Chaos Theory on Generative Adversarial Networks for Encryption and Decryption of Data

Chapter · January 2020

DOI: 10.1007/978-981-15-0339-9_20

4 authors, including:

Rajesh Rajagopal
ABV-Indian Institute of Information Technology and Management Gwalior
26 PUBLICATIONS **731** CITATIONS

SEE PROFILE

Chaos Theory on Generative Adversarial Networks for Encryption and Decryption of Data

Juhi Purswani, Rajesh Rajagopal, Riya Khandelwal and Anuraj Singh

Abstract Today's world involves sharing a tremendous amount of vital information and data over the web and cloud for almost everything. Any hacker or cyber-terrorist can get access to the data and hence the security of the data becomes extremely essential. Through this research, the possibilities of improving the cryptosystem has been explored by making use of generative adversarial networks in which our own shared key, which is generated with the help of chaotic generator has been incorporated. The key formed leads to the increase in randomness which in turn makes it even more difficult to crack it, thus making the system even more secure.

Keywords Cryptography · Generative adversarial networks · Neural networks · Chaotic maps · Security · Keras · Pseudo-random number generator

1 Introduction

Artificial intelligence is a field that is growing at a very fast pace. One major issue that it still faces is the security of the data shared. Datasets are an integral part of the field. Since only large companies and MNCs are privileged with these, these companies are very secretive about sharing the dataset with others.

For the growth of the company, it is genuine that it will be interested to share its data with the outsiders like data scientists so that the latter could work upon it and

J. Purswani (✉) · R. Rajagopal · R. Khandelwal · A. Singh
Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior 474001, Madhya Pradesh, India
e-mail: juhi.purswani7@gmail.com

R. Rajagopal
e-mail: rajesh@iiitm.ac.in

R. Khandelwal
e-mail: riyakwl28@gmail.com

A. Singh
e-mail: anuraj@iiitm.ac.in

L. C. Jain et al. (eds.), *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Advances in Intelligent Systems and Computing 1064,
https://doi.org/10.1007/978-981-15-0339-9_20

use it to validate the models to find a particular solution, but one thing that is still in question is how the data will be protected?

One possible solution for this problem is data anonymization, since securing or hiding the sensitive parts will lead to some security. But this also does not make the data secure due to inferences, since some part of the secured data can still be predicted with the other part that is not anonymized.

Another solution that could be thought of is with the help of homomorphic encryption. The homomorphic encryption is still new in the market but has a great potential to solve the cryptography-related problems in the field of AI. The idea behind homomorphic encryption is that the processing of the data, calculations, and so on, can be done on the encrypted texts or data itself without the need to decrypt it first as in [1]. But the technology is still very new and computationally not very efficient and not differentiable. So it is very early to think of this as a solution.

The solution that the Google Brain team came up with is with the help of generative adversarial networks as in [2]. The idea was to use neural networks to develop a cryptographic system which makes the data secure from other neural models with the help of generative adversarial neural networks. They found ways in which neural networks can learn to implement cryptography without any prior knowledge of cryptographic algorithms. The results were outstanding considering that traditionally neural networks were never thought of to be used for cryptography purposes. But still the accuracy was less, so it could not be applied in real-time scenarios.

For the security of data, encryption has also been done with the traditional cryptographic algorithms by introducing chaos in the common key to make it more random so that it cannot be broken down by the hackers [3]. The concept is called chaotic cryptography. The system proposed was for images only and cannot be applied to text or any other type of data [4].

The other solutions proposed the method of chaotic neural network for encryption for either text or image. Image encryption using chaotic neural network and text encryption through neural networks was proposed [5, 6]. But it could be applied to image or text only hence made it unsuitable for other data types like video, sound and so on. Time-delayed chaotic neural networks were also used [7] which increase the strength of encryption but there was much discrepancy in decrypted text and plain text, hence making them unsuitable to be used in real-life scenario. M. Arvandi, S. Whu and W.W. Melek also proposed the method of using recurrent neural network for data encryption [8] and their method prove to be quite robust but lack the accuracy.

To garner the problems to the above-stated issue and provide the solution, this paper proposes a framework for generative adversarial networks-based cryptographic system in which the shared key will be generated with the help of a dynamical chaotic random number generator [9, 10] to tackle the issue of accuracy. The key generated here will be more random and therefore more secure, making the system stronger in

terms of security, thus providing a way in which the data can be secured from other neural models in the real world. Also encryption and decryption will be performed on bits; hence it could be applied to any data type.

2 Related Works

Already research is going on in exploring the possibilities of implementing cryptography with the help of neural networks. All of these researches had one thing in common that is the key that was formed was from the trained weights, in which the models were then synchronized to perform encryption. This section represents key utilities achieved using generative adversarial networks and chaotic generators in the similar fields.

2.1 Generative Adversarial Networks and Their Utilities

A generative adversarial neural network, that is, GAN is a much more modified version of neural network having a pair of models competing with each other for the purpose of learning, analysing and capturing the dataset variations.

- GAN architecture can be well understood by keeping in mind the terminology of encryptor and decryptor. In this particular terminology the generator G can be thought of as the encryptor whose work is to encrypt the data. The discriminator D can be thought of as an expert trying to find the plain text and differentiating between the plain text formed and original text by taking in both the key and encrypted text as input. Both the models learn from each others' losses in the training phase as they are trained together. The architecture of a general GAN is shown in Fig. 1.
- The model has two components mainly:

  1. Generator: Its work is to predict the features, when labels are being fed into it.
  2. Discriminator: Its work is exactly opposite to that of generator; it predicts labels, when being fed features.

- Use of GAN for cryptography: The GANs were first used for cryptography in Google Brain research work [2]. Here the possibility of using a neural network for cryptography came into picture which traditionally was always considered a bad idea. The models were trained adversarially to learn encryption and decryption techniques; no particular algorithm was used for the purpose. The neural networks learned themselves.

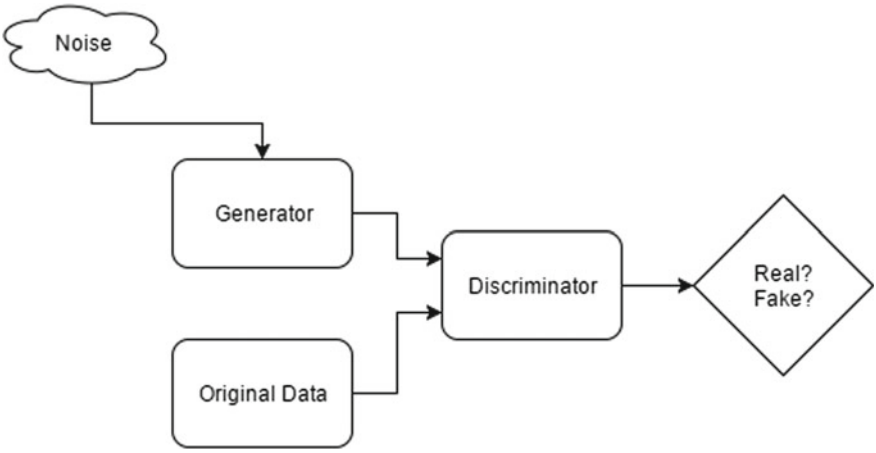

Fig. 1 General GAN architecture

2.2 Chaos Theory

Chaos theory is well defined as a nonlinear, complex dynamical system and is a branch of mathematics. This dynamical system is highly dependent on the initial properties and conditions because of which the outputs that are produced are very much random-like and highly unpredictable, but still easily reproducible and deterministic. This property becomes the main reason for the usefulness of chaotic maps in the field of secure communications as in [11].

- Chaos Dynamics for Encryption and Decryption: Chaotic cryptography is a field that makes use of the concepts of chaos theory and cryptography for the encryption purposes. Traditionally number theory or some algorithms have been used along with it to make encryption possible. The sensitivity of chaotic maps with their initial conditions is quite useful for the encryption purposes as in work [12].
- Pseudo-Random Number Generator with Chaotic Properties: One of the widely used methods for the purpose of generating pseudo-random sequences using chaos theory in a dynamical system is logistic map as in [9]. The sequence produced has been used for cryptography purposes but with traditional cryptographic algorithms only. Using it with GANs is totally a new dimension in the field of cryptography.

3 Proposed Framework

In order to use neural networks for cryptography in real-life scenarios, there should be no discrepancy between the plain text and the decrypted text. Thus the accuracy should be quite high.

The primary aim of this research is to develop a method that uses shared key with much more randomness and train a pair of neural networks, for stronger encryption of plain text and other for more accurate decryption of encrypted text by much less error in plain text and decrypted text. Our research has the following objectives:

(a) To generate more random bit sequence to be used as shared key
(b) To implement a GAN architecture for encryption and decryption of data in the form of bits
(c) And finally, to integrate the above so as to increase the accuracy and strength of the encryption.

By individually studying the three components, the complete overview of the system can be understood, namely:

(a) Generative adversarial neural network
(b) Chaotic pseudo-random number generator
(c) Mechanism and algorithm.

3.1 Generative Adversarial Network

Unlike the traditional way of training two models as generator and discriminator, here it is modified in a way that three neural networks will be used. A pair of neural network will work as a generator and third will be a discriminator in a modified way. The three neural networks will be:

(a) *Encryptor*: The input will be the plain text and the shared key, both in the binary sequence and generate the encrypted text.
(b) *Decryptor*: Input will be the encrypted text and the shared key to generate the decrypted text as output.
(c) *Eavesdropper*: It will take the input as encrypted text only, that is, it will intercept the text and will decrypt it without having the shared key.

Layers used in the generative adversarial network: All the three neural networks have the same architecture comprising the following layers:

(a) Fully connected dense layer
(b) Convolutional layer
(c) Flatten layer

So in total, we have used one dense layer, four convolutional layer and one flatten layer.

Replacing pooling layer with strided convolution: For down sampling the input matrix MaxPooling layer is used in between the convolution layers, which in turn decreases the total number of training parameters in network. But here instead of directly downsampling we have used strided convolutions so as to allow the network to learn its own spatial sampling.

Activation functions used: The encryption is performed on binary sequences, 0 and 1. So to normalize the output of every layer in $[-1, 1]$ we have used *tanh* activation in every layer except for the last layer where we have used *sigmoid* activation.

3.2 Chaotic Pseudo-Random Number Generator

To generate the shared key with chaotic properties, the chaos theory will be applied as chaotic mapping to generate the pseudo-random number generator. The numbers generated by this generator will then be mapped into the bit form 0 or 1.

(a) *Logistic mapping*: It is the one-dimensional iterative mapping [9], which produces chaos in the system by generating pseudo-random numbers. To obtain the value at given time 't', it uses '$t-1$' value, by the formula:

$$x_t = r.x_{t-1}.(1 - x_{t-1})$$
$$x_0 \in (0,\ 1)$$

To obtain all the future values in the range [0, 1], the initial value, that is value at time $t = 0$, takes the range (0, 1]. Also. the property of trajectory of the function depends on the parameter 'r' and experiments have shown that trajectory is chaotic when 'r' is strictly in the range [3.99, 4].

(b) *Henon map*: The second mapping scheme that was used for the pseudo-random number generation is the Henon map, which is a two-dimensional chaotic map. It is also an iterative map. The initial conditions are represented by the equations:

$$x_{k+1} = -\alpha x^2 + y_k + 1$$
$$y_{k+1} = \beta x_k,$$

where α and β have values 1.4 and 0.3, respectively [10]. To form a binary sequence, the outputs of these functions are mapped to 0, 1 based on the threshold value. It was found that the pseudo-random sequences formed were having great statistical properties.

Given the initial condition, the values obtained are deterministic, but considering their high susceptibility to the provided initial conditions which are chosen randomly in the above-stated range, the iterative values become unpredictable, hence making them suitable to be used as a random number to obtain the shared binary key.

3.3 Mechanism and Algorithm

The whole system is developed by implementing the following steps:

(a) For the given size of data, pseudo-random numbers were generated using chaotic mapping.
(b) These numbers were then mapped to 0,1 so as to get the random binary sequence. This binary sequence is then used as shared key in the neural network.
(c) For the neural network three architectures are laid, comprising three convolutional layers, one fully connected layer and one flatten layer with *tanh* as the activation function for all the layers except for the last layer where *sigmoid* function is used (Fig. 2).
(d) The data are converted into binary form and then fed into the networks
(e) The encryptor and decryptor are trained by freezing the weights of the eavesdropper. Encryptor takes the shared key and plain text, and decryptor takes the shared key and encrypted data from encryptor, all in bits.
(f) The output is stored and the re-constructional loss between plain text and decrypted data is calculated.
(g) This re-constructional loss is then fed into the eavesdropper for training by now freezing the weights of encryptor and decryptor both.
(h) The discriminator loss between plain text and output of eavesdropper is calculated.
(i) The same procedure is followed in every iteration with the goal of minimizing the re-constructional loss and maximizing the discriminator loss.

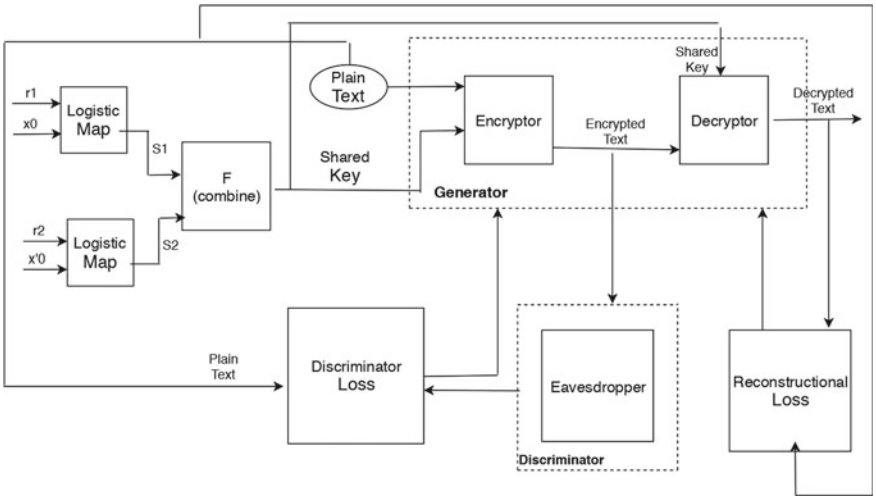

Fig. 2 System design and model overview

The pseudo-random numbers were generated with the above-stated logistic map and Henon map. So the above mechanism is followed individually with both mappings. The results obtained were then compared.

3.4 Simulation Results

The pseudo-random generator was implemented in python and the neural networks were implemented using Tensorflow keras api. Once the implementation process was completed, we took text data and tested our system. Following are the results drawn from the experiments, written according to the objectives:

(a) *Bitwise accuracy of the decrypted text*: The primary motive of our system was to decrease the discrepancy between the decrypted text and the plain so as to make the system suitable for real life applications.
As can be inferred, with the application of the chaos theory for the state-of-the-art system, the accuracy has increased.

(b) *Strength of the encrypted data*: The primary objective of the system is to provide a mechanism for the strong encryption of data so that it could not be decrypted by any other neural network. Carrying forward from the above-stated objective, we have therefore trained the discriminator as eavesdropper which try to decrypt the data without the use of shared key.
As per the analysis, it was found that the Henon key provides better security than logistic key for the given key size (Figs. 3 and 4, Tables 1 and 2).

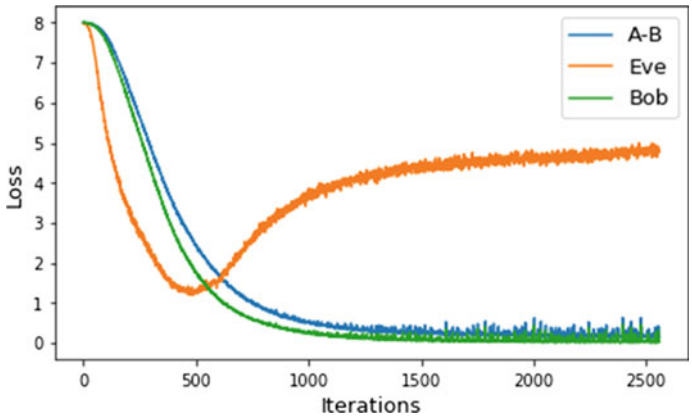

Fig. 3 Eavesdropper loss with logistic key

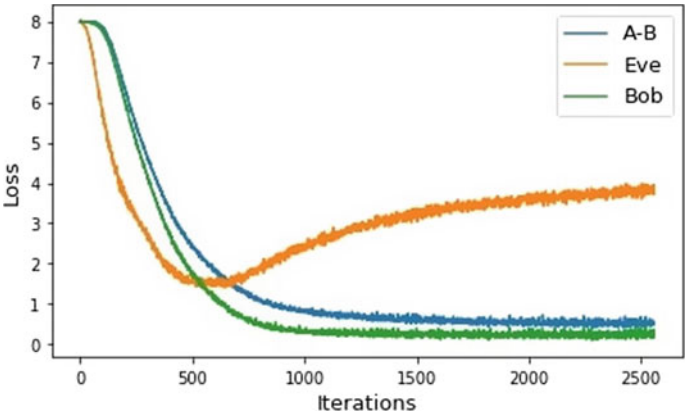

Fig. 4 Eavesdropper loss with Henon key

Table 1 Model accuracy

Sr. No.	Model	Accuracy (%)
1	Model formulated with python inbuilt random function	77.35
2	Model formulated with logistic chaotic key	86.74
3	Model implemented with Henon key	98.49

Table 2 Eavesdropper's loss

S. No.	Model	Loss
1	Without chaotic key	2.96
2	With logistic key	3.87
3	With Henon key	4.94

Higher loss indicates greater strength of encryption

4 Conclusion

A new methodology for encryption and decryption of data was proposed, which combines the benefit of chaos theory to produce shared key and generative adversarial network to perform strong encryption of data. The logistic map and Henon map were individually used to obtain the shared key. This shared key is then fed into the combination of three neural networks which are trained in a way to produce the strongly encrypted text.

Acknowledgements Funding agency: Science and Engineering Research Board (SERB), ECR/2015/000234.

References

1. T. Graepel, K. Lauter, M. Naehrig, ML Confidential: Machine Learning on Encrypted Data, Information Security and Cryptology ICISC 2012. ICISC 2012 (2013)
2. M. Abadi, D.G. Andersen, Learning to Protect Communications with Adverserial Neural Cryptography, Google Brain, October 24, (2016)
3. J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifurc. Chaos
4. A. Akhavan, A. Samsudin, A. Akhshani, A symmetric image encryption scheme based on combination of nonlinear chaotic maps. J. Frankl. Inst. 1797–1813 (2011)
5. M. Chauhan, R. Prajapati, Image encryption using chaotic based artificial neural network, Int. J. Sci. Eng. Res. (2014)
6. E. Volna, M. Kotyrba, M. Janosek, Cryptography based on neural network (Department of Informatics and Computers, 2005)
7. U. Wenwu, Z. Cao, Cryptography based on delayed chaotic neural networks (Department of Mathemetics, Southeast University, 28 March 2006)
8. M. Arvandi, S. Whu, W.W. Malik, Symmetric cypher design using recurrent neural networks, in *International Joint Conference on neural networks*, vol. 21 (2006)
9. A. Kanso, N. Samaoi, Logistic Maps for Binary Number Generation, (Departments of Mathematics and Computer Science, 30 October 2007)
10. M. Suneel, Cryptographic pseudo random sequences from the chaotic Henon Map, Defence Research and Development Organization, 9 April (2009)
11. T. Godhavari, N.L. Alamelu, R. Sundararajan, Cryptography using neural net- work, in *IEEE Indecon 2005 Conference*, (Chennai, India, 2005)
12. W. Yu, J. Cao, Cryptography based on delayed chaotic neural network. Phys. Lett. A (2006)