Indonesian Journal of Electrical Engineering and Computer Science
Vol. 14, No. 1, April 2019, pp. 169~177
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v14.i1.pp169-177

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

Arindam Sarkar[1], Joydeep Dey[2], Anirban Bhowmik[3]
[1]Ramakrishna Mission Vidyamandira, India
[2]M.U.C. Women's College, B.C. Road, India
[3]Cyber Research & Training Institute, Goodshed Road, India

Article Info

Article history:

Received Jun 23, 2018
Revised Sep 30, 2018
Accepted Dec 7, 2018

Keywords:

Multilayer neural network
Session key
Wireless communication

ABSTRACT

Energy computation concept of multilayer neural network synchronized on derived transmission key based encryption system has been proposed for wireless transactions. Multilayer perceptron transmitting machines accepted same input array, which in turn generate a resultant bit and the networks were trained accordingly to form a protected variable length secret-key. For each session, different hidden layer of multilayer neural network is selected randomly and weights of hidden units of this selected hidden layer help to form a secret session key. A novel approach to generate a transmission key has been explained in this proposed methodology. The last thirty two bits of the session key were taken into consideration to construct the transmission key. Inverse operations were carried out by the destination perceptron to decipher the data. Floating frequency analysis of the proposed encrypted stream of bits has yielded better degree of security results. Energy computation of the processed nodes inside multi layered networks can be done using this proposed frame of work.

Corresponding Author:

Arindam Sarkar,
Ramakrishna Mission Vidyamandira,
Belur Math-711202, WB, India.
Email: arindam.vb@gmail.com

1. INTRODUCTION

A wide range of techniques are developed to provide security wrapping on the data from eavesdroppers [1]-[6]. Such algorithms have their pros and cons. For example - in DES and AES algorithm, the cipher block length is non-flexible. In CSCT [7], KSOMSCT [8], technique uses two neural networks - one for sender and another for receiver, each with one hidden layer for producing synchronized weight vector for key generation. Now attacker can guess an idea regarding sender and receiver's neural machine topology 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 $n/2^{th}$ iteration then exactly same number of iterations i.e. n/2 are needed for decode the block which makes attackers' task easier. To solve these types of problems in this paper we have proposed a multilayer neural network guided encryption technique based on transmission key in wireless communication with an emphasis on its energy computation efficiency.

1.1. Mutual Perceptron Synchronization Technique

Source and destination multilayer perceptron in different 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, $w_{ij} \in \{-L,-L+1,...,+L\}$, 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).

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

Output of the ith hidden unit is defined as

$$\sigma_i = \mathrm{sgn}(h_i) \quad (2)$$

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 (2).

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

Compare the output values of both multilayer perceptron by exchanging the system outputs. If Output (Source) ≠ Output (Destination), then restart. Otherwise, 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 completely achieved, the synaptic weights are identical for both the system.

1.2. Problem Domain

During the symmetric key based encryption, the main problematic factor is the exchange of the key between the two parties. Because at the time of exchange of key over public channel; intruders can intercept the key by residing there silently. In classical cryptography, man-in-the middle attack is vulnerable type of common attack. The key streams with low randomized characteristics would be easy to decrypt by the intruders. Once the key is compromised then the entire transmission session will be revealed.

1.3. Proposed Solution Specification

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. After the end phase of the neural weight synchronization strategy between parties, generated identical weight vectors and activated hidden layer outputs for both the parties become identical. And hence this identical output of hidden layer for both parties can be used secret session key. Furthermore, second round of encryption has been applied on the last thirty two bits of the session key to generate the transmission key.

2. PROPOSED MULTILAYER PERCEPTRON BASED SESSION KEY GENERATION SYSTEM

A multilayer perceptron synaptic simulated weight based undisclosed key generation is carried out between recipient and sender. 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 later subsection.

2.1. Multilayer Neural Network Learning rule

Initially, the multilayer perceptron of source and destination 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, they communicate their output bits to each other. If they disagree, $\tau^{A} \neq$

τ^B, then 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.

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

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

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

Equation 5 with a function $f\left(\sigma_i, \tau^A, \tau^B\right)$, 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 the adjustment of weight vector of a hidden unit because it is responsible for the occurrence of attractive and repulsive steps as mentioned in equation 6.

2.3. Weight Distribution of Multilayer Neural Network

In case of the Hebbian rule as mentioned in equation 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 reverse equation, $\sigma_i x_{i,j} = -\mathrm{sgn}(w_{i,j})$. 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 $t \to \infty$ is calculated for the transition probabilities. This leads to [11].

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

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 $N \to \infty$ 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:

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

2.4. 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 (2*L + 1) variables.

$$P_{a,b}^{i} = P\left(w_{i,j}^{A} = a \wedge w_{i,j}^{B} = b\right) \tag{12}$$

which are defined as the probability to find a weight with $w_{i,j}^{A} = a$ in A's multilayer perceptron and $w_{i,j}^{B} = b$ 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 $P_{a,b}^{i}$ [12].

$$Q_i^A = \frac{1}{N} w_i^A w_i^A = \sum_{a=-L}^{L} \sum_{b=-L}^{L} a^2 P_{a,b}^{i} \tag{13}$$

$$Q_i^B = \frac{1}{N} w_i^B w_i^B = \sum_{a=-L}^{L} \sum_{b=-L}^{L} b^2 P_{a,b}^{i} \tag{14}$$

$$R_i^{AB} = \frac{1}{N} w_i^A w_i^B = \sum_{a=-L}^{L} \sum_{b=-L}^{L} ab P_{a,b}^{i} \tag{15}$$

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

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

2.5. Proposed 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 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 only after complete weight synchronization is achieved.

3. PROPOSED TRANSMISSION KEY BASED ENCRYPTION

After achieving complete synchronization but source and destination multilayer perceptrons, they do select the thirty two bits staring from least significant bit of the hidden layer vector. Such selected bits were partitioned into four bits pattern followed by decimal conversion with modulus operation. Now those eight numbers were stored at the header row of the double dimensional matrix. The entire matrix was filled through the entire sequence of the synchronized session key. Projection of each column was done on the basis of increasing order of the header row. Thus, concatenation of all such projected columns have resulted the transmission key. The transmission key thus generated would be masked with the original data. Reverse operations were performed by the destination perceptron.

Proposed Algorithm for Transmission Key Generation

Requirement(s): $Weight\ vector\ W[P]\ for\ perceptron\ with\ size\ P.$
Input(s): Last 32 bits of W[P].
Output(s): Transmission key Tk[P].
Method(s): Decimal conversion with modulus 8 on LSB 32 bits. Ascending projection of the columns of the matrix results Tk[P].

{/*Decimal Conversion with modulus */}

$$ARR[\ 0 \ldots 7\ ] \leftarrow Call\ Convert2DecimalMod8(4\ , W[(P-31)])$$

{/* Filling 2D matrix ARR[P/8 + 1][8]*/}
Set k1 ← 0
Set k2 ← 7
Set r ← 0
While [$r <= P/8$]
{
$CP[0 \ldots.7] \leftarrow Call\ Fill(\ W[(k1\ \ldots\ k2)\ ]\ )$
$ARR[r] \leftarrow\ CP[\ 0 \ldots 7]$
K1 ← $k1 + 8$
k2 ← $k2 + 8$
r ← $r + 1$
}

{/* Ascending Projection*/}
St[0 … 7] ← Call $ASC\_SORTING(\ ARR[0\ \ldots\ 7\ ]\ )$
Set $m1 \leftarrow 0$
Set $d \leftarrow 0$
for i= 1 to (P/8) do
$Tk[m1\ \ldots\ 7] \leftarrow Select\ ARR[St[d]]$
Set $m1 \leftarrow m1 + 8$
Set $d \leftarrow d + 1$
end for

Ready Tk as Transmision Key for Encryption

4. PROPOSED ENERGY COMPUTATION

Since it is a very difficult job to find out actual power consumed in terms of "Joule". It is a difficult job for a processor to find out which process is consuming how many amount of joules/sec, since processors run on time sharing mode. 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 are as follows.

Step1: Each atomic operation (Addition, subtraction, Division, Multiplication, assignment) is divided into cycles as per requirement.
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.

5. 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 by performing different types of experiment.

5.1. Statistical Analysis

For analysis of the statistical test, a large number of samples of bit sequences in the key have 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 equation 17.

$$Threshold\ value = (1-\alpha) - 3\sqrt{\left(\frac{\alpha\times(1-\alpha)}{m}\right)} \quad (17)$$

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 greater than expected proportion value. It is also noticed that proposed techniques outperform than existing TDES and AES technique.

5.2. Encryption/Decryption Time

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 1 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.

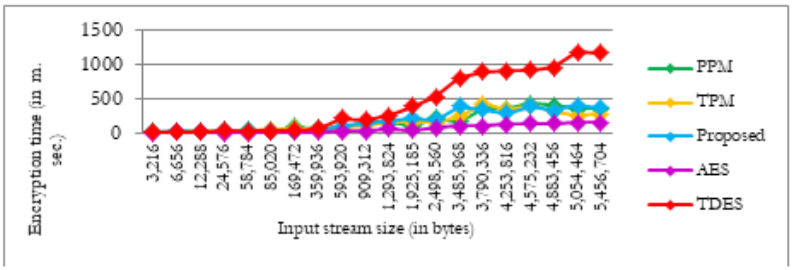

Figure 1. Encryption time against the varying size of input stream of *.dll* files

5.3. Analysis of Character Frequencies and Floating Frequencies

Analysis of character frequencies of source files has been performed using proposed. Figure 2 shows the spectrum of frequency distribution of characters for the input source stream. Figure 3 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. 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 4 shows the spectrum of floating frequencies of characters for the input source stream. Figure 5 shows the spectrum of floating frequencies of encrypted characters using proposed for the same input source stream. Comparison between the Figure 4 and the Figure 5, it has been observed that floating frequencies of proposed encrypted characters indicates the high degree of security.

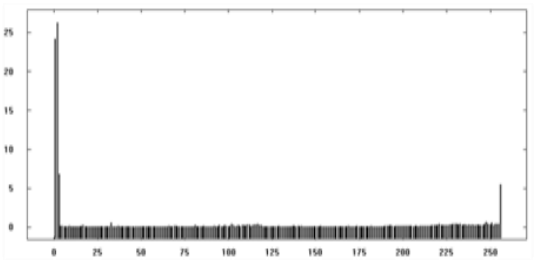

Figure 2. Spectrum of characters for the input source stream

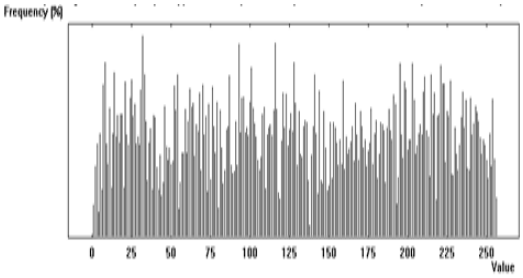

Figure 3. Spectrum of characters for the encrypted stream

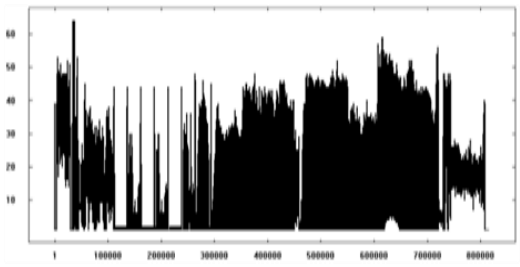

Figure 4. Floating frequency for the input source stream

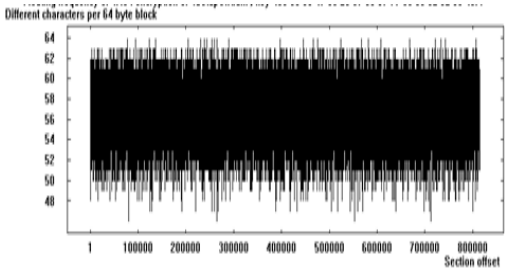

Figure 5: Floating frequency for the encrypted stream

5. SECURITY ISSUE

From results obtained in the above section, 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 followed by generation of transmission key. The security issue of existing algorithm can be improved by using multilayer perceptron 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 concept of MLP based key exchange protocol focuses mostly on two key attributes of MLP. This technique can be used in secured image encryption [19] and authentication in wireless network [20]. Firstly, two nodes coupled over a public channel will synchronize even though each individual network exhibits disorganized behavior. 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 (2*L+1)*(3*N) 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.

6. CONCLUSION AND FUTURE SCOPE

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. Energy waste management through GA tools, ACO, etc are the future scopes of this technique.

REFERENCES

[1] Kahate A. Cryptography and Network Security. 2nd edition. New Delhi: Tata McGraw Hill. 2010: 41-71.
[2] Ahmad, J.I. , Din, R. , Ahmad, M., Review on Public Key Cryptography Scheme-Based Performance Metrics, *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)* Vol. 12, No. 1, October 2018, pp. 386-392.
[3] Choi Y, et al. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors*. 2014; 14(6): 10081-10106.
[4] Jiang Q, 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] Kumari S, et al. User authentication schemes for wireless sensor networks: A review. *Ad Hoc Networks*. 2015; 27: 159-194.
[6] Chaudhry S A, 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. Intelligent Soft Computing based Cryptographic Technique using Chaos Synchronization for Wireless Communication (CSCT). *International Journal of Ambient Systems and Applications (IJASA).2014; 2*(3): 11-20.
[8] Sarkar A, Mandal J K. 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). 2014; 4*(5): 85-100.
[9] Sarkar A, Mandal J K. 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).2014; 1*(4): 49-59.
[10] Mislovaty R, Perchenok Y, Kanter I, Kinzel W. Secure Key-exchange Protocol with an absence of injective functions. Phys. Rev. E. 66:066102. 2002.
[11] Ruttor A, Kinzel W, Naeh R, Kanter I. Genetic attack on Neural Cryptography. Phys. Rev. E, 73(3):036121. 2006.

[12] Engel A, Van den Broeck C. Statistical Mechanics of Learning. Cambridge University Press, Cambridge, 2001.
[13] Godhavari T, Alainelu N R, Soundararajan R. Cryptography Using Neural Network. IEEE Indicon 2005 Conference. 2005.Chennai, India: 258-261.
[14] Wolfgang Kinzel, ldo Kanter. *Interacting Neural Networks and Cryptography*. Advances in Solid State Physics. Berlin. 2002; 42: 383.
[15] Wolfgang Kinzel, ldo Kanter. *Neural Cryptography*. Neural Information Processing(ICONIP 02) . Singapore. 2002.
[16] Dong Hu . A new service based computing security model with neural cryptography. IEEE07.2009.J
[17] Lu´ıs F, Seoane L F, Ruttor A. Successful attack on PPM-based neural cryptography.2011.
[18] Dolecki M, Kozera R, Lenik K. The Evaluation of the TPM Synchronization on the Basis of their Outputs. *Journal of Achievements in Materials and Manufacturing Engineering .2013; 57*(2);91-98.
[19] H. Ali-Pacha, N. Hadj-Said, A. Ali-Pacha, M. Mamat, and M. A. Mohamed, "An Efficient Schema of a Special Permutation Inside of Each Pixel of an Image for its Encryption," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 11, no. 2, 2018.
[20] Choi, Younsung. "Cryptanalysis on Privacy-aware two-factor Authentication Protocol for Wireless Sensor Networks". *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, Vol. 8, no. 2, pp. 296-301, 2017.