest is composed of the GAN model. The experimental results showed that this method had a great improvement in the generation effect, and it could almost coincide with the original model distribution in the later stage of training.

In addition to the improvement of the above structure, this paper also introduced the concept of parallel training. The original GAN model uses the idea of serial training, in which network A is used to distinguish between true and false tags, and network B is used to generate false tags. This method has many disadvantages in the training process, for example, the number of rounds of discriminating network training cannot be determined. In this paper, the idea of parallel training was used to solve the two problems. The parallel structure made the two networks compete at the same time, and the training was no longer sequential, which greatly improved the training speed. The improved network structure is shown in Fig. 4.

The self-encoder network consists of convolutional networks and deconvolutional networks, which correspond to encoder and decoder respectively. The convolution layer number of both is 4, and the activation function of the convolution layer is ReLU. The structure of the auto-encoder generation network is shown in Fig. 5.

The number of layers in the discriminant network is also 4, as shown in Fig. 6.

Simulation to Test the Improved Effect. In this paper, the improvement of training speed is verified through simulation. With the help of Python, the CPU training data collected for 200 times by serial and parallel models were counted, excluding the time consumption at the start of training. The results show that the parallel GAN cancels the cumbersome operations such as discriminating network parameter transmission and feedforward of the B network. It and can better adapt to the popular platform with moderate computing power and the time consumption is also greatly reduced. Statistics are shown in Table 2.

According to the study of the chaos model, when the parameter is greater than 3.5699456 and less than or equal to 4, Logistic mapping enters into the chaos state.

Fig. 4. Parallel training network structure.

Fig. 5. Generate network structure.

Fig. 6. Discriminate network structure.

Table 2. Time-consuming comparison of serial/parallel GAN network training.

Network type	Minimum value	Maximum value	Average value
Serial GAN	1.7241	1.9857	1.8325
Parallel GAN	0.8754	1.4732	1.1473

The purpose of this section is to train a network. The mapping equation $\mu = 3.5699456$ and $\mu = 4$ is selected as the input of the GAN generation model. During the training process, the networks competed with each other. When the accuracy d generated by the discriminator is 0.5, the key generated at this time is extracted and used for the subsequent encryption system. The key generation algorithm is as follows (Table 3):

Table 3. Key generation algorithm based on GAN.

The key generation algorithm based on GAN:
<ol style="list-style-type: none"> 1. Initialize network parameters 2. While $i < N$, do: <ol style="list-style-type: none"> a. for k-step, do: <ol style="list-style-type: none"> (1) Select m samples $\{z_1, z_2, \dots, z_m\}$ from P_z (2) Select m samples $\{x_1, x_2, \dots, x_m\}$ from P_{data} (3) Fixed generator, update discriminator network parameters by the gradient ascent algorithm: $\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log(1 - D(G(z^i)))]$ b. A block of m samples $\{z_1, z_2, \dots, z_m\}$ is sampled from the initial distribution P_x c. Fixed discriminator, updates generator network parameters by gradient descent algorithm: $\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^m 1 - \log(1 - D(G(z^i)))$

The results of the training are shown below. Figure 7 shows the results when the number of iterations is 5000 and the discriminator output accuracy is 0.5. Figure 8 shows the unexpected results generated by the model when the number of iterations is less than 500, or the discriminator output accuracy is not selected properly.

Fig. 7. The number of iterations is 5000 and the discriminator output accuracy is 0.5.

3.2 The Overall Encryption Algorithm Design

The adversarial encryption algorithm described in this paper mainly includes sufficient security of the encryption system and encryption key. 3.1 Section introduced the process

Fig. 8. The number of iterations is less than 500, or the discriminator output accuracy is not selected properly.

Table 4. Improved adversarial encryption algorithm based on GAN.

Process of the encryption algorithm
1. Plaintext $[P_0, P_1, \dots, P_n]$ and key $[k_0, k_1, \dots, k_n]$ are sent to the Alice network
2. From the bit-angle mapping formula $f(b) = \arccos(1 - 2b)$, the bit-angle is converted to Angle $[a_0, a_1, \dots, a_n]$
3. The full connection layer uses the hidden variable $[h_0, h_1, \dots, h_n]$ as the initial ciphertext with the Angle information
4. Using formula $f^{-1}(a) = \frac{1 - \cos(a)}{2}$, the initial ciphertext in 3 is transformed into the final ciphertext $[C_0, C_1, \dots, C_n]$
5. Alice, the encryptor, sends ciphertext $[C_0, C_1, \dots, C_n]$ to Bob
6. Bob and Eve receive ciphertext and output plaintext P
7. Start alternate training of Alice, Bob and Eve networks
8. When Eve's decryption accuracy is high, Alice selects the encryption key again and circulates 5, 6, 7 and 8
9. Stop training when Eve decrypts plaintext like random guesses

of key generation of simulated chaos model. This method can well hide information about the chaotic map and increase the difficulty of decoding. Then the key and plaintext are entered into the GAN counter communication model. By means of confrontation training, an efficient encryption method that can eliminate chaotic cycles is obtained. In the test session, this article only showed the performance research of the random key generation by the adversarial algorithm and the partial leakage of the key.

In view of the problems in the model in Sect. 2.1, this section made some improvements, mainly through the replacement of the activation function and the enhancement of the neural network structure to strengthen the encrypted communication model.

Figure 1 shows that the activation function used in the basic model is ReLU. According to the property of it, the result is always 0 when the input is negative, so when the neuron is negative, this property will affect the weight update [6]. To solve this problem, the ELU activation function is selected in this paper to replace ReLU, which alleviates the phenomenon of large area neuron death during weight updating. In addition, it is

negative at x and has a small absolute value, which has good anti-interference ability. And the ELU is better suited for subsequent normalization operations. The comparison of the two is shown in Fig. 9.

Fig. 9. Comparison of ReLU function and ELU function.

The model described in Sect. 2.1 cannot communicate normally after the increase in the amount of information leaked, that is, neither Bob nor Eve can decrypt normally, and the decryption ability reaches its limit. Therefore, this paper enhanced the network model of Bob and Eve to improve their decryption ability. By adding the full connection layer, the decryption ability of Bob and Eve was increased synchronously. The activation function took the tanh function, and the structure of Alice's network remained unchanged.

In addition, to prevent the model from falling into the local optimal mode due to the rising stability, and thus the performance cannot be improved, normalization processing was added to the full connection layer in this paper to improve the network's ability to learn the optimal state. Through experimental verification, data normalization maps the value range of data to different regions, eliminating the problem that the accuracy of classification results is affected by the inconsistency of data size range. The improved Alice, Bob, and Eve network models are shown in Fig. 10.

The improved adversarial encryption algorithm based on GAN is shown in the following (Table 4).

Fig. 10. Improved network model of Alice, Bob and Eve.

4 Experimental Procedure

4.1 Experimental Environment

This experiment was conducted on the Window System, using TensorFlow as the network learning framework, and demonstrated the performance research of the random key generation by the adversarial algorithm and the partial leakage of the key. In the key generation stage, this article chose Adam optimizer, and the learning rate is 0.0008. A Mini-batch with M of 4096 was used in the encryption-model.

4.2 Key Security Analysis

FID can well evaluate the quality of the generated data. It can give the same judgment results as human vision, and the computational complexity of FID is not high. Therefore, FID is selected as the performance evaluation index of GAN in this paper. The formula of FID is as follows:

$$FID = \|\mu_r - \mu_g\|^2 + T_r \left(\sum r + \sum g - 2 \left(\sum r \sum g \right)^{1/2} \right) \quad (6)$$

The evaluation process is shown in Table 5.

In this paper, the distribution of chaotic map data generated by GAN is within the upper and lower bounds. Figure 11 statistics the FID distance between the generated samples and the real samples in the early training range, where the abscissa represents the number of training rounds, and the ordinate represents the FID value. It can be seen from Fig. 11 that as the training progresses, the FID value gradually becomes smaller, indicating that the performance of GAN through training is also continuously increasing.

To prevent attackers from attacking through fuzzy keys, the sensitivity of keys must be high enough, and chaotic systems can satisfy this characteristic well [7]. To verify the high sensitivity of the key generated by the key generator, $\mu = 4$ was set in the experiment. The initial value of $x_{01} = 0.256$ and $x_{02} = 0.264$ were selected and input into the model to observe the sensitivity of the period. The results show that, when the initial selection gap is small, even when the distribution generated after training is almost

Table 5. FID evaluation process.

FID evaluation process
1. Send the samples generated by the generation network and the samples generated by the discriminant network to the classifier
2. Abstract features of the middle layer of the classifier
3. Assuming that the abstract feature matches the Gaussian distribution, estimate the mean and variance of the generated and training samples
5. Calculate the distance between two Gaussian distributions and use this to evaluate the performance of GAN

Fig. 11. FID distance between generated sample and real sample.

fitted to the original distribution, the difference between the two distributions generated is still obvious. This indicates that the generated chaotic key has high sensitivity. Besides, a different key sequence is used for each encryption process, which further improves the security of the generated key. The test results are shown in Fig. 12.

Fig. 12. Training results of the key generator based on GAN.

4.3 Analysis of Encryption Model Performance

The security of the encrypted communication system is inseparable from the ability of the attacker. When the attacker has a strong ability, the model still needs to ensure the security of the communication as far as possible. Therefore, in order to verify the security of the model in this paper, we let Eve know the ciphertext and some keys simultaneously to test the model in this paper. The results are shown in Fig. 13.

Fig. 13. Results of the security evaluation of the encryption model.

As can be seen from the figure above, as the training progresses, both Bob and Eve can achieve convergence rapidly, and Eve's error during convergence is very high. At this time, it can be considered that the communication between Alice and Bob is safe. In addition, the encryption performance of the communication model is much better than that of the basic model when the amount of key leakage increases gradually. According to the test and analysis, the adversarial encryption algorithm based on generating the chaotic sequence by GAN is secure. After a certain number of rounds of training, the model tended to be stable, and the performance of the model in resisting attacks was also improved.

5 Conclusion

Based on referring to a lot of literature and based on the anti-encryption communication model, this paper introduced a chaos model to optimize the generation mode of key and proposes a counter-encryption model based on Logistic mapping. And analyze the security of the entire system through model analysis, key analysis, and other methods. Finally, it is concluded that the key to the encryption algorithm can be changed from time to time, and the periodic problems in chaotic encryption can be eliminated to a certain extent. In addition, compared with the basic anti-encryption communication model, the security of the encryption model is greatly improved.

References

1. Abadi, M., Andersen, D.G.: Learning to Protect Communications with Adversarial Neural Cryptography. ICLR (2017)

2. Raghunathan, A., Segev, G., Vadhan, S.: Deterministic public-key encryption for adaptively-chosen plaintext distributions. *J. Cryptol.* **31**(4), 1012–1063 (2018). <https://doi.org/10.1007/s00145-018-9287-y>
3. Lin, Z., Yu, S., Li, J.: Chosen ciphertext attack on a chaotic stream cipher. In: Chinese Control and Decision Conference (CCDC), pp. 5390–5394 (2018)
4. Ashish, Cao, J.: A novel fixed point feedback approach studying the dynamical behaviors of standard logistic map. *Int. J. Bifurcat. Chaos.* **29**(01) (2019)
5. Tramer, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., Mc Daniel, P.: Ensemble adversarial training: attacks and defenses. *EprintArxiv* (2017)
6. Jain, A., Mishra, G.: Analysis of lightweight block cipher FeW on the basis of neural network. In: Yadav, N., Yadav, A., Bansal, J.C., Deep, K., Kim, J.H. (eds.) *Harmony Search and Nature Inspired Optimization Algorithms*. AISC, vol. 741, pp. 1041–1047. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-0761-4_97
7. Purswani, J., Rajagopal, R., Khandelwal, R., Singh, A.: Chaos theory on generative adversarial networks for encryption and decryption of data. In: Jain, L.C., Virvou, M., Piuri, V., Balas, V.E. (eds.) *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*. AISC, vol. 1064, pp. 251–260. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-0339-9_20

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.