Multilayer neural network synchronized secured session key based encryption in wireless communication

Arindam Sarkar
Ramakrishna Mission Vidyamandira, Belur Math-711202, India

Article Info

Article history:

Received Nov 15, 2018
Revised Jan 30, 2019
Accepted Feb 15, 2019

Keywords:

Multilayer Neural Network
Session Key
Wireless Communication

ABSTRACT

In this paper, multilayer neural network synchronized session key based encryption has been proposed for wireless communication of data/information. Multilayer perceptron transmitting systems at both ends accept an identical input vector, generate an output bit and the network are trained based on the output bit which is used to form a protected variable length secret-key. For each session, different hidden layer of multilayer neural network is selected randomly and weights or hidden units of this selected hidden layer help to form a secret session key. The plain text is encrypted through chaining, cascaded xoring of multilayer perceptron generated session key. If size of the final block of plain text is less than the size of the key then this block is kept unaltered. Receiver will use identical multilayer perceptron generated session key for performing deciphering process for getting the plain text. Parametric tests have been 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 technique.

Corresponding Author:

Arindam Sarkar,
Department of Computer Science,
Ramakrishna Mission Vidyamandira,
Belur Math-711202, India.
Email: arindam.vb@gmail.com

1. INTRODUCTION

In recent times wide ranges of techniques are developed to protect data and information from eavesdroppers [1-6]. These algorithms have their virtue and shortcomings. For Example in DES, AES algorithms the cipher block length is nonflexible. In CSCT [7], KSOMSCT [8], technique uses two neural network one for sender and another for receiver having one hidden layer for producing synchronized weight vector for key generation. Now attacker can get an idea about sender and receiver's neural machine because for each session architecture of neural machine is static. In NGKRMSMC algorithm [9] any intermediate blocks throughout its cycle taken as the encrypted block and this number of iterations acts as secret key. Here if n number of iterations are needed for cycle formation and if intermediate block is chosen as an encrypted block after n/2th iteration then exactly same number of iterations i.e. n/2 are needed for decode the block which makes easier the attackers life. To solve these types of problems in this paper we have proposed a multilayer neural network guided encryption technique in wireless communication. The organization of this paper is as follows. Section 2 of the paper deals with the problem domain and methodology. Proposed multilayer neural network based key generation has been discussed in section 3. Triangularization encryption technique is given in section 4. Section 5 presents energy computation technique. Experimental results are described in section 6. Analysis regarding various security aspects of the technique has been presented in section 7. Conclusions and future scope are drawn in section 8 and that of references at end.

2. PROBLEM DOMAIN AND METHODOLOGY

In security based communication the main problem is 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 has been addressed and a technique has been proposed technique addressed this problem. These are presented in section 2.1 and 2.2 respectively.

2.1. Man-In-The-Middle Attack

Intruders intercepting in the middle of sender and receiver and try to capture all the information transmitting from both parties. Diffie-Hellman key exchange technique [1] suffers from this type of problems. Intruders can act as sender and receiver simultaneously and try to steal secret session key at the time of exchanging key via public channel.

2.2. Methodology in Proposed technique

This well known problem of middle man attack has been addressed in proposed technique where secret session key is not exchanged over public insecure channel. At end of neural 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 SYSTEM

A multilayer perceptron synaptic simulated weight based undisclosed key generation is carried out between recipient and sender. Figure 1 shows multilayer perceptron based synaptic simulation system. 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. Either synchronized identical weight vector of sender and receivers' input layer, activated hidden layer and output layer becomes session key or session key can be form using identical output of hidden units of activated hidden layer. The key generation technique and analysis of the technique using random number of nodes (neurons) and the corresponding algorithm is discussed in the subsections 3.1 to 3.5 in details.

3.1. Mutual Synchronization procedure

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 (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

$$\sigma_i = \operatorname{sgn}(h_i) \qquad (2)$$

But in case of h_i = 0 then σ_i = -1 to produce a binary output. Hence a, σ_i = +1, if the weighted sum over its inputs is positive, or else it is inactive, σ_i = -1. The total output of a perceptron is the product of the hidden units expressed in (2)

$$\tau = \prod_{i=1}^{K} \sigma_i \qquad (3)$$

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.

3.2. Multilayer Neural Network Learning rule

At the beginning of the synchronization process multilayer perceptron of A and B start with uncorrelated weight vectors $w_i^{A/B}$. For each time step K, public input vectors are generated randomly and the corresponding output bits $\tau^{A/B}$ are calculated. Afterwards A and B communicate their output bits to each other. If they disagree, $\tau^A \neq \tau^B$, the weights are not changed. Otherwise learning rules suitable for synchronization is applied. In the case of the Hebbian learning rule [10] both neural networks learn from each other.

$$w_{i,j}^{+} = g\left(w_{i,j} + x_{i,j}\tau\Theta(\sigma_i\tau)\Theta\left(\tau^A\tau^B\right)\right) \tag{4}$$

The learning rules used for synchronizing multilayer perceptron share a common structure.

$$w_{i,j}^{+} = g\left(w_{i,j} + f\left(\sigma_i, \tau^A, \tau^B\right)x_{i,j}\right) \tag{5}$$

with a function $f\left(\sigma_i, \tau^A, \tau^B\right)$, which can take the values -1, 0, or +1. In the case of bidirectional interaction it is given by

$$f\left(\sigma_i, \tau^A, \tau^B\right) = \Theta\left(\sigma\tau^A\right)\Theta\left(\tau^A\tau^B\right)\begin{cases} \sigma & \textit{Hebbian learning} \\ -\sigma & \textit{anti-Hebbian learning} \\ 1 & \textit{Random walk learning} \end{cases} \tag{6}$$

The common part $\Theta\left(\sigma\tau^A\right)\Theta\left(\tau^A\tau^B\right)$ of $f\left(\sigma_i, \tau^A, \tau^B\right)$ controls, when the weight vector of a hidden unit is adjusted. Because it is responsible for the occurrence of attractive and repulsive steps [6].

3.3. Weight Distribution of Multilayer Neural Network

In case of the Hebbian rule (7), A's and B's multilayer perceptron learn their own output. Therefore the direction in which the weight $w_{i,j}$ moves is determined by the product $\sigma_i x_{i,j}$. As the output σ_i is a function of all input values, $x_{i,j}$ and σ_i are correlated random variables. Thus the probabilities to observe $\sigma_i x_{i,j} = +1$ or $\sigma_i x_{i,j} = -1$ are not equal, but depend on the value of the corresponding weight $w_{i,j}$ [11, 13-16].

$$P\left(\sigma_i x_{i,j} = 1\right) = \frac{1}{2}\left[1 + erf\left(\frac{w_{i,j}}{\sqrt{NQ_i - w_{i,j}^2}}\right)\right] \tag{7}$$

According to this equation, $\sigma_i x_{i,j} = \mathrm{sgn}(w_{i,j})$ occurs more often than the opposite, $\sigma_i x_{i,j} = -\mathrm{sgn}(w_{i,j})$. Consequently, the Hebbian learning rule pushes the weights towards the boundaries at -L and +L. In order to quantify this effect the stationary probability distribution of the weights for $t \rightarrow \infty$ is calculated for the transition probabilities. This leads to [11].

$$P(w_{i,j} = w) = P_0 \prod_{m=1}^{|w|} \frac{1 + erf\left(\frac{m-1}{\sqrt{NQ_i - (m-1)^2}}\right)}{1 - erf\left(\frac{m}{\sqrt{NQ_i - m^2}}\right)} \tag{8}$$

Here the normalization constant ρ_0 is given by

$$P_0 = \left(\sum_{w=-L}^{L} \prod_{m=1}^{|w|} \frac{1 + erf\left(\frac{m-1}{\sqrt{NQ_i - (m-1)^2}}\right)}{1 - erf\left(\frac{m}{\sqrt{NQ_i - m^2}}\right)} \right)^{-1} \tag{9}$$

In the limit $N \rightarrow \infty$ the argument of the error functions vanishes, so that the weights stay uniformly distributed. In this case the initial length of the weight vectors is not changed by the process of synchronization.

$$\sqrt{Q_i(t=0)} = \sqrt{\frac{L(L+1)}{3}} \tag{10}$$

But, for finite N, the probability distribution itself depends on the order parameter Q_i Therefore its expectation value is given by the solution of the following equation:

$$Q_i = \sum_{w=-L}^{L} w^2 P(w_{i,j} = w) \tag{11}$$

3.4. Order Parameters

In order to describe the correlations between two multilayer perceptron caused by the synchronization process, one can look at the probability distribution of the weight values in each hidden unit. It is given by (2L + 1) variables.

$$P^i_{a,b} = P(w^A_{i,j} = a \wedge w^B_{i,j} = b) \tag{12}$$

which are defined as the probability to find a weight with $w^A_{i,j} = a$ in A's multilayer perceptron and $w^B_{i,j} = b$ in B's multilayer perceptron. In both cases, simulation and iterative calculation, the standard order parameters, which are also used for the analysis of online learning, can be calculated as functions of $P^i_{a,b}$ [12].

$$Q^A_i = \frac{1}{N} w^A_i w^A_i = \sum_{a=-L}^{L} \sum_{b=-L}^{L} a^2 P^i_{a,b} \tag{13}$$

$$Q^B_i = \frac{1}{N} w^B_i w^B_i = \sum_{a=-L}^{L} \sum_{b=-L}^{L} b^2 P^i_{a,b} \tag{14}$$

$$R^{AB}_i = \frac{1}{N} w^A_i w^B_i = \sum_{a=-L}^{L} \sum_{b=-L}^{L} ab P^i_{a,b} \tag{15}$$

Then the level of synchronization is given by the normalized overlap between two corresponding hidden units

$$\rho_i^{AB} = \frac{w_i^A w_i^B}{\sqrt{w_i^A w_i^A}\sqrt{w_i^B w_i^B}} = \frac{R_i^{AB}}{\sqrt{Q_i^A Q_i^B}} \qquad (16)$$

3.5. 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. Activated hidden layer's output of source multilayer perceptron is used to construct the secret session key. This session key is not get transmitted over public channel because receiver multilayer perceptron has same identical activated hidden layer's output. Compute the values of the each hidden unit by

$$\sigma_i = \operatorname{sgn}\left(\sum_{j=1}^{N} w_{ij} x_{ij}\right) \operatorname{sgn}(x) = \begin{cases} -1 \; if & x < 0, \\ 0 \;\; if & x = 0, \\ 1 \;\; if & x > 0. \end{cases} \qquad (17)$$

For example consider 8 hidden units of activated hidden layer having absolute value (1, 0, 0, 1, 0, 1, 0, 1) becomes an 8 bit block. This 10010101 become a secret session key for a particular session and cascaded xored with recursive replacement encrypted text. Now final session key based encrypted text is transmitted to the receiver end. Receiver has the identical session key i.e. the output of the hidden units of activated hidden layer of receiver. This session key used to get the recursive replacement encrypted text from the final cipher text. In the next session both the machines started tuning again to produce another session key. Identical weight vector derived from synaptic link between input and activated hidden layer of both multilayer perceptron can also become secret session key for a particular session after full weight synchronization is achieved.

4. 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. Figure 1 show the process.

Figure 1 describes different options for choosing target block from triangle. This option is generated by modulo 4 division of the value of output neuron then adding 1. Then take the binary version of the decimal no. Each block size is represented by 5 bits and 3 bits are used to denoting the option no. for that block. So, total 8 bits are used to describe a single block length and option chosen. For multiple blocks several 8bits are attached together preceded by first 8 bits ($2^8 = 256$ blocks can be formed in one session) to describe total no. of block to forms intermediate sub key. Maximum length of this sub key will be (256 blocks X 8 bits per block) 2048 bits. This sub key is padded in the front of the encrypted text. Then the multilayer perceptron generated synchronized one time session key is repeatedly xored with the intermediate traingularized cipher text by considering same key & traingularized cipher text length. This mechanism is performed until all the blocks get exhausted.

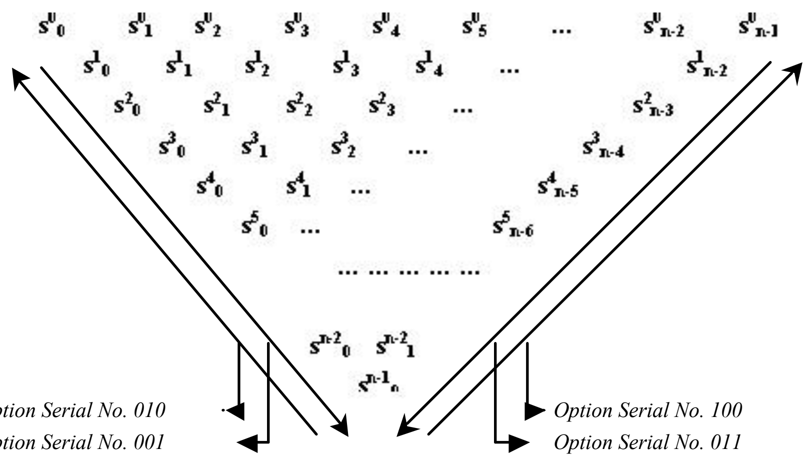

Figure 1. Options diagram for choosing Target Block from Triangle

5. ENERGY COMPUTATION

Since it is a very difficult job to find out actual power consumed in terms of "jule", approach here is to predict probable power consumption by the proposed technique. It is a difficult job for a processor to find out which process is consuming how many amount of Jules/sec, since processors run on time sharing basis. We can calculate what a single processor cycle consumes by taking a standard. All basic operations suppose addition, subtraction, multiplication requires multiple cycles. Based on that for each cycle we will calculate probable power consumed by any instruction and in the large scale we will try to find out total power consumed by the encryption/ Decryption technique. We will also try to estimate total power consumed in mutual synchronization technique of the sensor network nodes. The steps as follows:
Step1: Each atomic operation (Addition, subtraction, Division, Multiplication, assignment) is divided into cycles it require.
Step2: Assign each cycle with minimum possible power (like in terms of Micro Jule).
Step3: Then for all operation we will count number of cycles completed so far.
Step4: We will continue step 3 until all the instructions are finished execution.
Step5: Now total number of cycles is the total amount of power consumed.

Since each computer has different architecture and instruction execution totally depends of the architecture of the computer, so we have to be very conscious about the underlying machine structure. So we need to do thorough study of the architecture before drawing any conclusion.

6. EXPERIMENT RESULTS

In this paper result of the proposed technique is computed on different types of files with extensive analysis. The comparative study among proposed techniques, RSA, Triple-DES (168 bits), AES (128 bits) has been done based on twenty files by performing different types of experiment.

6.1. Statistical Analysis

For analysis of the statistical test, a large number of samples of bit sequences in the key has been considered. For m samples of bit sequences obtained from the key of a technique are tested by producing one P-value, a statistical threshold value is defined using (18).

$$Threshold\ value = (1-\alpha) - 3\sqrt{\left(\frac{\alpha\times(1-\alpha)}{m}\right)} \tag{18}$$

The objective of the test is to find proportion of zeroes and ones for the entire sequence which determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. In this experiment expected proportion for passing the test has been set to 0.972766. Table 1 shows proportion of passing and uniformity of distribution lying in the given ranges.

Table 1. Proportion of passing and uniformity of distribution for frequency

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	p-value of p-values
Proposed	0.972766	0.985437	Success	4.102711e-10
TDES		0.983333	Success	3.571386e-01
AES		0.984871	Success	3.915294e-07
RSA		0.986667	Success	4.122711e-10

From Table 1 it is seen that all proposed techniques along with existing techniques passed the frequency (monobits) test successfully because observed proportion values of all the proposed techniques are grater than expected proportion value. It is also noticed that proposed techniques outperform than existing TDES and AES technique.

6.2. Encryption/Decryption Time

Twenty files of different sizes varying from 3,216 bytes to 5,456,704 bytes have been taken to generate the data *containing* various attributes for evaluation of the proposed technique. The encryption times (Enc.) and decryption times (Dec.) of *.dll* type files obtained using proposed and existing PPM [17], TPM [18], TDES [1] and AES [1]. Figure 2 shows the graphical representation of the relationship between the encryption times against the *.dll* type source files for proposed, AES and TDES techniques. Enc. and Dec. for proposed and AES are near equal but much lower than that of PPM, TPM and TDES.

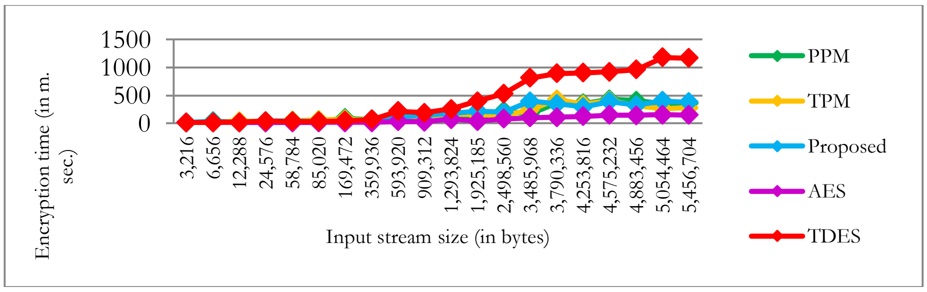

Figure 2. Encryption time against the varying size of input stream of *.dll* files

6.3. Analysis of Character Frequencies and Floating Frequencies

Analysis of character frequencies of twenty source files has been performed using proposed. Figure 3 shows the spectrum of frequency distribution of characters for the input source stream. Figure 4 shows the spectrum of frequency distribution of encrypted characters using proposed for the same input source stream. It has been observed that frequencies of characters are widely distributed in proposed encrypted file. Analysis of floating frequencies of twenty source different files has been performed using proposed. The floating frequency of a document is a characteristic of its local information content at individual points in the document. The floating frequency specifies how many different characters are to be found in any given 64-character long segment of the document. Figure 5 shows the spectrum of floating frequencies of characters for the input source stream. Figure 6 shows the spectrum of floating frequencies of encrypted characters using proposed for the same input source stream. From the figures it is observed that floating frequencies of proposed encrypted characters indicates the high degree of security.

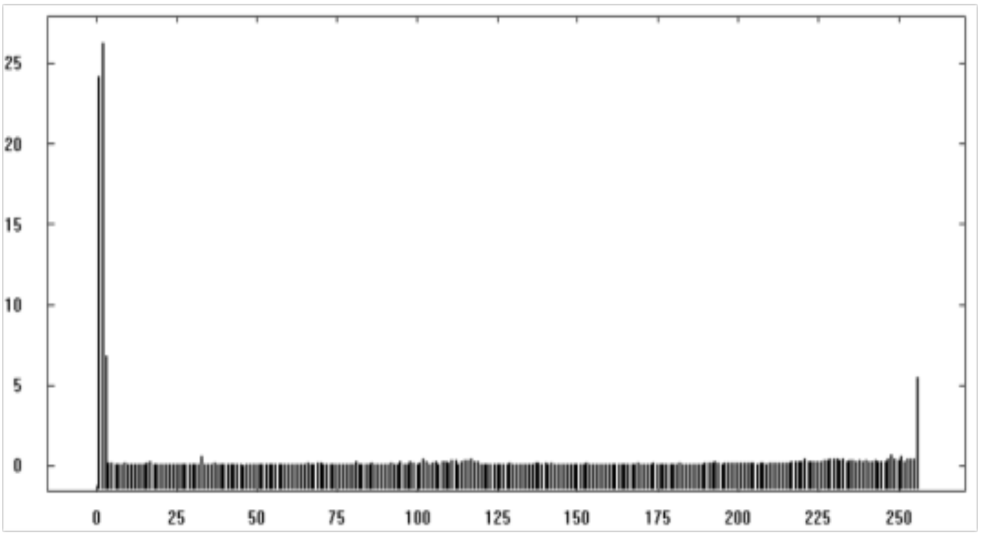

Figure 3. Spectrum of characters for the input source stream encrypted stream

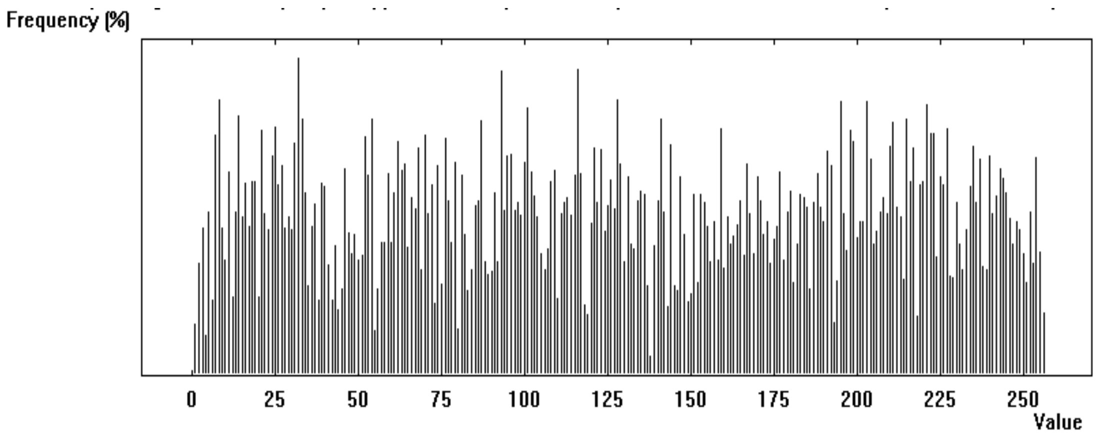

Figure 4. Spectrum of characters for the input source stream encrypted stream

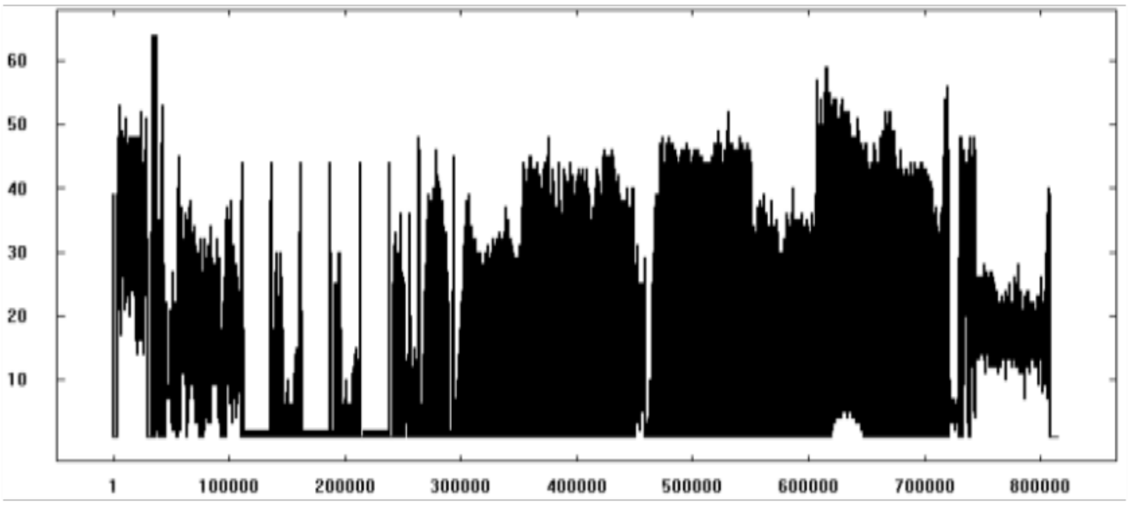

Figure 5. Floating frequency for the input source stream encrypted stream

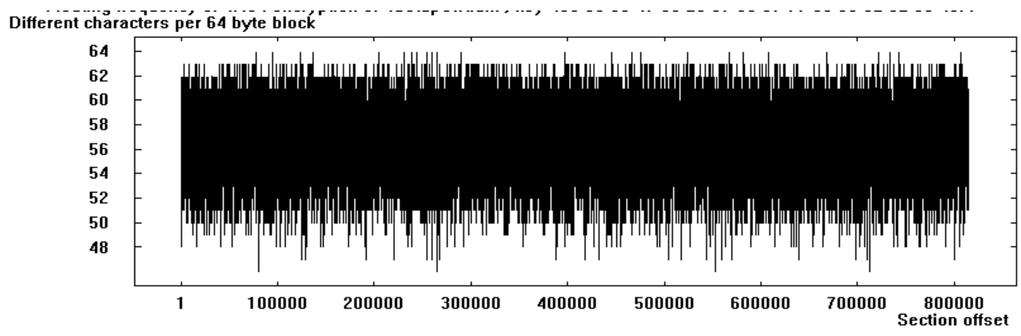

Figure 6. Floating frequency for the input source stream encrypted stream

7. SECURITY ISSUE

From results obtained it is clear that the technique will achieve optimal performances. Encryption time and decryption time varies almost linearly with respect to the block size. A user input key has to transmit over the public channel all the way to the receiver for performing the decryption procedure. So there is a likelihood of attack at the time of key exchange. To defeat this insecure secret key generation technique a neural network based secret key generation technique has been devised. The security issue of existing algorithm can be improved by using MLP secret session key generation technique. In this case, the two partners A and B do not have to share a common secret but use their indistinguishable weights or output of activated hidden layer as a secret key needed for encryption. The fundamental conception of MLP based key exchange protocol focuses mostly on two key attributes of MLP. Firstly, two nodes coupled over a public channel will synchronize even though each individual network exhibits disorganized behaviour. Secondly, an outside network, even if identical to the two communicating networks, will find it exceptionally difficult to synchronize with those parties, those parties are communicating over a public network. An attacker E who knows all the particulars of the algorithm and records through this channel finds it thorny to synchronize with the parties, and hence to calculate the common secret key. Synchronization by mutual learning (A and B) is much quicker than learning by listening (E) [10]. For usual cryptographic systems, we can improve the safety of the protocol by increasing of the key length. In the case of MLP, we improved it by increasing the synaptic depth L of the neural networks. For a brute force attack using K hidden neurons, K*N input neurons and boundary of weights L, gives (2L+1) KN possibilities. For example, the configuration K = 3, L = 3 and N = 100 gives us 3*10253 key possibilities, making the attack unfeasible with today's computer power. E could start from all of the (2L+1)3N initial weight vectors and calculate the ones which are consistent with the input/output sequence. It has been shown, that all of these initial states move towards the same final weight vector, the key is unique. This is not true for simple perceptron the most unbeaten cryptanalysis has two supplementary ingredients first; a group of attacker is used. Second, E makes extra training steps when A and B are quiet [10-12]. So increasing synaptic depth L of the MLP we can make our MLP safe.

8. FUTURE SCOPE & CONCLUSION

This paper presented a novel approach for generation of secret key proposed algorithm using MLP simulation. This technique enhances the security features of the key exchange algorithm by increasing of the synaptic depth L of the MLP. Here two partners A and B do not have to exchange a common secret key over a public channel but use their indistinguishable weights or outputs of the activated hidden layer 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 MLP model can be used in wireless communication. Some evolutionary algorithm can be incorporated with this MLP model to get well distributed weight vector.

REFERENCES

[1] Atul Kahate, Cryptography and Network Security, 2003, Tata McGraw-Hill publishing Company Limited, Eighth reprint 2006.
[2] Younsung Choi. Cryptanalysis on Privacy-Aware Two-Factor Authentication Protocol for Wireless Sensor Networks, *Indonesian Journal of Electrical Engineering and Computer Science* Vol. 8, No. 2, November 2017, pp. 296 - 301 DOI: 10.11591/ijeecs.v8.i2.pp296-301.

[3] Y Choi et al. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors. 2014; 14(6): 10081-10106.

[4] Q. Jiang et al. A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks. *International Journal of Network Management*. 2017; 27(3).

[5] S. Kumari et al. User authentication schemes for wireless sensor networks: A review. Ad Hoc Networks. 2015; 27: 159-194.

[6] S. A. Chaudhry et al. An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Networking and Applications. 2017; 10(1): 1-15.

[7] Sarkar, A., & Mandal, J. K. (2014). Intelligent Soft Computing based Cryptographic Technique using Chaos Synchronization for Wireless Communication (CSCT). *International Journal of Ambient Systems and Applications (IJASA), 2*(3), 11-20, DOI: 10.5121/ijasa.2014.2302, ISSN 2320-9259 [Online]; 2321-6344 [Print].

[8] Sarkar, A., & Mandal, J. K. (2014). Soft Computing based Cryptographic Technique using Kohonen's Self-Organizing Map Synchronization for Wireless communication (KSOMSCT). *International Journal in Foundations of Computer Science & Technology (IJFCST), 4*(5), 85-100, DOI: 10.5121/ijfcst.2014.4508, ISSN 1839-7662.

[9] Sarkar, A., & Mandal, J. K. (2014). Neuro Genetic Key Based Recursive Modulo-2 Substitution Using Mutated Character for Online Wireless Communication (NGKRMSMC). *International Journal of Computational Science and Information Technology (IJCSITY), 1*(4), 49-59, DOI: 10.5121/ijcsity.2014.1404, ISSN 2320-7442 [Online]; 2320-8457 [Print].

[10] 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.

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

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

[13] T. Godhavari, N. R. Alainelu and R. Soundararajan "Cryptography Using Neural Network" IEEE Indicon 2005 Conference, Chennai, India, 11-13 Dec. 2005.

[14] 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.

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

[16] Dong Hu "A new service based computing security model with neural cryptography"IEEE07/2009.

[17] Lu´ıs, F., Seoane, L.F., & Ruttor, A. (2011). Successful attack on PPM-based neural cryptography. *arXiv preprint arXiv*:1111.5792,24.

[18] Dolecki, M., Kozera, R., & Lenik, K. (2013). The Evaluation of the TPM Synchronization on the Basis of their Outputs. *Journal of Achievements in Materials and Manufacturing Engineering 57*(2), pp.91-98.

BIOGRAPHY OF AUTHOR

Dr. ARINDAM SARKAR is currently serving the Deparment of Computer Science & Electronics, Ramakrishna Mission Vidyamandira, Belur Math-711202, Howrah as an Astt. Professor. He has completed his Master of Computer Application (M.C.A) degree in the year of 2008 from VISVA BHARATI, Santiniketan, WB, India and he secured University First Class First Rank. In the year of 2011, Dr. Sarkar has completed his M.Tech in Comuter Science & Enggineering degree from University of Kalyani, WB, India and also secured University First Class First Rank. Dr. Sarkar has completed his Doctor of Philosophy in Engineering in the year of 2015 from University of Kalyani under the INSPIRE Fellowship Scheme of Department of Science & Technology (DST), New Delhi, India. In the year of 2016 he has secured 2nd Rank in the West Bengal College Service Commission examination. He has more than 65 International Journal and Conference publications.