Symmetric Cryptography using Neural Networks

V. NAVEENA[1], Dr. S. SATYANARAYANA[2]

M.Tech Scholar[1], Senior Assistant Professor[2]
[1,2]Department of Computer Science & Engineering, Raghu Engineering College, Dakamarri(V), Visakhapatanam, AP, India.

---***---

Abstract - *In this paper, I show Neural Networks is capable of performing symmetric encryption in an adversarial setting and improve on the known literature on this topic. I also show that Neural Networks is capable of detecting known cryptographically insecure communication*

The parity machine (PM) is a neural network applied in cryptography to generate a secret key. It is also used for a key exchange protocol. The TPM network is in fact an FNN that has input layer neurons constructed in the McCulloh-Pitts model (Protić, 2015), (Dolecki and Kozera, 2015). In the second layer, the network has neurons with specific activation functions. The outcome of the output neuron is the results of the entire PM network. Each PM network is described by three parameters: the number of hidden neurons - K, the number of input neurons connected to each hidden neuron - N, and the maximum value for weight {-L, ... L}. A PM consists of KN random input elements xji = ±1, j = 1…N, K binary hidden units σi = ±1, i = 1,…, K, and one binary output unit τ=Πiσi, where σi is determined via the function σi = sign(Σjwjixji). A PM that has three neurons in the hidden layer (K=3) is called the three parity machine.

The advantage of neural cryptography is that the algorithm used in generating a secret key can be a simple perception of the Tree Parity Machine (TPM). Synchronization of TPMs by mutual learning only works if both machines receive a common sequence of (random) input vectors. For that purpose, each communication party uses a separate, but identical pseudo-random number generator (PRNG). By mutual learning, two parties synchronize their networks without transmitting inputs over a public channel. Having reached full synchronization, parties can authenticate each other by knowing weight vectors which are identical, and represent a secret key

Key Words: Symmetric encryption, Secret Key.

1. INTRODUCTION

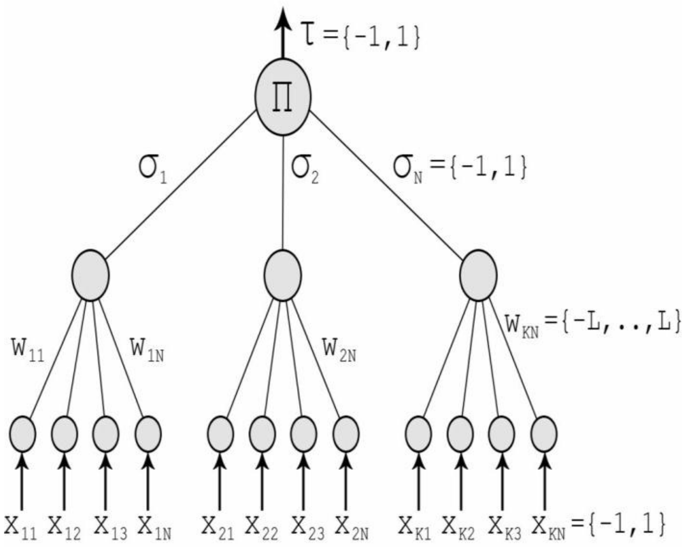
In case of neural cryptography, two identical dynamic systems, starting from different initial condition receive an identical input vector, generate an output bit and are trained based on the out bit. The dynamics of two networks and their weight vectors found exhibit a novel phenomenon, where network synchronize to a state with identical time dependent weights. This concept of synchronization by mutual learning can be applied to secret key exchange protocol over a public channel has been studied and generated key is used for encryption and decryption. The use of the synchronized neural networks is as key material for another algorithm or directly as a key stream as a stream cipher[1].

The encryption/decryption process is simple and faster. But the key distribution is problem. Public key cryptosystem solves the key distribution problem by using different secret key for internal user. Thus key exchange between users is not required and the security of such system is depends of computational complexity. Moreover the process involved in generating public key is very complex and time consuming. To overcome these disadvantage special characteristic of neural network can be used to generate common secret key over public channel.

2. Tree parity machine

The parity machine (PM) is a neural network applied in cryptography to generate a secret key. It is also used for a key exchange protocol. The TPM network is in fact an FNN that has input layer neurons constructed in the McCulloh-Pitts model (Protić, 2015), (Dolecki and Kozera, 2015). In the second layer, the network has neurons with specific activation functions. The outcome of the output neuron is the results of the entire PM network. Each PM network is described by three parameters: the number of hidden neurons - K, the number of input neurons connected to each hidden neuron - N, and the maximum value for weight {-L, ... L}. A PM consists of KN random input elements xji = ±1, j = 1…N, K binary hidden units σi = ±1, i = 1,…, K, and one binary output unit τ=Πiσi, where σi is determined via the function σi = sign(Σjwjixji). A PM that has three neurons in the hidden layer (K=3) is called the three parity machine.

The advantage of neural cryptography is that the algorithm used in generating a secret key can be a simple perception of the Tree Parity Machine (TPM). Synchronization of TPMs by mutual learning only works if both machines receive a common sequence of (random) input vectors. For that purpose, each communication party uses a separate, but identical pseudo-random number generator (PRNG). By mutual learning, two parties synchronize their networks without transmitting inputs over a public channel. Having reached full synchronization, parties can authenticate each other by knowing weight vectors which are identical, and represent a secret key.



3. NEURAL KEY EXCHANGE PROTOCOL

It is a key-exchange protocol which does neither use number theory nor a public key, but it is based on a learning process of neural networks: The two participants start from a secret set of vectors wA and wB without knowing the key of their partner. By exchanging public information the two keys develop to a common time dependent key wA (t) = wB (t), which can be used for authentication / key / hash is used to encrypt and decrypt a given message

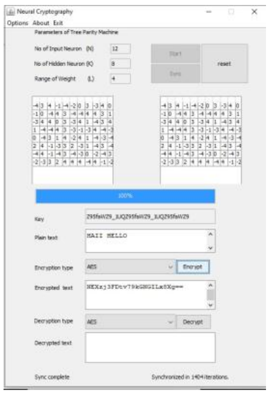
4. ONE TIME PASSWORD

One-Time Password (OTP) is always used as the strongest authentication scheme among all password-based solutions. The proposed a scheme that consists of three phases. With the common input vector, weight vectors of two TPMs are finally synchronized through mutual learning. The synchronized weight vector is then used as a seed to regenerate the input vector for the next step of learning. It is the rule for input vector (Iu and Is) updating.

5. Queries

Although neural key exchange algorithm for choosing the relevant inputs is sufficient to achieve a more or less secure key-exchange protocol, A and B could improve it by taking more information into account. Including queries in the training process of the neural networks. This means that alternately A and B are generating an input which is correlated with its state and A or B is asking the partner for the corresponding output bit. The overlap between input and weight vector is so low that the additional information does not reveal much about the internal states. But queries introduce a mutual influence between A and B which is not available to an attacking network E. In addition the method obtains a new (public) parameter which can be adjusted to give optimal security. In this work query incorporated to the case of the Hebbian training rule, which was successfully attacked using the majority of an ensemble of attackers.

6. RESULTS



Parameters of Tree Parity Machine

No of Input Neuron (N) 12
No of Hidden Neuron (K) 8
Range of Weight (L) 4

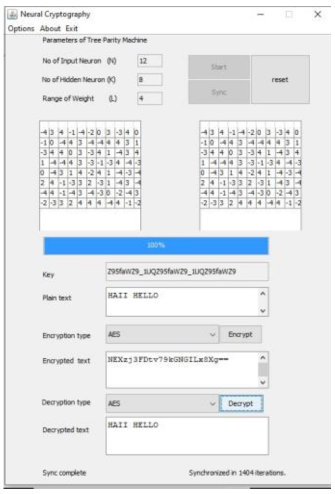
-4	3	4	-1	-4	-2	0	3	-3	4	0
-1	0	-4	4	3	-4	-4	4	4	3	1
-3	4	4	0	3	-3	4	1	-4	3	4
1	-4	-4	4	3	-3	-1	-3	4	-4	-3
0	-4	3	1	4	-2	4	1	-4	-3	-4
2	4	-1	-3	3	2	-3	1	-4	3	-4
-4	4	-1	-4	3	-4	-3	0	-2	-4	3
-2	-3	3	2	4	4	4	-4	4	-1	-2

-4	3	4	-1	-4	-2	0	3	-3	4	0
-1	0	-4	4	3	-4	-4	4	4	3	1
-3	4	4	0	3	-3	4	1	-4	3	4
1	-4	-4	4	3	-3	-1	-3	4	-4	-3
0	-4	3	1	4	-2	4	1	-4	-3	-4
2	4	-1	-3	3	2	-3	1	-4	3	-4
-4	4	-1	-4	3	-4	-3	0	-2	-4	3
-2	-3	3	2	4	4	4	-4	4	-1	-2

Key Z95faWZ9_1UQZ95faWZ9_1UQZ95faWZ9
Plain text HAII HELLO
Encryption type AES
Encrypted text NEXzj3FDtv79kGNGILx8Xg==
Decryption type AES
Decrypted text

Sync complete Synchronized in 1404 iterations.

As we entered the page we have to enter num of Input Neurons , num of Hidden Neuron, Range of Weight then Start the process then enter the Plain text then Sync the data key will be Generated .After that Encrypt .



Parameters of Tree Parity Machine

No of Input Neuron (N) 12
No of Hidden Neuron (K) 8
Range of Weight (L) 4

-4	3	4	-1	-4	-2	0	3	-3	4	0
-1	0	-4	4	3	-4	-4	4	4	3	1
-3	4	4	0	3	-3	4	1	-4	3	4
1	-4	-4	4	3	-3	-1	-3	4	-4	-3
0	-4	3	1	4	-2	4	1	-4	-3	-4
2	4	-1	-3	3	2	-3	1	-4	3	-4
-4	4	-1	-4	3	-4	-3	0	-2	-4	3
-2	-3	3	2	4	4	4	-4	4	-1	-2

-4	3	4	-1	-4	-2	0	3	-3	4	0
-1	0	-4	4	3	-4	-4	4	4	3	1
-3	4	4	0	3	-3	4	1	-4	3	4
1	-4	-4	4	3	-3	-1	-3	4	-4	-3
0	-4	3	1	4	-2	4	1	-4	-3	-4
2	4	-1	-3	3	2	-3	1	-4	3	-4
-4	4	-1	-4	3	-4	-3	0	-2	-4	3
-2	-3	3	2	4	4	4	-4	4	-1	-2

Key Z95faWZ9_1UQZ95faWZ9_1UQZ95faWZ9
Plain text HAII HELLO
Encryption type AES
Encrypted text NEXzj3FDtv79kGNGILx8Xg==
Decryption type AES
Decrypted text HAII HELLO

Sync complete Synchronized in 1404 iterations.

After that we are decrypt the data after the decryption then we can RESET the process. We have also been using not only AES but another 2 sample ones.

7. CONCLUSION

Mutual learning is used for synchronization between two parties that communicate across a public or private channel. During the synchronization, both communication parties use the same tree parity machines, receive common binary inputs generated randomly from the same random number generator, and exchange output bits. Adjusting the weights according to the learning rules leads to full synchronization in a finite number of steps. Networks trained on their mutual inputs synchronize to an identical time dependent weight vectors. This phenomenon is used to generate a secret key.

8. REFERENCES

[1] Navita Agarwal, Prachi Agarwal, "Use of Artificial Neural Network in the Field of Security", MIT International Journal of Computer Science & Information Technology, Vol. 3, No. 1, 42-44, 2013.

[2] Ajit Singh, Havir Singh, "Cryptography for Secret Key Exchange and Encryption with AES", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue. 5, 376-381, 2013.

[3] Siddeq Y. Ameen, Ali H. Mahdi, "AES Cryptosystem Development using Neural Networks", Inernational Journal of Computer and Electrical Engineering, Vol. 3, No. 2, 315-318, 2011.

[4] Eva Volna, Martin Kotyrba, Vaclav Kocian, Michal Janosek, "Cryptography based on neural network", Proceedings 6th European Conference on Modelling and Simulation, 2012.

[5] N. Prabakaran, P. Vivekanandan, " A New Security on Neural Cryptography with Queries", International Journal of Advanced Networking and Apllication (IJAIA), Vol.2, No. 1, 60-69, 2011.

[6] Wolfgang Kinzel, IdoKanter, "Neural Cryptography", Proceedings TH2002 Supplement, Vol. 4, 147-153, 2003.

[7] Einat Klein, Rachel Mislovaty, Idokanter, Andreas Ruttor, Wolfgang Kinzel, "Synchronization of neural network by mutual learning and its application to cryptography", International Proceeding of: Advances in Neural Information Processing System 17, Neural Information Processing System NIPS, 2004.