Eng. & Tech. Journal, Vol.30, No.11, 2012

Semi – Chaotic Mutual Learning Platform for Key – Exchange Algorithm Using Neural Network

Dr.Enas H. Salih
Software Engineering Department, AL-Rafidain University College/ Baghdad
Email:Enasammar@Yahoo.com
Mohamad AB. Saleh
Management & conomy Collage, University of Iraqi/ Baghdad
Email: mohamadABSaleh@Yahoo.com
Mohammed Gheni Alwan
Computer Science Department, University of Technology/ Baghdad
Email:mgaz_Mgaz@Yahoo.Com

Received on: 30 /11/2011& Accepted on: 5/4/2012

ABSTRACT

Neural network has been emerged the cryptography field as efficient tool for both cryptanalysis and cryptography due to its amazing ability to explore solution space for a given problem. One of the latest observations for the behavior of neural networks is its ability to synchronize itself to other neural network based on mutual learning rules; this phenomenon has been under the focus of specialist in cryptographic field due to its significant usage as highly secure key exchange algorithm.

This paper is presenting new approach to drive the synchronization based on semi-chaotic mutual learning, where the output of each neural network will be extracted through non-linear mapping to memory filled with balanced number of 1's and 0's as this paper will demonstrate.

Keywords: Neural Network, Synchronization, Cryptography, Mutual learning, Chaotic. Key exchange

التعلم المتبادل المعتمد على شبه العشوائية لانظمة تبادل المفاتيح باستخدام الشبكات العصبية

الخلاصة

لقد ظهرت الشبكات العصبية في مجال التشفير كواحدة من الادوات الفعالة في كل من تحليل الشفرة والتشفير وذلك لقدرتها الرائعة لمسح فضاء الحل لمشكلة معينة لايجاد الحل. واحدة من اخر الملاحظات التي تم توثيقها لتصرف الشبكات العصبية هي قدرتها على مزامنة نفسها مع شبكات عصبية اخرى بالاعتماد على قوانين التعلم المتبادل بغض النظر عن المدخلات وطبيعتها، هذه الظاهرة حازت على الاهتمام البالغ من المتخصصين في مجال التشفير وذلك بسبب قابلية استخدام هذه الظاهرة في نقل وتبادل المفاتيح بصورة سرية وامينة.

هذا البحث يقدم توجه جديد لسوق علمية التزامن بالاعتماد على التعلم المتبادل المبني على شبه الفوضى حيث ان مخارج الشبكة العصبية سوف يتم استخلاصها بطريقة لاخطية من خلال ذاكرة ممئلوءة بشكل متوازن من الواحدات والاصفار وكما سيتم توضحيه في هذا البحث.

https://doi.org/10.30684/etj.30.11.11
2412-0758/University of Technology-Iraq, Baghdad, Iraq

INTRODUCTION

The ability to build a secure channel is one of the most challenging _elds of research in modern communication. Since the secure channel has many applications, in particular for mobile phone, satellite and internet-based communications, there is a need for fast, effective and secure transmission protocols.[1]

Synchronization is an interesting phenomenon, which can be observed in a lot of physical and also biological systems. It has been first discovered for weakly coupled oscillators, which develop a constant phase relation to each other. While a lot of systems show this type of synchronization, a periodic time evolution is not required. This is clearly visible in the case of chaotic systems. These can be synchronized by a common source of noise or by interaction.[1,2]

As soon as full synchronization is achieved, one observes two or more systems with identical dynamics. But sometimes only parts synchronize. And it is even possible that one finds a fixed relation between the states of the systems instead of identical dynamics. Thus these phenomena look very different, although they are all some kind of synchronization. In most situations it does not matter, if the interaction is unidirectional or bidirectional. So there is usually no difference between components, which influence each other actively and those which are passively influenced by the dynamics of other systems.[1]

Synchronization of neural networks is a special case of an online learning situation. Two neural networks can be trained on their mutual output synchronized to an indentical time dependant random weight vector. In each time step they receive a common input vector, calculate their outputs, and communicate them to each other. If they agree on the mapping between the current input and the output, their weights are updated according to a suitable learning rule.In the case of discrete weight values this process leads to full synchronization in a finite number of steps. Afterwards corresponding weights in both networks have the same value, even if they are updated by further applications of the learning rule. Thus full synchronization is an absorbing state.[1,3]

NEURAL SYNCHRONIZATION AND CRYPTOGRAPHY

Neural synchronization is a phenomenon that may be related to, a fundamental way that the brain communicates information among its many computational entities where a central issue of cognitive neuroscience is to understand how a large collection of coupled neurons combines external signals with internal memories into new coherent patterns of meaning. An external stimulus localized at some input spreads over a large assembly of coupled neurons, building up a collective state univocally corresponding to the stimulus. Thus, the synchronization of spike trains of many individual neurons is the basis of a coherent perception.[1,4]

Two identical dynamical systems, starting from different initial conditions, can be synchronized by a common external-signal which is couples to the two systems. Two networks which are trained on their mutual output can synchronize to a time dependent state of identical synaptic weights, the two system will be self-organized in term of knowledge. [1, 4, 5, 6]

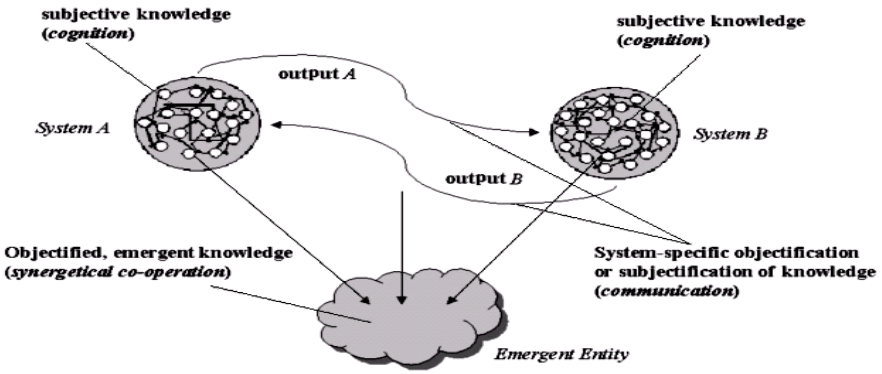
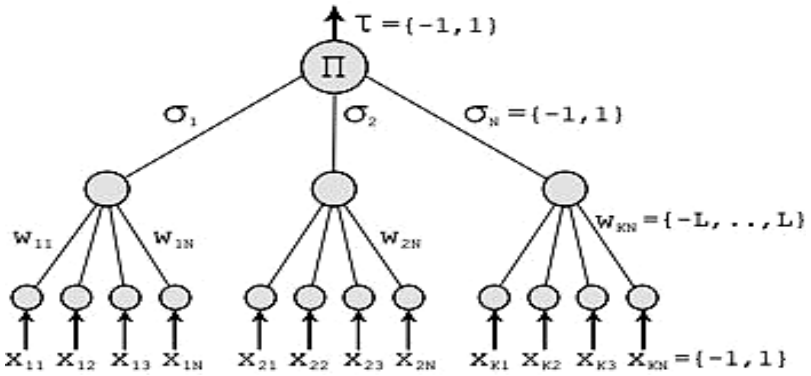


Figure (1) two systems synchronized through external-signal

A "teacher" network is presenting input/output pairs of high dimensional data, and a "student" network is being trained on these data. Training means, these synaptic weights adopt by simple rules to the input/output pairs. After the training phase the two partners will have identical weights which can be used as a secret key needed for encryption. In neural network an attacker who knows all the details of the algorithm and records any communication transmitted through this channel finds it difficult to synchronize with the parties, and hence to calculate the common secret key.[5]

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 K*N input neurons, as it is depicted in figure (2). [3,5]



Inputs to the network are binary:

Figure (2) Tree parity machine architecture

$$x_{ij} \in \{-1, +1\} \quad \text{.... (1)}$$

The weights between input and hidden neurons take the values:

$$w_{ij} \in \{-L, ..., 0, ..., +L\} \quad \text{.... (2)}$$

Output value of each hidden neuron is calculated as a sum of all multiplications of input neurons and these weights:
Signum is a simple function, which returns -1,0 or 1:

$$\sigma_i = \operatorname{sgn}\left(\sum_{j=1}^{N} w_{ij} x_{ij}\right) \quad \text{.... (3)}$$

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \quad \text{.... (4)}$$

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 is then computed as the multiplication of all values produced by hidden elements: [5]

$$\tau = \prod_{i=1}^{K} \sigma_i \quad \text{.... (5)}$$

SYNCHRONIZATION PROTOCOL THROUGH MUTUAL LEARNING

Each party (A and B) uses its own tree parity machine. Synchronization of the tree parity machines is achieved in these steps:

1. Initialize random weight values
2. Execute these steps until the full synchronization is achieved
    1. Generate random input vector X
    2. Compute the values of the hidden neurons
    3. Compute the value of the output neuron
    4. Compare the values of both tree parity machines
        1. Outputs are others: go to 2.1
        2. Outputs are same: one of the suitable learning rules is applied to the weights

After the full synchronization is achieved (the weights w_{ij} of both tree parity machines are same), A and B can use their weights as keys. This method is known as bidirectional learning. One of the following learning rules can be used for the

- Hebbian learning rule:

$$w_i^+ = w_i + \sigma_i x_i \Theta(\sigma_i \tau)\Theta(\tau^A \tau^B) \quad \text{… (6)}$$

- Anti-Hebbian learning rule:

$$w_i^+ = w_i - \sigma_i x_i \Theta(\sigma_i \tau)\Theta(\tau^A \tau^B) \quad \text{… (7)}$$

- Random walk:

$$w_i^+ = w_i + x_i \Theta(\sigma_i \tau)\Theta(\tau^A \tau^B) \quad \text{… (8)}$$

RELATED WORK

Neural synchronization can be used to construct a crypto-graphic key-exchange protocol. Here the partners benefit from mutual interaction, so that a passive attacker is usually unable to learn the generated key in time. The success probabilities of different attack methods are determined by numerical simulations and scaling laws are derived from the data. [1]

If the synaptic depth is increased, the complexity of a successful attack grows exponentially, but there is only a polynomial increase of the effort needed to generate a key. Therefore the partners can reach any desired level of security by choosing suitable parameters. In addition, the entropy of the weight distribution is used to determine the effective number of keys, which are generated in different runs of the key-exchange protocol using the same sequence of input vectors.[1]

CHAOTIC BASED NEURAL SYNCHRONIZATION

In this proposal two neural networks will be bi-directionally, synchronized over a network to present synchronized stream cipher key generator as figure (3) is presenting.

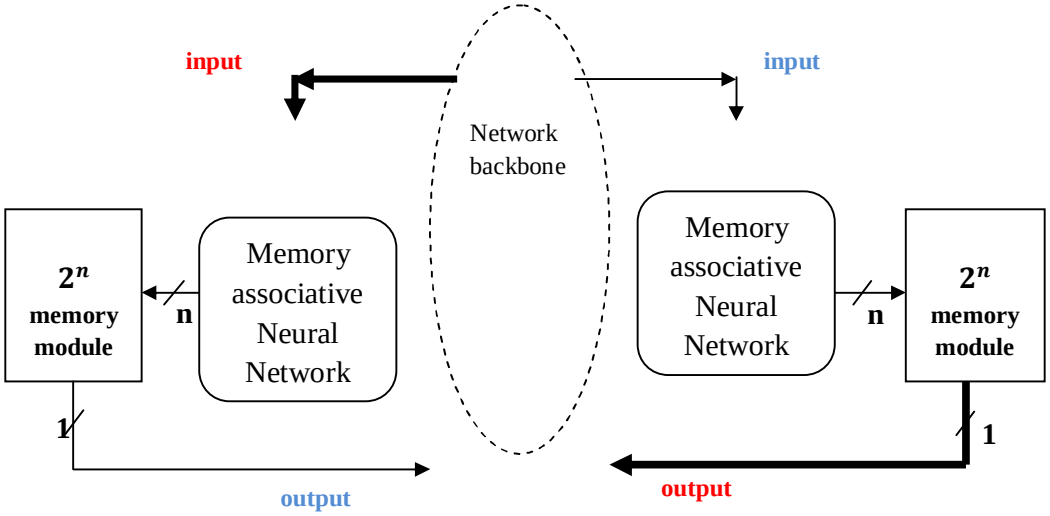


Figure (3) two neural networks synchronized to Present stream cipher generator

Neural networks can synchronize by learning from each other. For that purpose they receive common inputs and exchange their outputs. Adjusting discrete weights according to a suitable learning rule then leads to full synchronization in finite steps.

Associative memory is so-called content-addressable memory; this type of memory is not stored on any individual neuron but is a property of the whole network. It is by inputting to the network part of the memory. This is very different from conventional computer memory where a given memory (or piece of data) is assigned a unique address which is needed to recall that memory.

A Hopfield net is a form of recurrent artificial neural network invented by John Hopfield. Hopfield nets serve as content-addressable memory systems with binary threshold units.

The Hopfield network represents an auto-associative type of memory-it can retrieve a corrupted or incomplete memory, Hopfield has storage capacity which is the largest number of fundamental memories that can be stored and retrieved correctly.

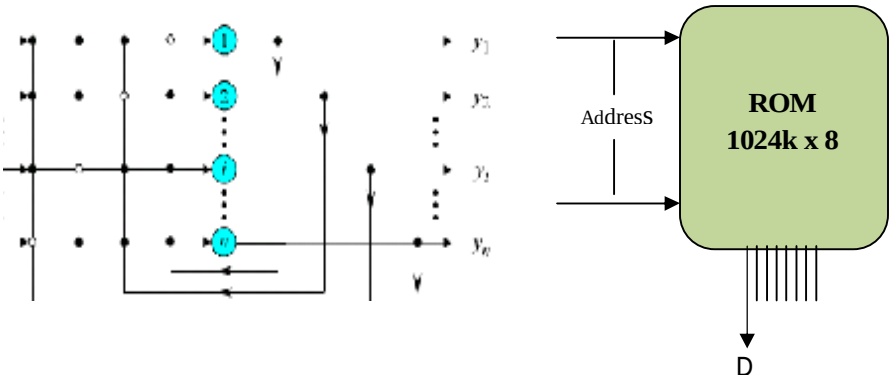


Figure (4) Hopfield neural network used as address generator

$$w_{t+1}^{A} = w_t^A + O^B . X \quad \text{..... (9)}$$

$$w_{t+1}^{B} = w_t^B + O^A . X \quad \text{..... (10)}$$

$O^B \in \{0,1\}$, **Balanced Memory**

$O^A \in \{0,1\}$, **Balanced Memory**

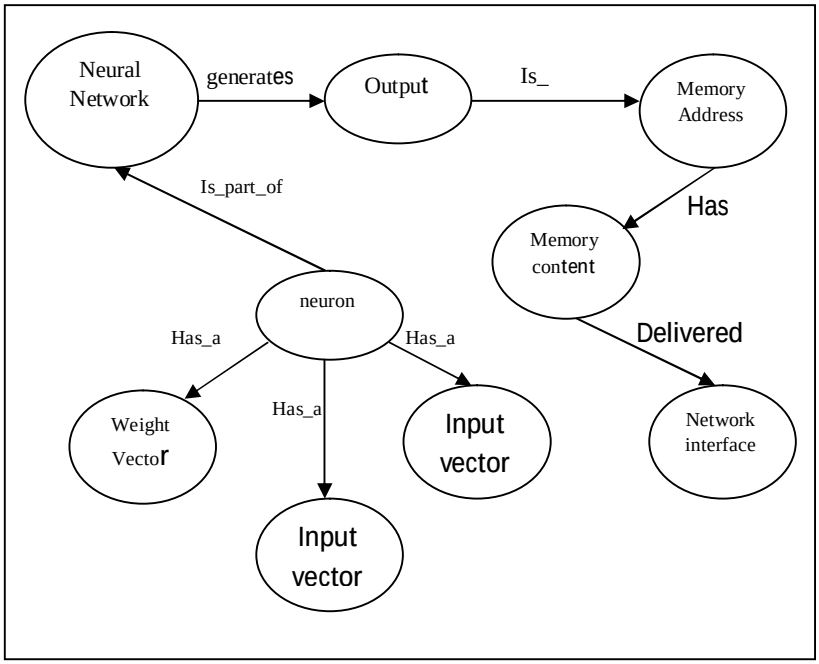


Figure (5) Conceptual Model for Chaotic Driven Neural Network Synchronization

Figure 2 presents Hopfield neural network as memory address generator where each input to Hopfield will end up with an address within the memory. After getting Transmitter's and receiver's neural network got synchronized only initial is required to fire up the output which in its return will feedback the input to generate next key.

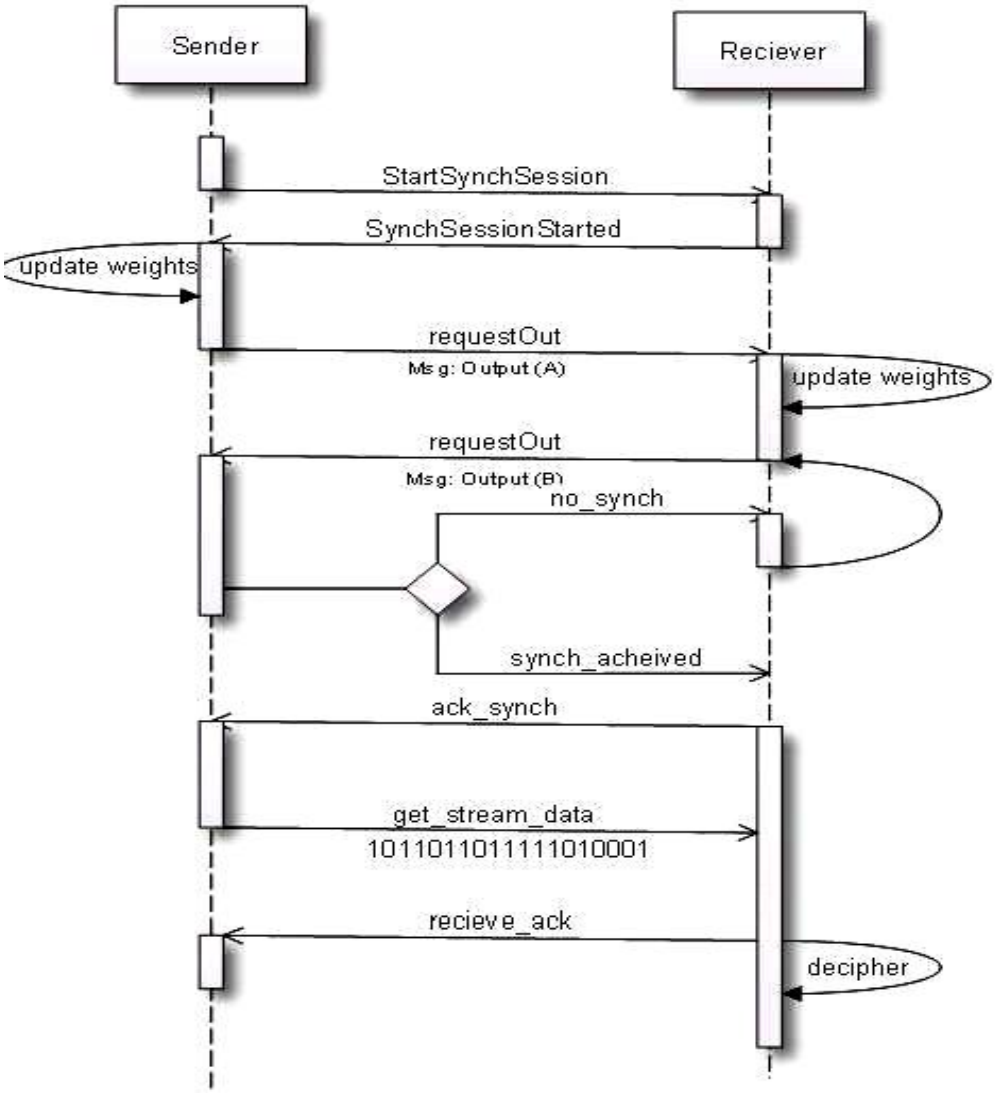


Figure (6) Hopfield Neural Networks Synchronization Sequential Diagram

Figure (7) two neural networks synchronized over 6000 rounds

CONTRIBUTION

Synchronized neural network will be used as an essential part of stream cipher key generator, where basically, Linear Feedback Shift Register (LFSR) plays that role in traditional stream cipher system. This approach can be a promising approach to replace LFSR due to

1- Can be implemented easily on hardware
2- Non-algorithmic approach to map input to output (neural feature)
3- Neural network synchronization can be used as key-exchange protocol and key generator at the same time due to the auto-associative behavior of the Hopfield.

REFERENCES

[1]Kanter, I. and W.Kinzel, " The Theory of Neural Networks and Cryptography", Minerva Center, Bar-Ilan University, Israel, 2003.

[2]Andreas Ruttor,"Neural Synchronization and Cryptography", Maximilians University, Germany, 2007

[3] Argyris, A. D. Syvridis, L. Larger, V. Annovazzi-Lodi, P. Colet, I. Fischer, J. Garc´ıa-Ojalvo, C. R. Mirasso, L. Pesquera, and K. A. Shore." Chaos-based communications at high bit rates using commercial fibre-optic links", 2005.

[4]Arecchi, F. Tito, " Chaotic neuron dynamics, synchronization, and feature binding: Quantum aspects", Australian National University, 2003.

[5] Jogdand, R. M. and Sahana S.Bisalapur, "Design of An Efficient Neural Key Generation", Gogte Institute of Technology, 2008.

[6]Christian Fuchs, "Knowledge Management In Self-Organizing Social Systems", Journal of Knowledge Management practice, 2004.