SECURED WIRELESS COMMUNICATION THROUGH SIMULATED ANNEALING GUIDED TRAINGULARIZED ENCRYPTION BY MULTILAYER PERCEPTRON GENERATED SESSION KEY (SATMLP)

Arindam Sarkar and J. K. Mandal

Department of Computer Science & Engineering,
University of Kalyani, Kalyani-741235,
Nadia, West Bengal, India.
{arindam.vb, jkm.cse}@gmail.com

ABSTRACT

In this paper, simulated annealing guided traingularized encryption using multilayer perceptron generated session key (SATMLP) has been proposed for secured wireless communication. Both sender and receiver station uses identical multilayer perceptron and depending on the final output of the both side multilayer perceptron, weights vector of hidden layer get tuned in both ends. After this tunning step both perceptrons generates identical weight vectors which is consider as an one time session key. In the 1st level of encryption process traingularized sub key1 is use to encrypt the plain text. In 2nd level of encryption simulated annealing method helps to generates sub key 2 for further encryption of 1st level generated traingularized encrypted text. Finally multilayer perceptron generated one time session key is used to perform 3rd level of encryption of 2nd level generated encrypted text. Recipient will use same identical multilayer perceptron guided session key for performing deciphering process for getting the simulated annealing guided intermediate cipher text. Then using sub key 2 deciphering technique is performed to get traingularized encrypted text. Finally sub key 1 is used to generate the plain text. In this proposed technique session key is not transmitted over public channel apart from few data transfers are needed for weight simulation process. Because after synchronization process both multilayer perceptron generates identical weight vector which acts as a session key. Parametric tests are done and results are compared in terms of Chi-Square test, response time in transmission with some existing classical techniques, which shows comparable results for the proposed system.

KEYWORDS

multilayer perceptron; simulated annealing; traingularized; session key; wireless communication.

Jan Zizka (Eds) : CCSIT, SIPP, AISC, PDCTA - 2013
pp. 217–224, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3624

1. INTRODUCTION

These days a range of techniques are available to preserve data and information from eavesdroppers [1]. Each algorithm has its own advantages and disadvantages. Security of the encrypted text exclusively depends on the key used for encryption. In cryptography the main security intimidation is man-in-the-middle attack at the time of exchange the secret session key over public channel.

In this paper, a secret session key generation mechanism has been projected using multilayer perceptron synchronization scheme i.e. session key generation using activated hidden layer's weight vector of sender & receiver's multilayer perceptron synchronization process.

The sturdiness of the key is calculated in terms of linear complexity, randomness and correlation immunity. To devise a key following 4 basic features like large period of key, large linear complexity of the key, good random key sequence, high order of correlation immunity of the key sequence is required. In this paper, additional sub key generation mechanism has been also proposed using local random search algorithm i.e. Simulated Annealing (SA) for encryption of the 1st level traingularized encrypted text. This SA based key generation methods generates key which satisfies all 4 basic features.

The organization of this paper is as follows. Section 2 of the paper deals with the problem domain and methodology. Proposed SATMLP i.e. multilayer perceptron synchronization system for generation of session key has been discussed in section 3. Section 4 deals with proposed SA based sub key generation. Traingularized encryption and decryption technique described in section 5 and 6 respectively. Experimental results of proposed technique are given in section 7. Section 8 provides the security related analysis. Conclusions and future scope are drawn in section 9 and that of references at end.

2. PROBLEM DOMAIN AND METHODOLOGY

In cryptography, the main problem is to distribution of key between sender and receiver. Because at the time of exchange of key over public channel intruders can intercept the key by residing in between them. This particular problem is discussed in following sub section and proposed technique addressed this problem.

A. Man-In-The-Middle Attack

Intruders may exist between sender & receiver and tries to capture every single information transmitting from both parties. Diffie-Hellman key swap over technique [1] suffers from this problem. Intruders can perform as sender and receiver concurrently and try to whip secret session key at the time of exchanging key via public channel. This renowned problem is being addressed in SATMLP where secret session key is not exchanged over public insecure channel. At end of neural weight synchronization strategy of both parties' generates identical output from the hidden layer when generates one time secret session key.

B. Methodology in SATMLP

This well known problem of middle man attack has been addressed in SATMLP where secret session key is not exchanged over public insecure channel. At end of multilayer perceptron

weight synchronization strategy of both parties' generates identical weight vectors and activated hidden layer outputs for both the parties become identical. This identical output of hidden layer for both parties can be use as one time secret session key for secured data exchange.

3. MULTILAYER PERCEPTRON BASED SESSION KEY GENERATION

A multilayer perceptron synaptic weight based key generation is carried out between receiver and sender [2, 3, 4, 5, 6, 7, 8]. Sender and receivers multilayer perceptron select same single hidden layer among multiple hidden layers for a particular session. For that session all other hidden layers goes in deactivated mode means hidden (processing) units of other layers do nothing with the incoming input. Synchronized identical weight vector of sender and receivers' input layer activated hidden layer and output layer becomes session key. The key generation technique and analysis of the technique is discussed in the subsections A and B. Sender and receiver multilayer perceptron in each session acts as a single layer network with dynamically chosen one activated hidden layer and K no. of hidden neurons, N no. of input neurons having binary input vector, $x_{ij} \in \{-1,+1\}$, discrete weights, are generated from input to output, are lies between -L and +L, $w_{ij} \in \{-L,-L+1,\ldots,+L\}$. Where i = 1,…,K denotes the ith hidden unit of the perceptron and j = 1,…,N the elements of the vector and one output neuron. Output of the hidden units is calculated by the weighted sum over the current input values. So, the state of the each hidden neurons is expressed using (eq.1)

$$h_i = \frac{1}{\sqrt{N}} w_i x_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} w_{i,j} x_{i,j} \qquad (1)$$

Output of the ith hidden unit is defined as $\sigma_i = \text{sgn}(h_i)$ (2)

But in case of $h_i = 0$ then $\sigma_i = -1$ to produce a binary output. Hence a, $\sigma_i = +1$, if the weighted sum over its inputs is positive, or else it is inactive, $\sigma_i = -1$. The total output of a perceptron is the product of the hidden units expressed in (eq. 2) $\tau = \prod_{i=1}^{K} \sigma_i$ (3)

Multilayer Perceptron Synchronization Algorithm

Input: - Random weights, input vectors for both multilayer perceptrons.

Output: - Secret key through synchronization of input and output neurons as vectors.

Method:-

Step 1. *Initialization of random weight values of synaptic links between input layer and randomly selected activated hidden layer. Where,* $w_{ij} \in \{-L,-L+1,\ldots,+L\}$ (4)

Step 2. *Repeat step 3 to 6 until the full synchronization is achieved, using Hebbian-learning rules.* $w_{i,j}^{+} = g\left(w_{i,j} + x_{i,j}\tau\Theta(\sigma_i\tau)\Theta\left(\tau^A\tau^B\right)\right)$ (5)

Step 3. *Generate random input vector X. Inputs are generated by a third party or one of the communicating parties.*

Step 4. *Compute the values of the activated hidden neurons of activated hidden layer using (eq. 6)*

$$h_i = \frac{1}{\sqrt{N}} w_i x_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} w_{i,j} x_{i,j} \quad (6)$$

Step 5. *Compute the value of the output neuron using* $\tau = \prod_{i=1}^{K} \sigma_i$ (7)

Compare the output values of both multilayer perceptron by exchanging the system outputs. if Output (A) ≠ Output (B), Go to step 3 else if Output (A) = Output (B) then one of the suitable learning rule is applied only the hidden units are trained which have an output bit identical to the common output. Update the weights only if the final output values of the perceptron are equivalent. When synchronization is finally achieved, the synaptic weights are identical for both the system.

Hidden Layer as a Secret Session Key

At end of full weight synchronization process, weight vectors between input layer and activated hidden layer of both multilayer perceptron systems become identical. Identical weight vector derived from synaptic link between input and activated hidden layer of both multilayer perceptron can also becomes secret session key for a particular session after full weight synchronization is achieved.

4. PROPOSED SA BASED SUB KEY GENERATION

Simulated Annealing (SA) algorithm which is based on the analogy between the annealing of solids and the problem of solving combinatorial optimization problems SA provides local search technique that helps to escape from local optima. In this scheme SA is used to generate key after satisfying some desired features such as good statistical properties, long period, large linear complexity, and highly order degree of correlation immunity.

Initialization Procedure: At the preliminary state of SA, each sequence is represented as a binary string of an equal number of 0's and 1's of a given length. So, each generated keystream (solution) coded as a binary string.

Fitness Calculation: For each sequence fitness value is calculated examining the keystream. Using the following sequence of steps fitness is calculated:

Frequency test of 0's & 1's: Proportion of 0's & 1's in the total sequence are being checked using eq. (8). $Frequency\_Fault = \left| \psi_0 - \psi_1 \right|$ (8)

Ψ_0*: No. of 0's in total sequence.* Ψ_1*: No. of 1's in total sequence.*

Binary Derivative Test: Binary Dervative test is applied to the sequence by taking the overlapping 2 tupels in the original bit stream. Here, an equal proportion of 1's and 0's in the new bit stream is checked by eq. (9)

$$Binary\_Derivative\_Fault = \left| \left(C\_\psi_0 - C\_\psi_1 \right) - 1 \right| \quad (9)$$

Table 1. Binary xor Operation Table

First Bit	Second Bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

Where, C_Ψ_0: Count no. of 0's in new bit stream. C_Ψ_1: Count no. of 1's in new bit stream.

Change Point Test: In this test a check point is created for observing maximum difference between, the proportions of 1's including the check point and the proportions of 1's after the check point using eq. (10) , eq. (11) & eq. (12).

$$D[C_p] = \Psi * K[C_p] - C_{p^*} K[\Psi] \quad (10)$$

$$\rho_r = \exp\left(-2M^2 / \psi * K[\psi] * (\psi - K[\psi])\right) \quad (11)$$

$$Change\_Point\_Fault = \rho_r \quad (12)$$

Ψ: Total no. of bit in the stream. K[Ψ]: Total no. of 1's in the bit stream.

C_p: Change Point. K[C_p]: Total no. of 1's to bit C_p (Change Point). D[C_p]: Difference respect to the Change point. M: MAX (ABS (D[C_p])), for $C_{p=1..}$ Ψ.

ρ_r : Probability of statistics that smaller value of ρ_r more significant the result.

Finally, fault of every test is summed up for calculating fitness function. Using eq. (13) fitness is calculated.

$$Fitness\_Function = \frac{1}{1 + Fault} \quad (13)$$

Cooling Procedure

To determining the cooling schedule in case of optimization problem in annealing process requires some parameters for initial value of control parameter, decrement function of control parameter, length of individual distance parameter, stopping criteria.

5. TRAINGULARIZATION ENCRYPTION TECHNIQUE

During plain text encryption, in the first phase consider a block $S = s^0_0\ s^0_1\ s^0_2\ s^0_3\ s^0_4\ s^0_5 \ldots s^0_{n-2}\ s^0_{n-1}$ of size n bits, where $s^0_i = 0$ or 1 for $0 <= i <= (n-1)$. Now, starting from MSB (s^0_0) and the next-to-MSB (s^0_1), bits are pair-wise XORed, so that the 1st intermediate sub-stream $S^1 = s^1_0\ s^1_1\ s^1_2\ s^1_3\ s^1_4\ s^1_5 \ldots s^1_{n-2}$ is generated consisting of (n-1) bits, where $s^1_j = s^0_j \oplus s^0_{j+1}$ for $0 <= j <= n-2$, $\oplus$ stands for the exclusive OR operation. This 1st intermediate sub-stream S^1 is also then pair-wise XORed to generate $S^2 = s^2_0\ s^2_1\ s^2_2\ s^2_3\ s^2_4\ s^2_5 \ldots s^2_{n-3}$, which is the 2nd intermediate sub-stream of length (n-2). This process continues (n-1) times to ultimately generate $S^{n-1} = s^{n-1}_0$, which is a single bit only. Thus the size of the 1st intermediate sub-stream is one bit less than the source sub-stream; the size of each of the intermediate sub-streams starting from the 2nd one is one bit less

than that of the sub- stream wherefrom it was generated; and finally the size of the final sub-stream in the process is one bit less than the final intermediate sub-stream. Table 2 & figure 1(a) show the process.

Table 2. Options for choosing Target Block from Triangle

Option No.	Target Block	Method of Formation
001	$s^0_0\ s^1_0\ s^2_0\ s^3_0\ s^4_0\ s^5_0\ \ldots\ s^{n-2}_0\ s^{n-1}_0$	Taking all the MSBs starting from the source block till the last block generated
010	$s^{n-1}_0\ s^{n-2}_0\ s^{n-3}_0\ s^{n-4}_0\ s^{n-5}_0 \ldots\ s^1_0\ s^0_0$	Taking all the MSBs starting from the last block generated till the source block
011	$s^0_{n-1}\ s^1_{n-2}\ s^2_{n-3}\ s^3_{n-4}\ s^4_{n-5}\ \ldots\ s^{n-2}_1\ s^{n-1}_0$	Taking all the LSBs starting from the source block till the last block generated
100	$s^{n-1}_0\ s^{n-2}_1\ s^{n-3}_2\ s^{n-4}_3\ s^{n-5}_4\ \ldots\ s^1_{n-2}\ s^0_{n-1}$	Taking all the LSBs starting from the last block generated till the source block

Table 2 describes different options for choosing target block from triangle.

6. TRAINGULARIZATION DECRYPTION TECHNIQUE

To ease the explanation of decryption technique, let us consider, $e^0_{i-1} = s^{i-1}_{n-i}$ for $1 <= i <= n$, so that the encrypted block becomes $E = e^0_0\ e^0_1\ e^0_2\ e^0_3\ e^0_4 \ldots e^0_{n-2}\ e^0_{n-1}$. After the formation of the triangle, for the purpose of decryption, the block $e^{n-1}_0\ e^{n-2}_0\ e^{n-3}_0\ e^{n-4}_0\ e^{n-5}_0 \ldots e^1_0\ e^0_0$, i.e., the block constructed by taking all the MSBs of the blocks starting from the finally generated single-bit block E^{n-1} to E, are to be taken together and it is to be considered as the decrypted block. Figure 1(b) show the triangle generated and hence the decrypted block obtained. Here the intermediate blocks are referred to as $E^1, E^2, \ldots, E^{n-2}$ and the final block generated as E^{n-1}.

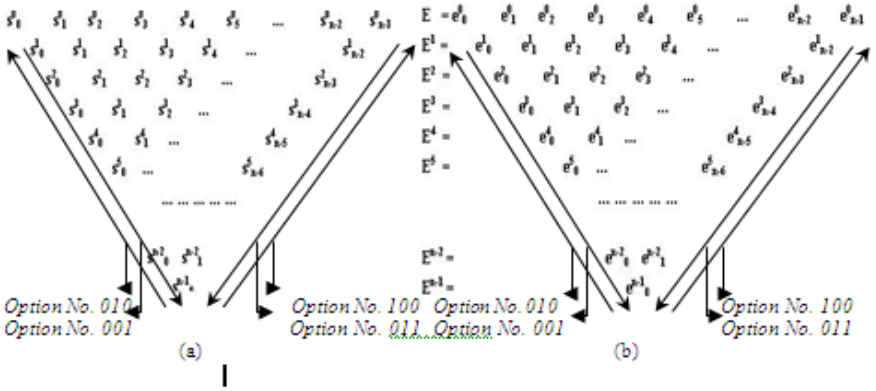

Figure 1. (a) Generation of Target Block from Source . (b) Generation of Source Block from Target

7. EXPERIMENTAL RESULTS

In this segment the results of implementation of the proposed technique has been presented in terms of Chi-Square test, no. of data exchanged & iterations needed for neural hidden layer synchronization with different parameter's value, synchronization snapshot & synchronized weight. The results are also compared with existing RSA & TDES [1] technique. Table 3 shows Chi-Square value for different source stream size after applying different encryption algorithms. It is seen that the Chi-Square value of proposed is better compared to the algorithm TDES and comparable to the Chi-Square value of the RSA algorithm. Figure 2. shows graphical representation of stream size vs. Chi square value. Figure 3. Shows simulation graph for formation of asymmetrical to symmetric synchronized weight matrix for N=10, K=10,L=4. Table 4 shows total no. of iteration needed & no. of data being transferred for synaptic simulation process with different input(N) & hidden(H) neurons and varying synaptic depth(L).

Table 3. Source size vs. Chi-Square value

Stream Size (bytes)	Chi-Square value (TDES)	Chi-Square value (Proposed SATMLP)	Chi-Square value (RSA)
1500	1228.5803	2627.7534	5623.14
2500	2948.2285	5719.8522	22638.99
3000	3679.0432	6739.73621	12800.355
3250	4228.2119	7009.2813	15097.77
3500	4242.9165	11624.2315	15284.728

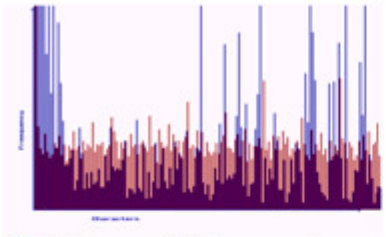

Fig 1 Size vs. Chi-Square value

Table 4. Shows Data Exchanged & No. of Iterations For Different Parameters Value

No. of Input Neurons(N)	No. of Hidden Neurons(K)	Synaptic Weight (L)	Total No. of Iterations	Data Exchanged (Kb)
5	15	3	624	48
30	4	4	848	102
25	5	3	241	30
20	10	3	1390	276
8	15	4	2390	289

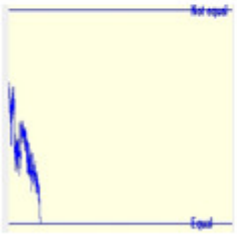

Fig 2 Shows simulation's difference chart for N=10, k=10, L=4

8. SECURITY OF THE SATMLP TECHNIQUE

The main difference between the partners and the attacker in neural machine is that sender and receiver are able to influence each other by communicating their output bits τ^A & τ^B while attacker can only listen to these messages. Of course, sender and receiver use their advantage to select suitable input vectors for adjusting the weights. These confirm that the security of neural cryptography is based on the bidirectional interaction of the partners. Each partener uses a seperate, but identical pseudo random no. generator. As these devices are initialized with a secret seed state shared by sender and receiver. They produce exactly the same sequence of input bits.

Where as attacker does not know this secret seed state. By increasing synaptic depth average synchronize time will be increased polynomial time. But success probability of attacker will be drop exponentially

9. FUTURE SCOPE & CONCLUSION

This paper presents a novel approach for generation of secret key proposed algorithm using neural hidden layer. This technique enhances the security features of the key exchange algorithm by increasing of the synaptic depth L of the neural machine. In this case, the two partners sender and receiver do not have to exchange a common secret key over a public channel but use their indistinguishable weights as a secret key needed for encryption or decryption. So likelihood of attack proposed technique is much lesser than the simple key exchange algorithm. Future scope of this technique is that this neural model can be used in wireless communication. Some evolutionary algorithm can be incorporated with this neural model to get well distributed weight vector.

ACKNOWLEDGMENT

The author expressed deep sense of gratitude to the Department of Science & Technology (DST) , Govt. of India, for financial assistance through INSPIRE Fellowship leading for a PhD work under which this work has been carried out, at the department of Computer Science & Engineering, University of Kalyani.

REFERENCES

[1] Atul Kahate, Cryptography and Network Security, 2003, Tata McGraw-Hill publishing Company Limited, Eighth reprint (2006).

[2] Mandal, J. K., Sarkar Arindam, “An Adaptive Genetic Key Based Neural Encryption For Online Wireless Communication (AGKNE)”, International Conference on Recent Trends In Information Systems (RETIS 2011) BY IEEE, 21-23 December 2011, Jadavpur University, Kolkata, India. ISBN 978-1-4577-0791-9, (2011)

[3] R. Mislovaty, Y. Perchenok, I. Kanter, and W. Kinzel. Secure key-exchange protocol with an absence of injective functions. Phys. Rev. E, 66:066102, (2002).

[4] A. Ruttor, W. Kinzel, R. Naeh, and I. Kanter. Genetic attack on neural cryptography. Phys. Rev. E, 73(3):036121, (2006).

[5] A. Engel and C. Van den Broeck. Statistical Mechanics of Learning. Cambridge University Press, Cambridge, (2001).

[6] T. Godhavari, N. R. Alainelu and R. Soundararajan “Cryptography Using Neural Network ” IEEE Indicon 2005 Conference, Chennai, India, 11-13 Dec. (2005).gg

[7] Wolfgang Kinzel and ldo Kanter, "Interacting neural networks and cryptography", Advances in Solid State Physics, Ed. by B. Kramer (Springer, Berlin. 2002), Vol. 42, p. 383 arXiv- cond-mat/0203011, (2002)

[8] Wolfgang Kinzel and ldo Kanter, "Neural cryptography" proceedings of the 9th international conference on Neural Information processing(ICONIP 02), (2002).