Cryptography Using Artificial Neural Network

Sawsan S. Abed*

Department of Chemistry, College of Education/Ibn-Al-Haithem*

Abstract

Neural cryptography deals with the problem of “key exchange” between two neural networks by using the mutual learning concept. The two networks exchange their outputs (in bits) and the key between two communicating parties ar eventually represented in the final learned weights, when the two networks are said to be synchronized. Security of neural synchronization is put at risk if an attacker is capable of synchronizing with any of the two parties during the training process.

التشفير باستخدام الشبكات العصبية

سوسن صاحب عبد
قسم الكيمياء / كلية التربية ابن الهيثم / جامعة بغداد

الخلاصة

التشفير بطريقة الشبكة العصبية تتعامل مع مشكلة تبادل المفاتيح ما بين شبكتين عصبيتين باستخدام مفهوم التعلم العصبي المتبادل ، الشبكتين تتبادل الاخراجات و المفاتيح فيما بينهم تتمثل بالأوزان التعليمية النهائية ، متى ما كانت هذه الشبكات متزامنة كان من الصعوبة على المهاجم أن يخترق النظام أو يتزامن معهم أثناء الفترة التعليمية.

Key words: Neural Network, Cryptography, Stream Cipher, Tree Parity Machine (TPM), Symmetric Key Encryption

1. INTRODUCTION

Nowadays information security has become an important aspect in every organization. In other words, the people have to be assured that the information to be read by only the sender and receiver. The basic need to provide security is using cryptography. In our work we are combining neural network and cryptography.

2. Artificial neural networks & stream cipher.

Artificial neural networks ar the principles of finding the decision automatically by calculating the appropriate parameters (weights) to make the compatibility of the system and this is very important to have the keys that used in stream cipher cryptography to make the overall system goes to high security .

2.1 Neural network

Artificial neural networks have motivated from their inception by the recognition that the brain computes in an entirely different way from the conventional digital computer. The brain contains billions of neurons with massive interconnections. Similarly, neural networks are massively parallel-distributed processors that are made up of artificial neurons with interconnections. These are nonlinear dynamic machines which expand the expression of input data as a linear combination of inputs to synapses and then perform a nonlinear transformation to compute output [1].

2.2 Stream cipher

Information security is generally accepted to be essential to modern business and technology, both for privacy of transactions and communications, as well as for defense against malicious intruders. Cryptography is the study of information security and the feasibility of communication over an insecure channel while preserving the secrecy of the information transmitted [2], [3]. Cryptographic techniques (such as stream cipher which will be used in this research) should offer at least the following three security features concerning data transmission: confidentiality, authentication and integrity. Confidentiality is fundamental third parties are expected to see the encrypted data but should not be able to decipher it. Authentication methods allow the receiver to verify that the sender is legitimate. Lack of authentication makes systems vulnerable to fraudulent transactions and denial of service attacks. Integrity of the transmitted data must be verifiable, i.e., the receiver should be able to check that no part of the message was lost or altered during transmission. As the sophistication of cryptanalytic attacks increases and their cost decreases, there is constant pressure to improve cryptographic methods on all three of these fronts [4], [5].

2.3 Related Literature

- A thorough literature survey indicates that there has been an increasing interest in the application of different classes of neural networks to problems related to cryptography in the past few years [6].
- The relationship between cryptography and machine learning in general and neural networks in particular has studied in [7], [8].
- Recent works have examined the use of neural networks in different layers of cryptosystems. Typical examples include key management, generation and exchange protocols; design of pseudo random generators; prime factorization; hash functions; symmetric ciphers; authentication; and authorization.
- A considerable number of studies have presented the innovative use of the Tree Parity Machine (TPM) in cryptography [9]–[15]. Kinzel and Kanter proposed and analytically studied a neural cryptography scheme which was based on the TPM and the mutual learning process between two parity feed-forward neural networks with discrete and continuous weights. The synchronization process is non-self-averaging and the analytical solution is based on random auxiliary variables. The main advantage of their approach is the generation of symmetric key exchange over a public channel using chaos synchronization of Tree Parity Machines [16], [17]. The learning time of an attacker that is trying to imitate one of the networks has examined analytically and reported to be much longer than the synchronization time.
- Kilmov et al. [18] have shown that Kinzel’s protocol can be broken by geometric, probabilistic, and genetic attacks, and such as is not entirely secure.
- J. Zhou et al. [19] have also demonstrated that this protocol is not completely secure under regular and majority flipping attacks, and have proposed a scheme to improve the security against flipping attacks by splitting the mutual information and the training process.
- Mislovaty et al. [20] reported a new attack strategy involving a large number of cooperating attackers that succeeds in revealing the encryption key. Attempts to restore the security against cooperating attacks are presented in [21].

2.4 purpose of the research

This research is very important to implement the secure system in digital communication and internet application or any system deal with data transportation and to reject the attacker in uncomplicated and inexpensive hard ware system.

3. Artificial Neural Network (ANN)

Artificial Neural Network is an information processing and modeling system which mimics the learning ability of biological systems in understanding unknown process or its behavior [5].

ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well [7].

ANNS have developed as generalizations of mathematical models of human cognition or neural biology. Based on the assumptions that [4]:

1. Information procures at many simple elements called neuron.
2. Signals are passed between neurons over connection links.
2. Each connection link has associated weight. Which in a typical neural net multiplies the signal transmitted?
4. Each neuron applies an activation function usually nonlinear to its net input (sum of weighted input signals) to determine its output signal.

An Artificial Neural Network is a network of many very simple processors (units), each possibly having a (small amount of) local memory. The units are connected by unidirectional communication channels which carry numeric data. The units operate only on their local data and on the inputs they receive via the connections. The design motivation is what distinguishes neural networks from other mathematical techniques: A neural network is a processing device, either an **algorithm**, or **actual hardware**, whose design was motivated by the design and functioning of human brains and components .

There are many different types of Neural Networks, each of which has different strengths particular to their applications. The abilities of different networks can be related to their **structure**, **dynamics** and **learning methods** [5].

3.1 Architecture of Neural Networks

To characterize a ANN, it is necessary to specify the number of neurons, how they are interconnected and the processing that takes place throughout the network.

The manner in which the neurons of neural of network are structured intimately linked with the learning algorithms used to train the network. In general , be identified : single layer feed forward network , multi layer feedback networks, and recurrent network figure (1) shows the architecture of ANN.

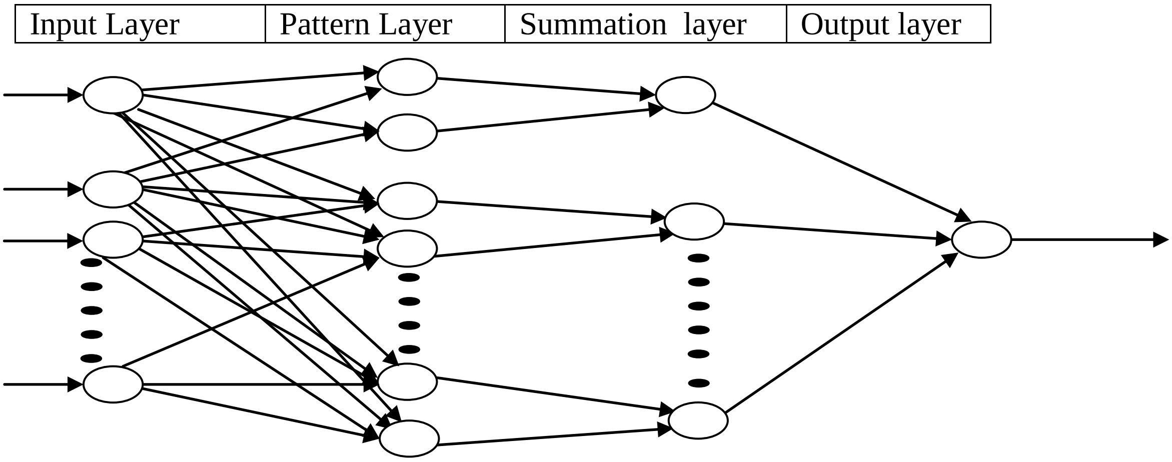


Figure (1) : Architecture of ANN

3. 2 Supervised Learning

Which incorporates an external teacher, so that each output unit is told what itis desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning.

An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error.

4.Cryptography

Cryptography is the science of providing secure services. Until 1970s cryptography was considered the domain of military and governments only. However the ubiquitous use computers and the advent of internet has made it an integral part of our daily lives. Today cryptography is at the heart of many secure applications such as online banking, online shopping, online government services such personal income taxes, cellular phones, and wireless LANS (Local Area Networks) etc. In this paragraph we provide an introduction to some cryptographic primitives which are used to design secure applications.

4.1 Requirements for Cryptography

Cryptography is generally used in practice to provide four services: *privacy, authentication, data integrity* and *non-repudiation*. The goal of privacy is to ensure that communication between two parties remain secret. This often means that the contents of communication are secret; however in certain situations the of fact communication took place and must be a secret as well. Encryption is generally used to provide privacy in modern communication. Authentication of one or both parties during a communication is required to ensure that information is exchanged with the legitimate party. Passwords are common examples of one-way authentication in which users authenticate themselves to gain access to system.

4.2 Symmetric Key Encryption

In symmetric key encryption a secret key is shared between the sender and receiver. The word "symmetric" refers to the fact that both sender and receiver use the same key to encrypt and decrypt the information.

Block Ciphers: A block cipher is symmetric key cryptographic primitive which takes as input an *n*-bit block of plaintext and a secret key and outputs an *n*-bit block of cipher text using a fixed transformation. Figure (2) shows the general structure of a block cipher. The common block sizes are 64 bits, 128 bits and 256 bits. For a fixed key the block cipher defines a permutation on the *n*-bit input.

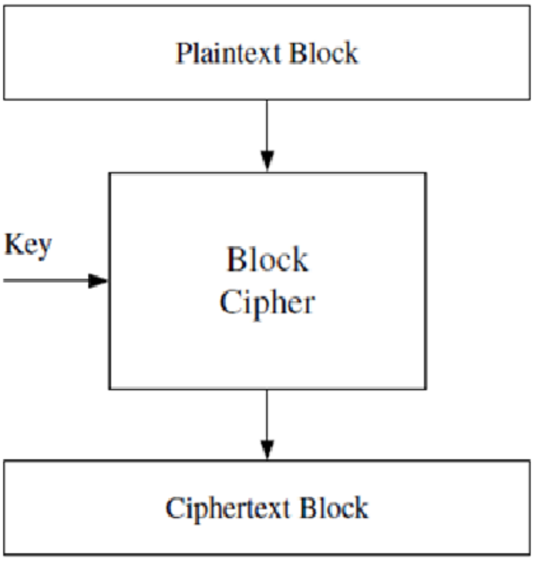


Figure (2) general structure of a block cipher

4.3 (Neural cryptography), Why using neural with cryptography

This idea came from the principles of make the keys that used in the (encryption and decryption) are hidden to any attacker want to break this system because of the breaking to many cryptographically system by neural, genetics and artificial intelligence attacker, so the need for the smart system is very urgent.

5. Own research design, Tree Parity Machine

The tree parity machine is a special type of multi-layer feed-forward neural network. It consists of one output neuron, K hidden neurons and KN input neurons. Inputs to the network are binary; the weights between input and hidden neurons take the values. Output value of each hidden neuron is calculated as a sum of all multiplications of input neurons and these weights, if the scalar product is 0, the output of the hidden neuron is mapped to -1 in order to ensure a binary output value. The output of neural network has computed as the multiplication of all values produced by hidden elements.

5.1 Protocol of Synchronization

Synchronization is the point of starting of the system each party A and B uses its own tree parity machine. Synchronization of the tree parity machines as shown in figure (3).

After the full synchronization has achieved the weights of both tree parity machines are the same, A and B can use their weights as encryption or decryption keys.

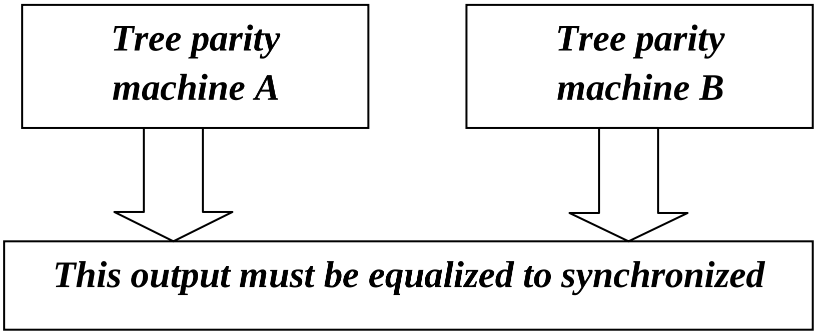


Figure (3) Two Tree parity machines

5.2 Attacks And Security Of This Protocol

In every attack it has considered, that the attacker E can eavesdrop messages between the parties A and B, but does not have an opportunity to change them. Learning with own tree parity machine one of the basic attacks can be provided by an attacker, who owns the same tree parity machine as the parties A and B. He wants to synchronize his tree parity machine with these two parties. In each step there are three situations possible:

1) Output A ≠ Output B: None of the parties updates its weights.
2) Output A = Output B = Output E: All the three parties update weights in their tree parity machines.

3) Output A = Output B ≠ Output E: Parties A and B update their tree parity machines, but the attacker cannot do that. Because of this situation his learning is slower than synchronization of parties A and B. It has proven, that synchronization of two parties is faster than learning of attacker. It can be improved by increasing of the synaptic depth L of the neural network. That gives this protocol enough security and an attacker can find out the key only with small probability.

For conventional cryptographic systems, we can improve the security of protocol by increasing of the key length. In the case of neural cryptography, we improve it by increasing of the synaptic depth L of the neural networks.

6. Proposed System

Because of this effect neural synchronization can be used to construct a cryptographic key-exchange protocol. Here partners benefit from mutual interaction, so that a passive attacker is usually unable to learn the generated key in time.

6.1 The description of system

As shown in the figure (4) the system stages have several logic states:-

1) Initialize random weight values.
2) Execute the following steps until the full synchronization has achieved
3) Generate random input vector X.
4) Compute the values of hidden neurons.
5) Compute the value of output neuron.
6) Compare the values of both tree parity machines.
7) Outputs are others: go to step 2.1.
8) Outputs are same.

6.2 Synchronization of neural networks

Neural networks learn from examples, when the networks teacher as well as student have N weights, the training process needs the order of N examples to obtain generalization abilities. This means, that after the training phase the student has achieved some overlap to teacher, their weight vectors hare correlated. As a consequence, student can classify an input pattern which does not belong to the training set. The average classification error decreases with the number of training examples.

Training can be performed into two different modes: Batch and on-line training. In the first case all examples are stored and used to minimize the total training error. In the second case only one new example has used per time step and then destroyed. Therefore on-line Training may be considered as a dynamic process: at each time step teacher creates a new example which student uses to change its weights by a tiny amount.

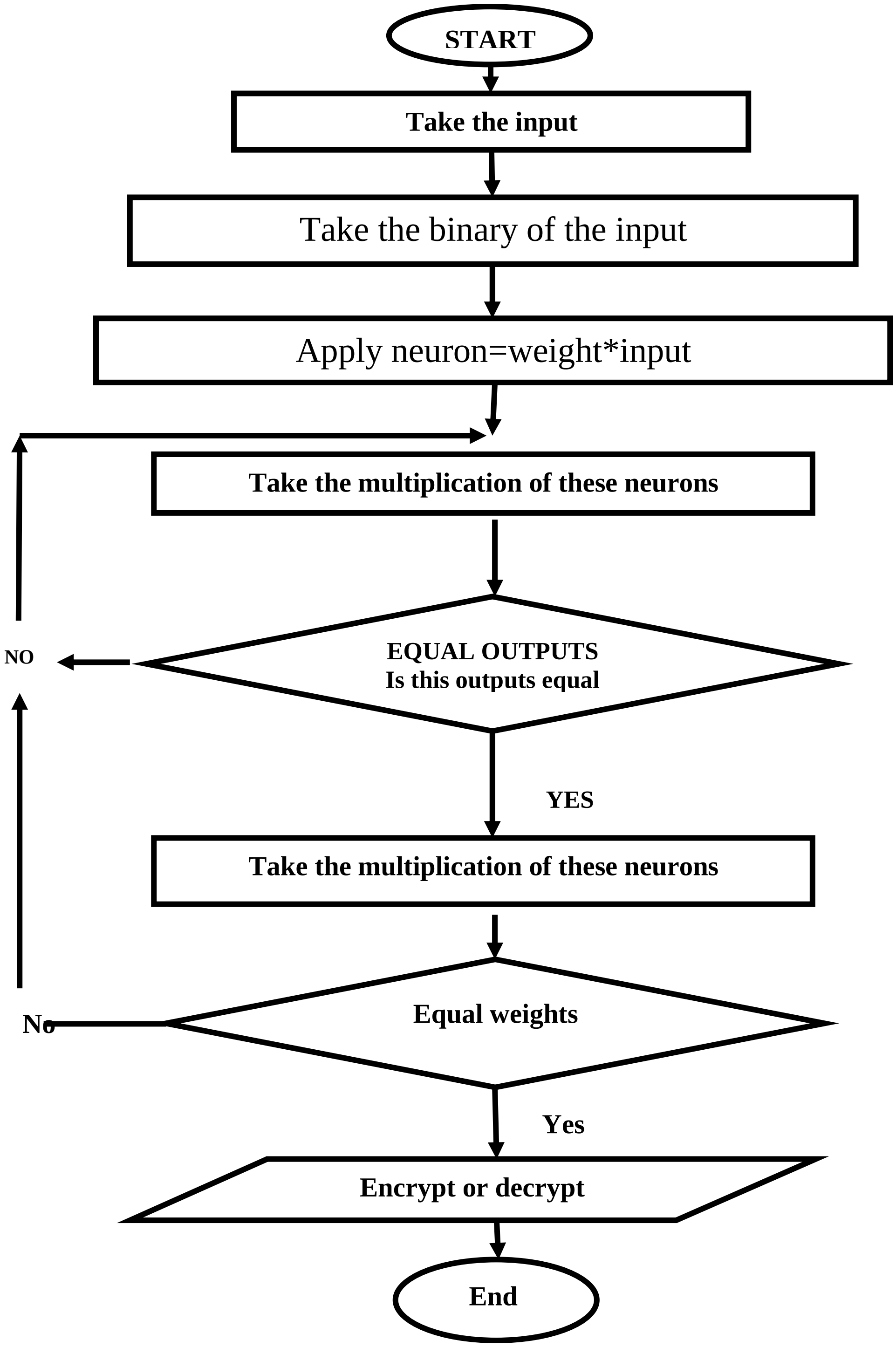

START
Take the input
Take the binary of the input
Apply neuron=weight*input
Take the multiplication of these neurons
EQUAL OUTPUTS
Is this outputs equal
NO
YES
Take the multiplication of these neurons
Equal weights
No
Yes
Encrypt or decrypt
End

Figure (4) the description of the system

6.3 Mutual Hebbian Learning

Hebbian theory concerns with how neurons may connect themselves to become anagrams. Hebb's theories are on the form and function of cell assemblies which can be understood from the following:

We start by presenting the process of mutual learning for a simple network: Two perceptions receive a common random input vector x and change their weights w according to their mutual bit σ, as sketched in Fig (5).

Both networks have trained to the examples generated by their partner and finally obtain an anti parallel alignment. Even after synchronization the networks keep moving.

We want to apply synchronization of neural networks to cryptography. In the previous section we have seen that the weight vectors of two perceptions learning from each other can synchronize. The new idea is to use the common weights w^A = - w^B as a key for encryption [11].

Now the components of the random input vector x are binary $x_i \in \{+1,-1\}$. If the two networks produce an identical output bit σ^A =σ^B, then their weights move one step in the direction -$x_i\sigma^A$.

But the weights should remain in the interval (8), therefore if any component moves out of this interval |w_i| =L+ 1, it sets back to the boundary w_i =± L.

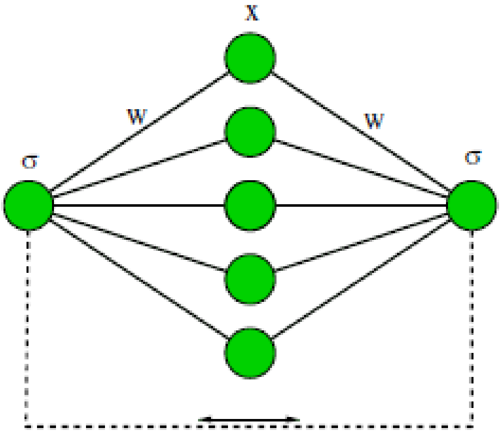


Figure (5) Two perceptions receive identical input x and learn their mutual output bits σ.

A single perception transmits too much information. the attacker, who knows the set of input/output pairs, can derive the weights of the two partners. On one hand, the information should be hidden so that attacker does not calculate the weights, but on the other hand enough information should be transmitted so that the two partners can synchronize. We found that multilayer networks with hidden units may be candidates for such as task [11].More precisely, we consider a Tree Parity Machine (TPM), with three hidden units as shown in Fig.6.

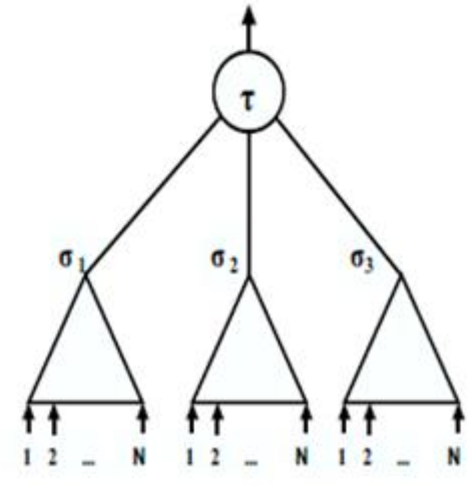


Figure (6) Triple Tree parity machine

Each hidden unit is a perception (1) with discrete weights (8). The output bit T of total network is the product of three bits of the hidden units.

For conventional cryptographic systems, we can improve the security of the protocol by increasing of the key length. In the case of neural cryptography, we improve it by increasing of the synaptic depth L of neural networks.

6.4 Benefits of neural key generation.

Benefits are neutrally generation of the keys can be approved by the following:-

1-In which a large population of attackers has trained, and every new time step each attacker is multiplied to cover the 2^{K-1} possible internal representations of$\{\sigma_i\}$ for the current output.

2-As dynamics precedes successful attackers stay while the unsuccessful have removed. *The Probabilistic Attack*, in which the attacker tries to follow the probability of every weight element by calculating the distribution of local field of every input and using the output, which is publicly known.

3-As mentioned in paragraph two about the learning conception that the weight adaptation achieved randomly, so this randomization is unknown to the attacker, this ideas lead to far away region of adaptation.

7. Software implementation by MATLAB program

The using of MATLAB is because of its high speed and performance in memory managements.

As the following figure(7) GUI introduce the program that determine the weights of stream cipher as mentioned earlier.

Figure (7) GUI introduce the program

Press Enter to go to the two types of parity machine or press Exit to end the program then go to figure (8).

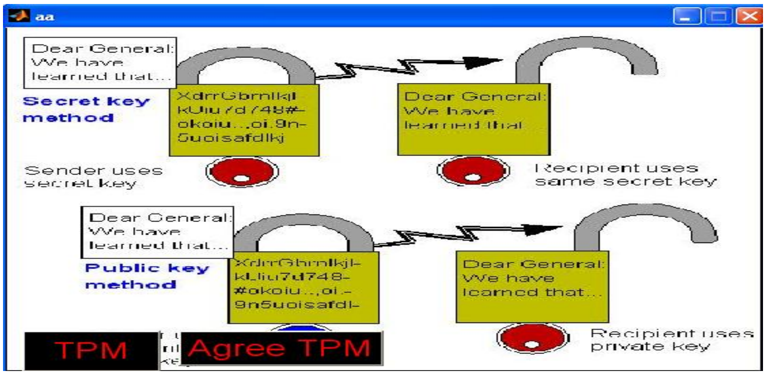
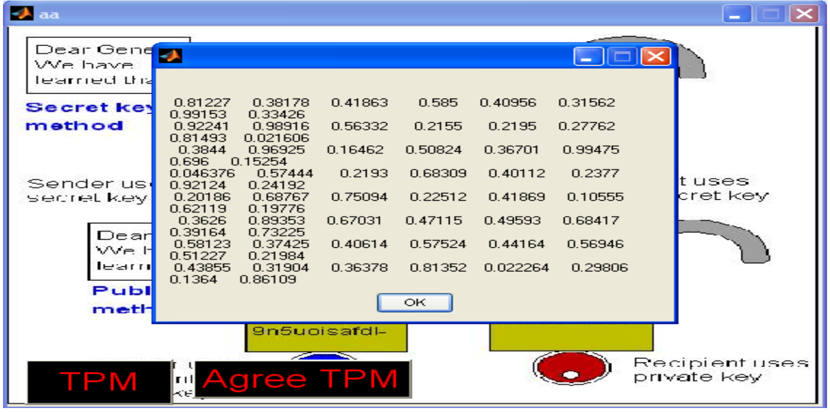

aa
Dear General: We have learned that...
Secret key method
XdrrGbrnlkjl kUiu7d748#-okoiu..,oi.9n-5uoisafdlkj
Dear General: We have learned that...
Sender uses secret key
Recipient uses same secret key
Dear General: We have learned that...
Public key method
XdrrGbrnlkjl-kUiu7d748-#okoiu..,oi.-9n5uoisafdl-
Dear General: We have learned that...
Recipient uses private key
TPM
Agree TPM

Figure (8) TPM algorithm

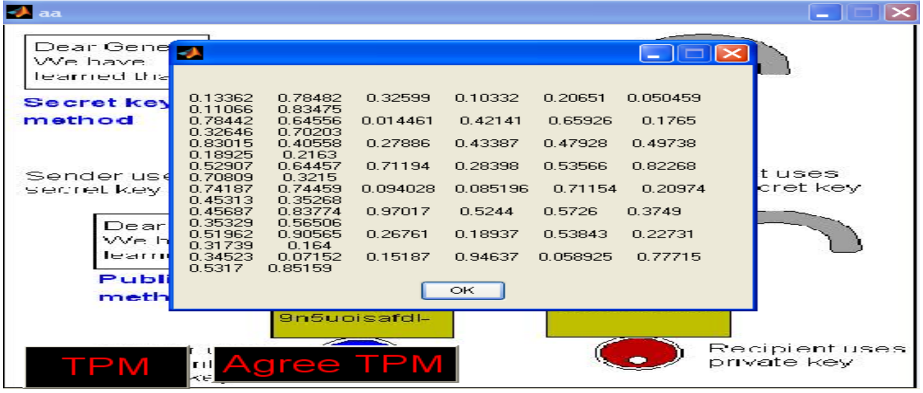
Press TPM to find the as the TPM algorithm to find the following table.



```
 0.81227    0.38178    0.41863     0.585    0.40956    0.31562
0.99153    0.33426
 0.92241    0.98916    0.56332    0.2155     0.2195    0.27762
0.81493   0.021606
  0.3844    0.96925   0.16462   0.50824    0.36701    0.99475
0.696   0.15254
0.046376    0.57444     0.2193    0.68309    0.40112    0.2377
0.92124    0.24192
 0.20186    0.68767    0.75094    0.22512    0.41869    0.10555
0.62119    0.19776
  0.3626    0.89353   0.67031   0.47115   0.49593    0.68417
0.39164    0.73225
 0.58123    0.37425    0.40614    0.57524    0.44164    0.56946
0.51227    0.21984
 0.43855    0.31904    0.36378    0.81352   0.022264    0.29806
0.1364    0.86109
```

Figure (9) TPM algorithms (find the following table).

Or press Agree TPM find the as the TPM algorithm to find the following table.



```
0.13362   0.78482   0.32599   0.10332   0.20651   0.050459
0.11066   0.83475
0.78442   0.64556   0.014461   0.42141   0.65926   0.1765
0.32646   0.70203
0.83015   0.40558   0.27886   0.43387   0.47928   0.49738
0.18925    0.2163
0.52907   0.64457   0.71194   0.28398   0.53566   0.82268
0.70809    0.3215
0.74187   0.74459   0.094028   0.085196   0.71154   0.20974
0.45313   0.35268
0.45687   0.83774   0.97017    0.5244   0.5726   0.3749
0.35329   0.56506
0.51962   0.90565   0.26761   0.18937   0.53843   0.22731
0.31739    0.164
0.34523   0.07152   0.15187   0.94637   0.058925   0.77715
0.5317   0.85159
```

Figure (10) TPM algorithms (find the following table).

8.Conclusion

Artificial Neural Networks is a simple powerful technique which has the ability to emulate highly complex computational machines. In this project, we have used this technique.

A comparative study has done between two different neural network architectures and their merits/demerits are mentioned. ANNs can be used to implement much complex combinational as well as sequential circuits.

Data security is a prime concern in data communication systems. The use of ANN in the field of Cryptography has investigated by using two methods. A sequential machine based method for encryption of data is designed. Also, a neural network for digital signal cryptography is analyzed. Better results can be achieved by improvement of code or by use of better training algorithms. Thus, Artificial Neural Network can be used as a new method of encryption and decryption of data.

References

- [1] S. Haykin. *Neural Networks: a Comprehensive Foundation*. 2nd edition, Prentice Hall, 1998.
- [2] R. Stinson. *Cryptography, Theory and Practice*, 2nd edition, CRC Press, 2002.
- [3] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. Handbook of Applied Cryptography,CRC Press, 1997.
- [4] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Prentice Hall, 2003.
- [5] B. Schneier. *Applied Cryptography*. John Wiley & Sons Inc., New York, 1996.
- [6] A. Sadeghian, M. Zamani, and V. Akbarzadeh, “Applications of Artificial Neural Networks in Cryptography—A Survey,” in progress.
- [7] R.L. Rivest, “Cryptography and Machine Learning,” Advances in Cryptology, Lecture Notes In Computer Science, vol. 739, pp. 427–439, 1991.
- [8] D. Pointcheval, “Neural Networks and Their Cryptographic Applications,” in *Proc. Of Eurocode*, pp. 183–193, 1994.
- [9] W. Kinzel, and I. Kanter, “Neural Cryptography,” *Proc. of the 9th Int’l Conf. on Neural Information Processing (ICONIP’02)*, vol. 3, pp. 1351–1354, 2002.
- [10] A. Ruttor, W. Kinzel, L. Shacham and I. Kanter, “Neural cryptography with feedback,” *Phys. Rev. E.*, (69): 046110—1-7, 2004.
- [11] R. Mislovaty, E. Klein, I. Kanter, and W. Kinzel, “Public Channel Cryptography by Synchronization of Neural Networks and Chaotic Maps,” *Phys. Rev. Lett., (91), 118701-1-4*, 2003.
- [12] L.N. Shacham, E. Klein, R. Mislovaty, I. Kanter, and W. Kinzel, “Cooperating attackers in neural cryptography,” *Phys. Rev. E.*, (69), 066137-1-4, 2004.
- [13] R. Mislovaty, Y. Perchenok, I. Kanter, and W. Kinzel, “Secure key-exchange protocol with an absence of injective functions,” *Phys. Rev. E.*, (66): 066102-1-5, 2002.
- [14] W. Kinzel and I. Kanter, “Interacting Neural Networks and Cryptography,” *Advances in Solid State Physics*. B. Kramer (ed.), vol. 42, pp. 383–391, Berlin: Springer, 2002.
- [15] M. Rosen-Zvi, I. Kanter and W. Kinzel, “Cryptography based on neural networks—analytical results,” *J. Phys. A: Math. Gen.*, vol. 35, no. 47, pp. 707–713, 2002.

- [16] M. Volkmer and S. Wallner, “Tree Parity Machine Rekeying Architectures,” *IEEE Trans. Computers*, vol. 54, no. 4, pp. 421–427, April 2005.
- [17] E. Klein, R. Mislovaty, I. Kanter, A. Ruttor, and W. Kinzel, “Synchronization of neural networks by mutual learning & its application to cryptography,” *NIPS*, pp. 689–696, 2004.
- [18] A. Klimov, A. Mityaguine, and A. Shamir, “Analysis of Neural Cryptography,” *AsiaCrypt 2002*, pp. 288–298, 2002.
- [19] J. Zhou, Q. Xu, W. Pei, Z. He, and H. Szu, “Step to improve neural cryptography against flipping attacks,” *Int. J. Neural Syst.*, vol. 14, no. 6, pp. 393–405, Dec. 2004.
- [20] R. Mislovaty, E. Klein, I. Kanter, and W. Kinzel, “Security of neural cryptography,” *11th IEEE Int’l Conf. on Electronics, Circuits and Systems*, pp. 219–221, pp. 13–15, 2004.
- [21] A. Ruttor, W. Kinzel and I. Kanter, “Neural cryptography with queries,” *J. Stat. Mech.*, P01009, 2005