Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Secure Multiparty Key Generation Protocol using Neural Cryptography

Vandana
C S Department
EPCET, VTU
Banglore, India,

G.K.Patra
H P C & Cyber Security
CSIR Fourth Paradigm Institute
Banglore,India

Heena Kousar
C S Department
EPCET, VTU
Banglore,India

Abstract— Multiparty key generation protocol has many applications on networked systems such as secure video conferencing, group problem solving, instant messaging etc. A group of users can communicate securely over an open network by using a multiparty key generation protocol by generating a common secret key. This key is used to encrypt a sensitive message to be transmitted over an insecure channel .We have used the concept of neural cryptography to generate a common secret key. In this research work ,each communicator will run their own Tree Parity Machine, which is an special type of neural network ,and synchronize their weight .When weights of Tree Parity Machines gets synchronized it will be used as common secret key.

Keywords— Neural Cryptography, Tree Parity Machine, Synchronization, Encryption, Neural network.

I. INTRODUCTION

When a group of users wants to communicate over the open network, it generates the need of multiparty communication protocol. It has various applications like video conferencing, group problem solving, chatting, instant messaging etc. and in current time it is of great importance. It is important that such kind of communication should me more secure from third party attack. Commonly, asymmetric keys are considered more secure as there are two separate keys for encryption and decryption. But the problem with it is the slow computational speed and it requires more processing power to encrypt and decrypt the messages. [9]

On the other hand if we use the symmetric keys, it will be much faster computationally than asymmetric keys, as encryption process is less complicated. But distribution of symmetric key is a major security issue. As there are mainly two ways for key distribution either it should be passed personally (which will be cost effective) or it passes through the insecure communication channel where it will always be at the risk of being attacked by the intruder.

So, we need some mechanisms by which we can distribute the keys without passing through the insecure channels and we can avail the benefit of faster symmetric key encryption. Therefore we have used the concept of neural cryptography, which will not restrict itself for passing the keys through insecure channel instead it will be able to generate the symmetric keys at the premises of communicator.

To perform this communication each party will use a special kind of artificial neural network called Tree Parity machine. There will be one common mediator node through which output and initial input of Tree Parity machine can communicate. All the communicating parties will get their initial input from this mediator node and communicate the output of their Tree Parity Machine to each other through this node only .On the basis of which each TPM will train their neural network to get the same weights, by using suitable learning rules .When the weights will get synchronize it will be used as symmetric keys.

For increasing the strength of key we can increase numbers of input neurons, synaptic length of neural network, and number of hidden neurons. We have tested up to 100 input neurons, synaptic length L=6, Number of hidden neurons H=6, which are giving the required computational performance. We have performed the testing of this protocol by varying the different parameters, giving different computational performance. Different attack had been performed to test the security level of this multiparty communication protocol mainly focusing on geometric attacks. [6][7]

II. IMPLEMENTATION

The implementation of this multiparty key agreement protocol requires the knowledge of neural cryptography concept. Each communicator will run their own Tree Parity Machine, which is a special kind of multi-layered feed forward neural network. It consist of one output neuron, K hidden neurons and K*N input neurons .Input to the network will be binary i.e. either -1 or +1. [1]

Weight of input and hidden neuron takes the value: $W_{ij} \in \{ -L,\ldots,0,\ldots,+L\}$

Output of each hidden neuron is given as a sum of all products of input neurons and its weights.

$$\sigma i = Sgn(\sum_{j=1}^{N} \text{Wij Xij})$$

Signum is a function, which return -1, 0, +1

Sgn (x) = {-1 if x<0 , 0 if x=0, 1 if x>0 }

$$\tau = \prod_{i=1}^{K} \sigma i$$

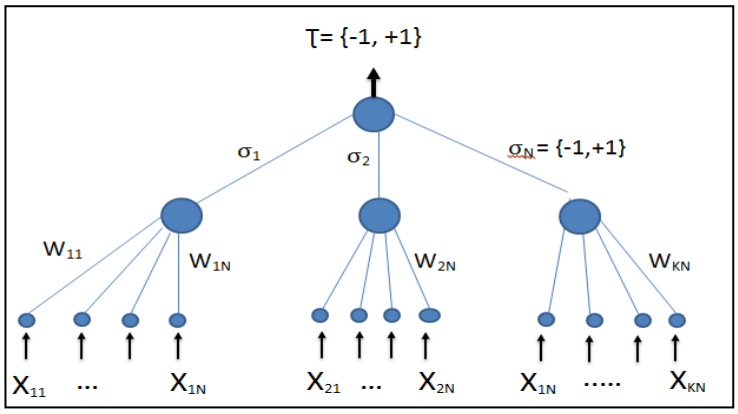


Fig1. Tree Parity Machine Structure

A. Synchronization Algorithm

Synchronization of all the communicators Tree Parity machine is achieved by following these steps. [1]

1. The mediator node will generate the random weight values for all the communicators.
2. Execute these steps until full synchronization is achieved
    1. The mediator node will generate the random input neuron value which will be same for all the communicators.
    2. Calculate the value of hidden neurons at each communicator side.
    3. Compute the value of output neuron at their respective sides.
    4. Mediator node will compare the output value of all the Tree Parity Machines.
        (a) If there outputs are not same : go to step 2
        (b) If there outputs are same, then apply learning rule to synchronize their weights

B. Different learning rules

Hebbian learning rule:

$W_{ij} = g\{W_i + \sigma i\ xi\ \theta\ (\sigma i\ \tau)\ \theta\ (\tau_A\ \tau_B)\}$

Anti-Hebbian learning rule:

$W_{ij} = g\{W_i - \sigma i\ xi\ \theta\ (\sigma i\ \tau)\ \theta\ (\tau_A\ \tau_B)\}$

Random walk :

$W_{ij} = g\{wi + xi\ \theta\ (\sigma i\ \tau)\ \theta\ (\tau_A\ \tau_B)\}$

Where, Θ=1,if input is positive, θ=0, if input is negative. The function of g is to keep the values of weights in the range of {-L, +L}.After synchronization of all the Tree Parity Machines their weights will be equal ,which will be used as symmetric keys.

III. RESULTS

This multiparty communication had tested on three nodes, which want to communicate their data, via a Mediator Node. All of three nodes are getting their initial weight values for their Tree Parity Machine from Mediator node. Mediator node is also generating common input value for neurons. All of the three communicator calculate the output of their TPM and pass it to Mediator node, which will compare the output of all TPM. I f output are others then again mediator node will reinitialize the input neurons values and follow the same procedure .If output of all three parties are same then apply suitable learning rule to synchronize their weights. Once the weights gets synchronize it can be used as symmetric keys.

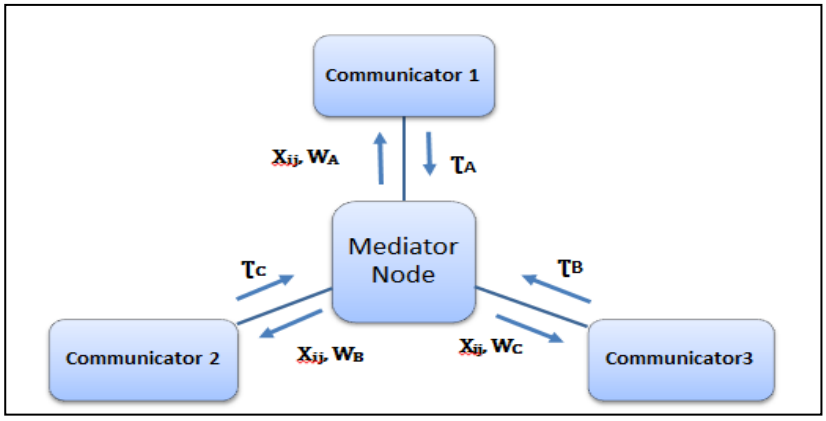


Fig2.Multiparty communication via Mediator Node

The result produced are shown in graphs:

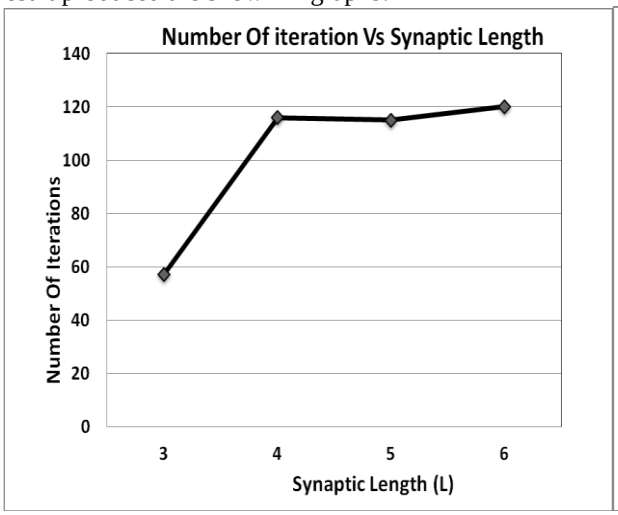


Fig 3.No of iterations, when N=100 and K=3

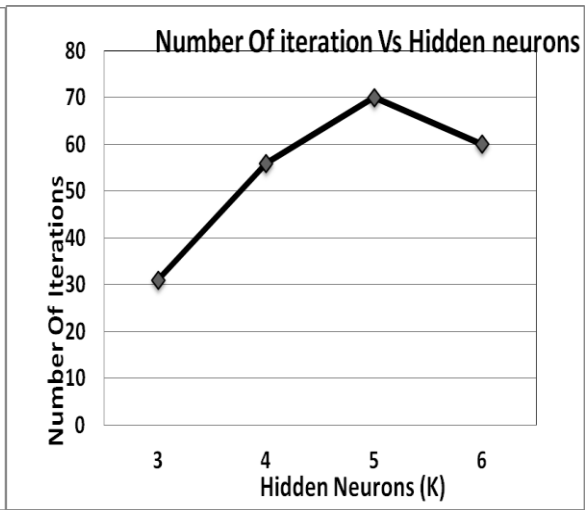
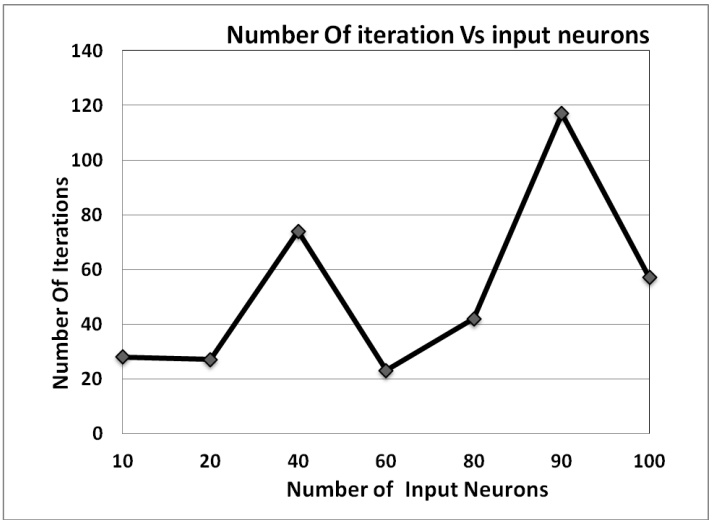


Fig 4.Number of iterations when N=100 and L=3



IV. ATTACKS

For a secure key exchange protocol, any attacker who knows all of the details of the protocol and all of the information exchanged should not have the computational power to calculate the secret key. We assume that the attacker E knows the algorithm, the sequence of input vectors and the sequence of output bits. The attacker can start with the $(2L+1)^{3N}$ initial weight vectors. E uses the same algorithm as one of the communicator. If ƮA=ƮB=ƮC the weight vector of E are flipped for which σEi is identical to ƮA.[6] The result of performing geometric attack is shown in the graphs

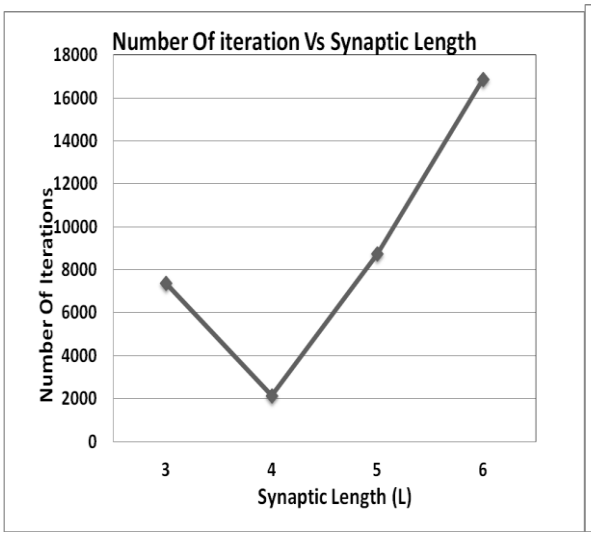


Fig 6.No of iterations when N=100 and K=3

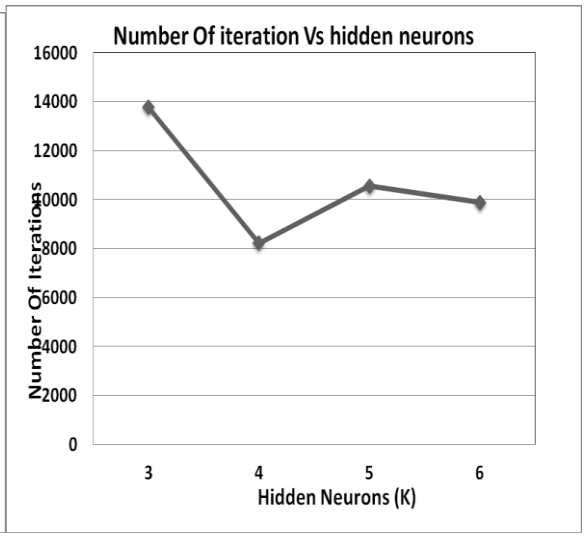


Fig 7. No of iterations when N=100 and L=3

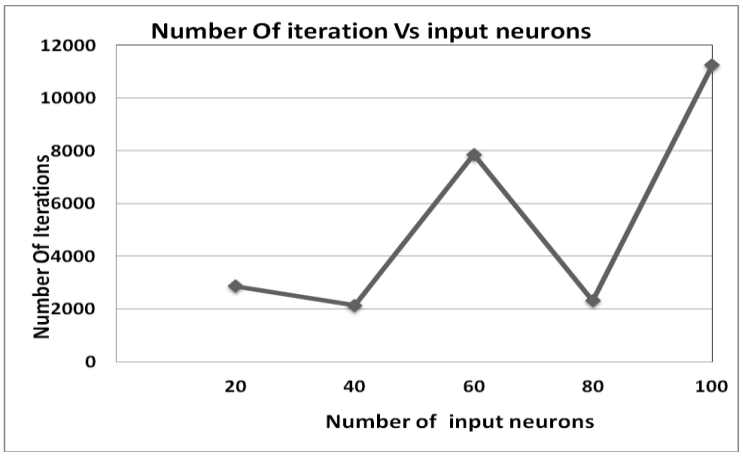


Figure-8: Number of iteration Vs. No of input Neuron (N), K=3, L=3

V. CONCLUSION

All the communicator are able to generate the secrete key over insecure channel, by training their Tree Parity Machines. They start with random initial weights and learn from each other by seeing their outputs .Attacker based on the information which is public- initial weight, input and output of Tree Parity Machine try to synchronize their Tree Parity Machine with other communicator but it is taking a huge number of iteration, or either expires before synchronization.
It is assumed that attacker knows the synchronization algorithm .Further we have to show that whether this multiparty communication remains secure for more advanced attacks. To our knowledge, neural cryptography is the first algorithm used for key generation which is not based on number theory. Therefore it is simple and fast and for each communication a new key can be used as no information is stored permanently. Therefore this multiparty key generation protocol may lead to novel application in the future.

VI. ACKNOWLEDGMENT

Vandana is thankful to the SPARK program of CSIR Fourth Paradigm Institute, Bangalore for allowing her to carry out their project at the Institute. This work is partially supported by the project ARiEES, funded by CSIR, India, under the 12th Five year plan.

REFERENCES

[1] Sagun Man Singh Shrestha ,Dept. of electronics and computer science –Khatmandu Engg. College ,Nepal ,C++ implementation of Neural cryptography for public key exchange and secure message encryption with Rijandael cipher.
[2] Neural cryptography , en.wikipedia.org/wiki/.
[3] A. Forouzan., "Cryptography and Network Security ", First Edition. McGraw-Hill, (2007), USA.
[4] William Stalling, "Network Security Essentials (Applications and Standards)", Pearson Education, 2004.
[5] Zahir Tezcan ,Computer Engineering ,Bilkent University,Public Key exchange by Neural Network.
[6] Dr.Ajit Singh,Aarti Nandal CSE ,SES ,BPSMV,India,Neural Cryptography for secret key exchange and encryption with AES.
[7] Wolfgang Kinzel ,Ido Kanter ,Proceedings of the 9th International conference on neural Information processings ,vol3
[8] Nam-su jho, Myung-Hwan Kim ,Do Won Hong and Byung-Gil Lee, "Multiparty key agreement using Bilinear map"
[9] Chu-Hsing Lin ; Tunghai Univ., Taichung ; Hsiu-Hsia Lin ; Jen-Chieh Chang ,Multiparty Key agreement for secure teleconferencing ,published in , 2006. SMC '06. IEEE International Conference on systems ,Man and Cybemetics.

