SOFT COMPUTING BASED CRYPTOGRAPHIC TECHNIQUE USING KOHONEN'S SELF-ORGANIZING MAP SYNCHRONIZATION FOR WIRELESS COMMUNICATION (KSOMSCT)

Arindam Sarkar[1] and J. K. Mandal[2]

[1]Department of Computer Science & Engineering, University of Kalyani, W.B, India

ABSTRACT

In this paper a novel soft computing based cryptographic technique based on synchronization of two Kohonen's Self-Organizing Feature Map (KSOFM) between sender and receiver has been proposed. In this proposed technique KSOFM based synchronization is performed for tuning both sender and receiver. After the completion of the tuning identical session key get generates at the both sender and receiver end with the help of synchronized KSOFM. This synchronized network can be used for transmitting message using any light weight encryption/decryption algorithm with the help of identical session key of the synchronized network. Exhaustive parametric tests are done and results are compared with some existing classical techniques, which show comparable results for the proposed system.

KEYWORDS

Kohonen's Self-Organizing Map (KSOFM), soft computing, cryptography, Wireless Communication.

1. INTRODUCTION

A range of techniques are obtainable to protect data and information from attackers [1, 5, 6, 7, 8, 9, 10]. Existing TPM and PPM [2, 3, 4] technique have some limitations like secret seed values used in the generation of identical input vector has to be transmitted to the other party via public channel in the SYN frame in each iteration. This significantly increases the synchronization time. Also for ensuring the security this parameters should not be transmitted via public channel. Furthermore, TPM and PPM needs significant amount of synchronization steps. This may not suitable in wireless communication because of resource constraints criteria. Proposed method of this paper eliminates all the above stated drawbacks of the TPM and PPM. The proposed technique performs the KSOFM based synchronization for generation of secret tuned session key at both ends with fewer amounts of synchronization steps compared to TPM and PPM. The organization of this paper is as follows. Proposed cryptographic technique has been discussed in section 2. Experiments results of this technique are given in section 3. Conclusions are drawn in section 4 and that of references at end.

2. THE TECHNIQUE

The proposed technique constructs the secret key at both sides using exchanged information. For ensuring the randomness in every session, certain parameters value get randomly change in every session like seed value for generating random inputs and weights, number of iteration to train the map, different mathematical functions (Radial basis, Gaussian, Mexican Hat) for choosing the random points from the KSOFM.

DOI:10.5121/ijfcst.2014.4508

2.1 Synchronization Methodology

In this section, a novel Kohonen Self-Organizing Feature Map (KSOFM) based synchronization of both sender and receiver machine has been presented. In this proposed technique input examples are used to build the map through vector quantization method along with unsupervised competitive learning for performing the synchronization. Proposed method uses unsupervised learning method to represent input space of the training samples in a discrete 2D maps. Neighborhood of each neuron (i.e. the connections of the neuron with adjacent neurons) in the map depends on the dimension of the map. 2D regular spacing in a hexagonal or rectangular grid uses to arrange the neurons. Following sub sections discussed about detailed methodology used in synchronization procedure.

2.1.1 Initialization of weight vector

In this proposed technique KSOFM comprises of neurons along with a weight vector for each neuron having a dimension same as the dimension of the input vectors. Consider the input vector $X = [x_1, x_2, ..., x_n]^T$ and weight vector $W = [w_1, w_2, ..., w_n]^T$ .This scheme initially, assigns a weight vector to each neuron (point) by arbitrarily choosing a neuron (point) of the input space. The value of the weights vector is set to a tiny random numbers.

2.1.2 Selection of point

For initially assigning a weight vector to each neuron (point) an arbitrary neuron (point) $\boldsymbol{x} \in$ $input\ space\ X$ get selected.

2.1.3 Unsupervised Training of KSOFM

The necessity of unsupervised learning mechanism in KSOFM is to produce similar response from different parts of the network for a certain input patterns. This proposed technique uses competitive learning in the training period to train the KSOFM. Euclidean distance between each neuron and an arbitrary neuron (point) x get calculated by:

$$Distance_k = \sqrt{\left[(x_1 - w_{k1})^2 + (x_2 - w_{k2})^2 + ... + (x_n - w_{kn})^2\right]}$$

Where,

$k = 1,2,\ldots,P$

P is the neuron number

W_{kj} is the entry of j of the weight of neuron k

The neuron (point) whose weight vector is most similar to the input is called the Best Matching Unit (BMU). The weights of the BMU and neurons (point) close to it in the KSOFM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance (within the lattice) from the BMU. This proposed technique uses 2D KSOFM with 100 neurons. A learning rate α of 0.1 is used to train the KSOFM and decreasing the spread of the neighborhood function by the rule:

$$\sigma = \sigma_0\left(1 - \frac{Iteration\ \_number}{Total\ \_no\ \_of\ \_iteration}\right)$$

Here, σ is initial spread. The value of σ decreases from the initial value to the final value (0) constantly. Hence the neighborhood function influences all neurons of the map in the first time and its influence on far neurons vanishes progressively. Near the end of the training only the winner neuron will be updated so as to drive neurons to gravity centers.

2.1.4 Selection of Winner neuron

The winner neuron is a neuron which has a minimum distance to x (arbitrary point). Winner neuron gets selected based on the distance factor. The minimum distance $Distance_{winner}$ fulfills the condition:

$$Distance_{winner} \leq Distance_k$$

$$Where\ k = 1,2,\ldots,P$$

2.1.5 Updating of KSOFM

An updating rule has been applied over the entire map keeping in mind the priority of the winner neuron and its closest neighbors. In this proposed technique updating is done using following formula

$$w_{k,new} = w_{k,old} + \alpha.Neighbor(winner,k).(x - w_{k,old})$$

Where α is the learning step and $Neighbor(winner,k)$ is a neighborhood function with a bell shape centered at the winner neuron. It is a function of the distance between the winner neuron and the neuron k.

$$Neighbor\ (winner\ ,k) = e^{-\frac{|winner\ -k|^2}{\sigma^2}}$$

2.1.6 Reiteration on KSOFM

A new arbitrary point $y \in input\ space\ X$ get selected and all the steps discussed in sub section 3.1.3 to 3.1.5 get perform again. This process is repeated for each input vector for a (usually large) number of cycles λ.

2.2 Synchronization parameters

The spreading of the neighbourhood function (σ) is important since it controls the convergence of the map. It should be large at the beginning and shrink progressively to reach a small value in order to globally order the neurons over the whole map. The maximum value of $Neighbor(winner,winner) = 1$ corresponds to the winner neuron and value of *Neighbor function* decreases when the distance between neurons k and *winner* increases. Concerning the value of the learning rate α , it should be small enough to ensure the convergence of the KSOFM. In this proposed synchronization technique the sender and receiver both uses the identical KSOFM architecture along with identical parameters in each session. Following are the list of parameters used in each session.

- Dimension of the KSOFM (2D or 3D)
- Number of neurons which specifies the number of different possible session keys
- Dimension of the weight vector specify the length of the key
- Seed value for generating random inputs and weights
- Number of iteration to train the map

- Different mathematical functions as a mask for choosing the random points from the KSOFM (Radial basis, Mexican Hat, Gaussian etc.)
- Different index value for choosing different neurons (key) on the mathematical mask at each session for forming the session key

Parameters that get negotiated at the initial stage of synchronization procedure between sender and receiver by mutual agreement are completely random. By changing each of the parameters randomly in each session security of proposed technique can be enhanced which in turns decrease the success rate of the attackers.

In this proposed technique both A's and B's KSOFM are starts synchronization by exchanging some control frames for negotiation of parameters value.

Synchronization (SYN) Frame:

$Command\ Code$ 00	SYN_ID	$DIM\ Index$	$Weight\ DIM\ Index$	$Iteration\ Index$	$Mask\ Index$	$Seed\ Index$	$Neuron\ DIM\ Index$	$CRC\ (Cyclic\ Redundancy\ Checker)$
2	4	1	2	16	2	4	4	16 (bits)

Figure 3: Synchronization (SYN) Frame

- Sender constructs a SYN frame and transmitted to the receiver for handshaking purpose in connection establishment phase. SYN usually comprises of
  $Command\ code$: Following table illustrate the different command code in respect to the different frame.

Table 1 Command Code

Command code	Frame
00	SYN
01	ACK_SYN
10	NAK_SYN
11	$Index$

SYN_ID: 4 bits SYN_ID is used to identify different SYN frame in different session.
$DIM\ index$: 1 bit is used to specify the dimension of KSOFM. Following table illustrate the $DIM\ index$ corresponds to the dimension.

Table 2 Command Code

DIM Index	KSOFM Dimension
0	2D
1	3D

$Iteration\ index$: 16 bits are used to illustrate the $Iteration\ index$. Following table shows the $Weight\ DIM\ index$ corresponds to the number of weights.

$Weight\ DIM\ index$: 2 bits are used to illustrate the $Weight\ DIM\ index$.

Table 3 Command Code

Weight DIM Index	Number of weights
00	64
01	128
10	192
11	256

$Mask\ Index$: 2 bits are used to illustrate the $Mask\ Index$. Following table illustrate the different $Mask\ Index$ value corresponds to the different mathematical mask function.

Table 4 Command Code

Mask Index	Mathematical Mask Function
00	Mexican Hat
01	Gaussian
10	Radial Basis
11	Reserved

$Seed\ index$: 4 bits are used to illustrate the $Seed\ index$.
$Neuron\ DIM\ index$: 4 bits are used to illustrate the $Neuron\ DIM\ index$ which is the total number of neurons.
$CRC\ (Cyclic\ Redundancy\ Checker)$: 16 bits are used in CRC.

- When the receiver receives the frame SYN, the receiver should carry out integrity test.
- Receiver performs Integrity test after receiving the SYN frame. If the messages are received as sent (with no replication, incorporation, alteration, reordering, or replay) the receiver will execute the synchronization phase.

Acknowledgement of Synchronization (ACK_SYN) Frame:

$Command\ \ Code$ 0010	SYN_ID	CRC $(Cyclic\ Redundancy\ \ Checker)$
4	8	16 (bits)

Figure 4: Acknowledgement of Synchronization (ACK_SYN) Frame

ACK_SYN frame send by the receiver to the sender for positive acknowledgement of the parameters value.

Negative Acknowledgement of Synchronization (NAK_SYN) Frame:

$Command\ \ Code$ 0011	SYN_ID	CRC $(Cyclic\ Redundancy\ \ Checker)$
4	8	16 (bits)

Figure 5: Negative Acknowledgement of Synchronization (NAK_SYN) Frame

NAK_SYN frame send by the receiver to the sender for negative acknowledgement of the parameters value.

Finish Synchronization (FIN_SYN) Frame:

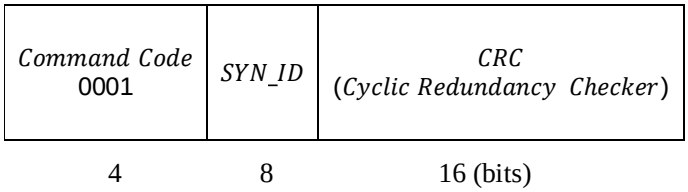


Figure 6: Finish Synchronization (FIN_SYN) Frame

Finally, the receiver should send the frame FIN_SYN to alert the sender.

2.3 Synchronization Algorithm

Input : Assign a weight vector to each neuron by arbitrarily choosing a point of the input space
Output : Synchronized KSOFM
Method : The process operates on sender's and receiver's Kohonen's Self-Organizing Feature Map (KSOFM) and generate synchronized session key.
Step 1. Randomize the map's nodes' weight vectors
Step 2. Select an arbitrary input vector
Step 3. Traverse each node in the map
Step 3.1 Use the Euclidean distance formula to find the similarity between the input vector and the map's node's weight vector
Step 3.2 Track the node that produces the smallest distance (this node is the best matching unit, BMU)
Step 4. Update the nodes in the neighborhood of the BMU (including the BMU itself) by pulling them closer to the input vector
Step 5. Increase s and repeat from step 2

After a mutually predetermined iteration steps both sender and receiver stop their iteration. Now, both sender and receiver have the identical final KSOFM because they have started with same initial configuration and proceeds with same mutually agreement parameters. In this situation the sender uses mathematical function as a mask to form the session key. The receiver would use the same mutually pre agreed mathematical function as a mask in the map for reconstruct the session key. Use of mathematical mask increases the security of the scheme because instead of one single neuron, key can be constructed using several neurons on the mask. Also changing the mask parameters several key can be generated. From this discussion it can be concluded that initially mask method get slightly more amount of time but once the mask get set it takes less amount of time. An adversary could not find the key because they do not have the map, mathematical function for masking and other mutually pre agreed parameters. Mask method associate several neurons around the designated neuron to form the key. This provides a significant improvement to the security of the generation of session key. A mask hides all neurons other than the winner. In this proposed scheme different mask functions like Gauss, Radial basis, Mexican hat functions are get used randomly in different session. The winner neuron fixes the centre of the mask and

each neuron around the winner will be weighted and summed to the winner. The result is a different key depending on the shape of the mask. The relation of the key is obtained by:

$$Key_{u,q} = \sum_{i,j} w_{ij,q} \cdot f_{winner}(i, j) \quad q = 1,2,\ldots,N$$

Where,

$f_{winner}(i, j)$ is the mask centred at the winner neuron,

$w_{ij,q}$, is the component q of the vector associated to neuron (i, j) and

$Key_{u,q}$, is the component q of the ultimate (final) key.

A general form of the mask is represented by the the following equation:

$$f_{winner}(x) = a.e^{-\frac{|k-w|^2}{\sigma_1^2}} - be^{-\frac{|k-w|^2}{\sigma_2^2}}$$

with $a, b, \sigma_1, \sigma_2 \in R$, x a neuron in the KSOFM, $winner$ is the winner neuron. A huge number of masks could be generated by changing parameters a, b, σ_1, σ_2. Using the mask incontestably enhances the security of the key. In wireless communication, instead of starting from the initial state of the KSOFM key generation procedure a user may use the same trained KSOFM in different session with different users by changing only the parameters value of the mask or mask function used to determine the ultimate key. This procedure helps to save the resources of wireless communication very efficiently.

3. SECURITY ANALYSIS

In this paper a KSOFM synchronized cryptographic technique has been proposed. The technique generates the synchronized session key by tuning KSOFM of both sender and receiverThe Following attacks are considered to ensure the robustness of the proposed technique.

Cipher text only Attack: In this type of attack, the attacker has access to a set of cipher text. In cipher text only attack, encryption algorithm and cipher text is known to an attacker. An attacker tries to break the algorithm or in simple words tries to deduce the decryption key or plaintext by observing the cipher text. This proposed technique nullifies the success rate of this attack by producing a robust KSOFM encrypted cipher text. The strength of resisting exhaustive key search attack relies on a large key space. So, cipher text produces by this proposed methodology is mathematically difficult to break. Thus a hacker has to try all such key streams to find an appropriate one. Key stream have high degrees of correlation immunity. Thus it is practically difficult to perform a brute-force search in a key-space.

Known Plaintext Attack: The attacker has access to one or more cipher text and some characters in the original data. The objective is to find the secret key. Proposed technique offers better floating frequency of characters. So, known plain text attack is very difficult in this proposed technique.

Chosen Plaintext Attack: Here, the attacker has liberty to choose a plaintext of his/her choice and get the corresponding cipher text. Since the attacker can choose plaintext of his/her choice, this attack is more powerful. Again the objective of this attack is to find the secret key. This attack is impractical because there is no obvious relationship between the individual bits of the sequence in plain text and cipher text. So, it is not possible to choose a plaintext of his/her choice and get the corresponding cipher text.

Chosen Cipher text Only Attack: The attacker can choose cipher text and get the corresponding plaintext. By selecting some cipher text a cryptanalyst has access to corresponding decrypted plaintext. Chosen cipher text only attack is more applicable to public key cryptosystems. This technique has a good Chi-Square value this confirms good degree of non-homogeneity. So, it will be difficult to regenerate plain text from the cipher text.

Brute Force Attack: A cryptanalyst tries all possible keys in finite key space one by one and check the corresponding plaintext, if meaningful. The basic objective of a brute force attack is to try all possible combinations of the secret key to recover the plaintext image and or the secret key. On an average, half of all possible keys must be tried to achieve success but brute force attack involves large computation and has a very high complexity. Due to high complexity brute force attack will not be feasible. Proposed technique has a good entropy value near to equal 8 which indicates that brute force attack is very difficult in this proposed technique.

4. RESULTS

A total of 15 statistical tests of The NIST Test Suite have been performed to evaluate randomness of the KSOFM synchronized session key proposed in this paper. These tests focus on a variety of different types of non-randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests. The 15 tests are following:

Table 5 Statistical Results

Statistical Test	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
Frequency (Monobits) Test	0.972766	0.976329	Success	2.835246e-03	Non-uniform
Test for Frequency within a Block	0.972766	0.972818	Success	3.407162e-04	Non-uniform
Runs Test	0.972766	0.975746	Success	0.321683e-01	Uniform
Longest Run of Ones in a Block	0.972766	0.97051	Unsuccess	1.491737e+02	Uniform
Binary Matrix Rank Test	0.972766	0.990000	Success	7.491904e-02	Uniform
Discrete Fourier Transform Test	0.972766	0.968329	Unsuccess	0.000000e+00	Non-uniform
Non-overlapping	0.972766	0.988891	Success	0.000000e+00	Non-uniform

(Aperiodic) Template Matching Test					
Overlapping (Periodic) Template Matching Test	0.972766	0.980201	Success	7.259823e-02	Non-uniform
Maurer's "Universal Statistical" Test	0.972766	1.000000	Success	7.830128e-01	Uniform
Linear Complexity Test	0.972766	0.975318	Success	2.945727e-01	Uniform
Serial Test	0.977814	0.973903	Unsuccess	0.000000e+00	Non-uniform
Approximate Entropy Test	0.972766	0.985830	Success	2.837463e-02	Uniform
Cumulative Sums Test	0.977814	0.971289	Unsuccess	0.000000e+00	Non-uniform
Random Excursions Test	0.983907	0.942500	Unsuccess	0.000000e+00	Non-uniform
Random Excursions Variant Test	0.985938	0.972963	Unsuccess	0.000000e+00	Non-uniform

Table:6 Comparisons of 128 bit key length vs. average synchronization time (in cycle)

Key Length (128 bits)	**Average Synchronization time (in cycle)**
KSOFM	2516,41
TPM (L=25)	2624,27
PPM	2811,04

Table shows KSOFM technique needs 2516,41 cycles in average to generate session key having a length of 128 bit. Whereas existing TPM (L=25) and PPM needs 2624,27 and 2811,04 cycles respectively, which larger than all the proposed techniques. KSOFM outperforms over existing TPM and PPM. This is quite affordable in terms of resources available in wireless communication.

Table:7 Comparisons of 128 bit key length vs. average synchronization time (in cycle) for group synchronization (Group size=4)

Key Length (128 bits)	**No. of Parties participating in the Group Synchronization**	**Average Synchronization time (in cycle)**
KSOFM	4	15098,46
TPM (L=25)	4	15745,62
PPM	4	16866,24

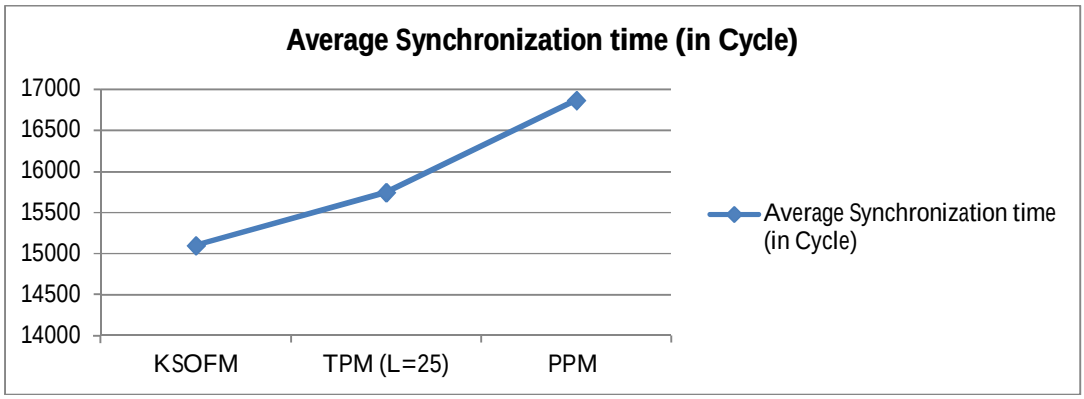


Figure 16: Comparisons of 128 bit key length vs. average synchronization time (in cycle) for group synchronization (Group size=4)

Table and figure shows KSOFM, TPM, PPM technique needs (15098,46), (15745,62), (16866,24) cycles respectively in average to generate session key having a length of 128 bit for synchronize group of 4 parties . This clearly indicates that proposed technique outperforms than all other existing techniques at the time of group synchronization.

Table:8 Comparisons of 192 bit key length vs. average synchronization time (in cycle)

Key Length (192 bits)	Average Synchronization time (in Cycle)
KSOFM	3173,41
TPM (L=25)	3347,15
PPM	3571,48

Table and figure shows KSOFM technique needs 3173,41 cycles in average to generate session key having a length of 128 bit. Whereas existing TPM (L=25) and PPM needs 3347,15 and 3571,48 cycles respectively, which larger than the proposed techniques. KSOFM outperforms over existing TPM and PPM. This is quite affordable in terms of resources available in wireless communication.

Table: 9 Comparisons of 192 bit key length vs. average synchronization time (in cycle) for group synchronization (Group size=4)

Key Length (192 bits)	No. of Parties participating in the Group Synchronization	Average Synchronization time (in Cycle)
KSOFM	4	19040,46
TPM (L=25)	4	20082,90
PPM	4	21428,88

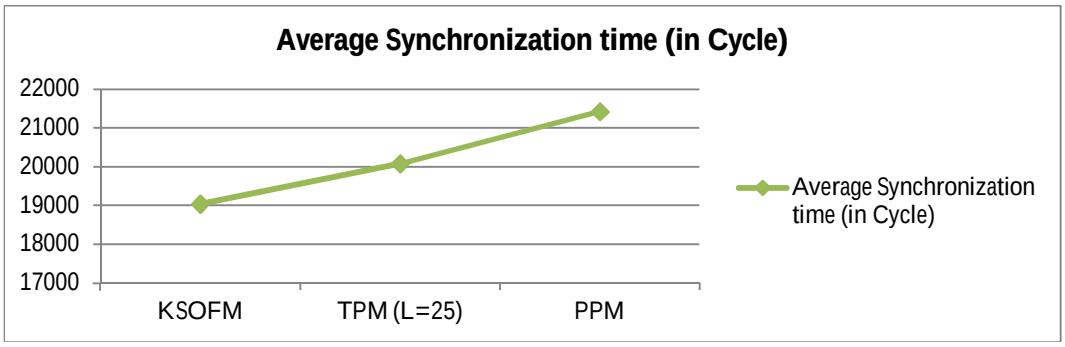


Figure 17: Comparisons of 192 bit key length vs. average synchronization time (in cycle) for group synchronization (Group size=4)

Table shows KSOFM technique needs 19040,46 cycles in average to generate session key having a length of 192 bit for synchronize group of 4 parties This clearly indicates that proposed technique outperforms than all other proposed and existing techniques at the time of group synchronization.

Table: 10 Comparisons of 256 bit key length vs. average synchronization time (in cycle)

Key Length (256 bits)	Average Synchronization time (in Cycle)
KSOFM	4719,72
TPM (L=25)	4851,86
PPM	5193,03

Table and figure shows KSOFM technique needs 4719,72 cycles in average to generate session key having a length of 128 bit. Whereas existing TPM (L=25) and PPM needs 4851,86 and 5193,03 cycles respectively, which larger than the proposed techniques. This is quite affordable in terms of resources available in wireless communication.

Table: 11 Comparisons of 256 bit key length vs. average synchronization time (in cycle) for group synchronization (Group size=4)

Key Length (256 bits)	No. of Parties participating in the Group Synchronization	Average Synchronization time (in Cycle)
KSOFM	4	28318,32
TPM (L=25)	4	29111,16
PPM	4	31158,18

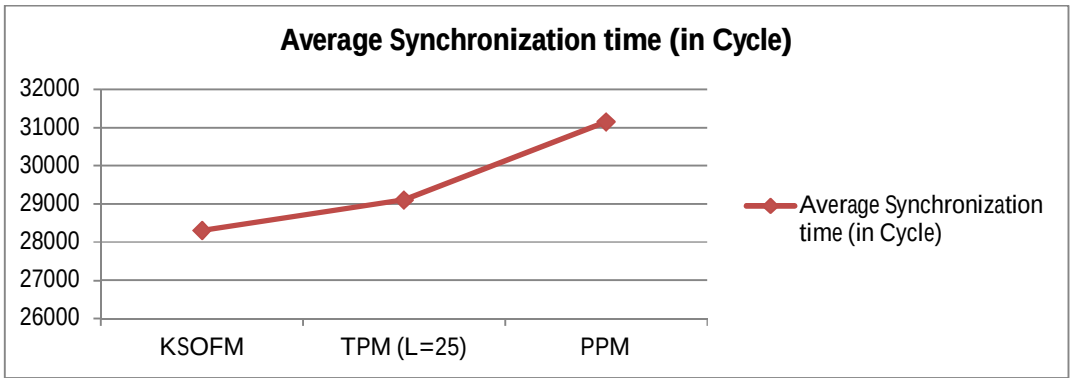


Figure 18: Comparisons of 256 bit key length vs. average synchronization time (in cycle) for group synchronization (Group size=4)

Table shows KSOFM technique needs 28318,32 cycles respectively in average to generate session key having a length of 192 bit for synchronize group of 4 parties . This clearly indicates that proposed technique outperforms than all other proposed and existing techniques at the time of group synchronization.

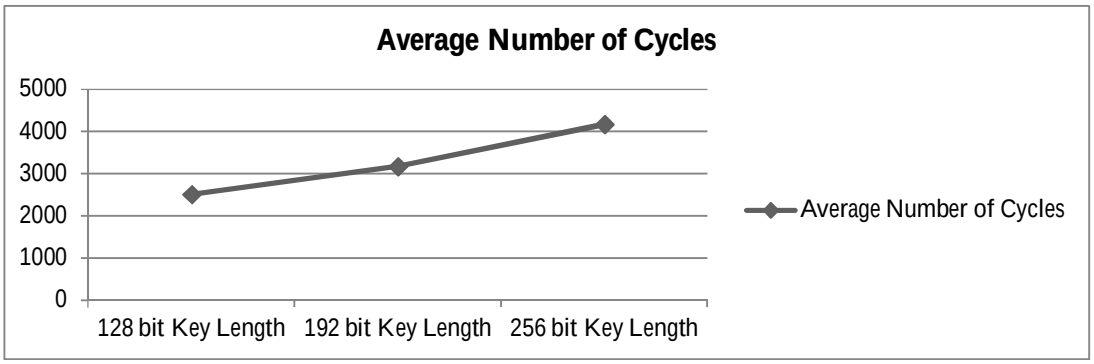


Figure 19: Key length vs. average number of iterations

From figure it has been observed that if the length of the session key get increased then the increased of average synchronization steps is linear.

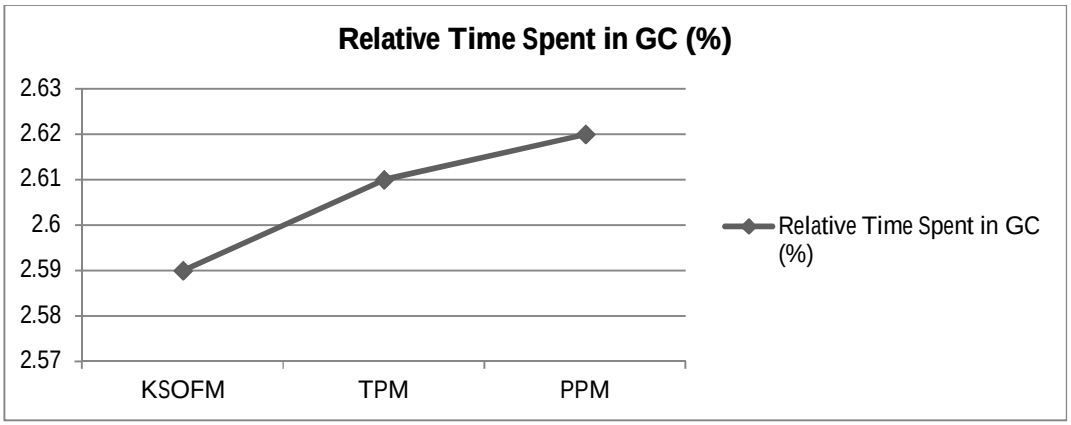


Figure 20: Comparisons of relative time spent in GC to generate 128 bit session key

From the above table and figure it has been shown that increasing order sequence of relative time spent in GC in group synchronization phase is KSOFM, TPM and PPM.

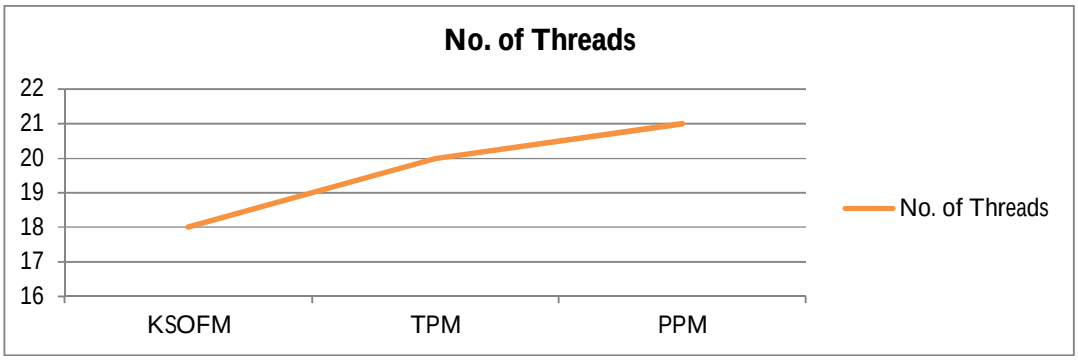


Figure 21: Comparisons of number of Threads required generating 128 bit session key

From the above table and figure it has been shown that increasing order sequence of number of thread required in group synchronization phase is KSOFM, TPM and PPM.

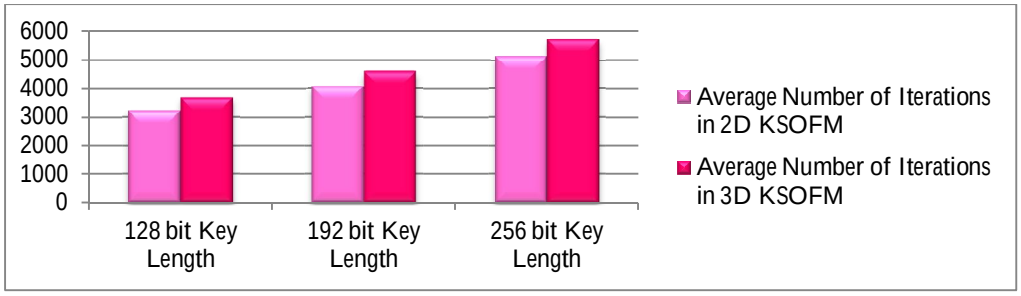
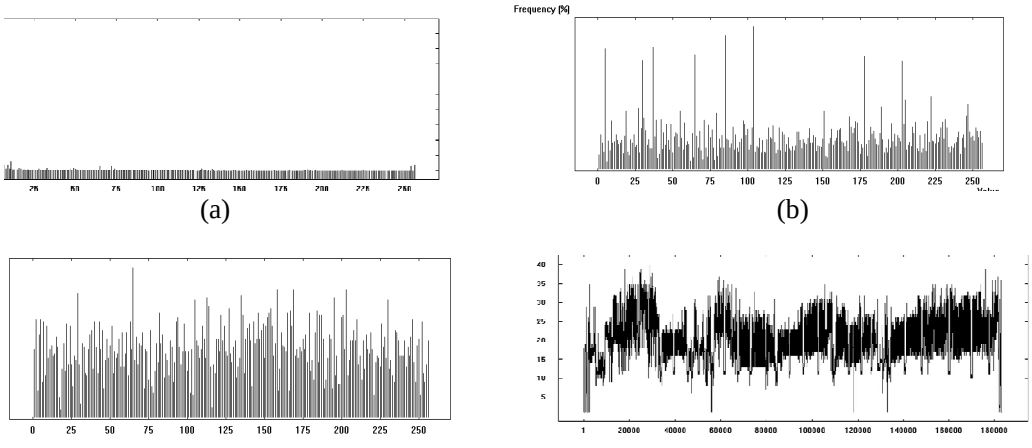


Figure shows the average number of iterations to be needed for generating 128, 192 and 256 bit session key in 2D and 3D KSOFM. The above figure depicts that 3D KSOFM takes more iterations to train the map in compared to 2D KSOFM. So, the energy consumption is more in 3D KSOFM than 2D. For this reason 2D KSOFM is the best alternative in wireless communication where resource constrains (in terms of energy, memory) is a vital issues for generation of session key.



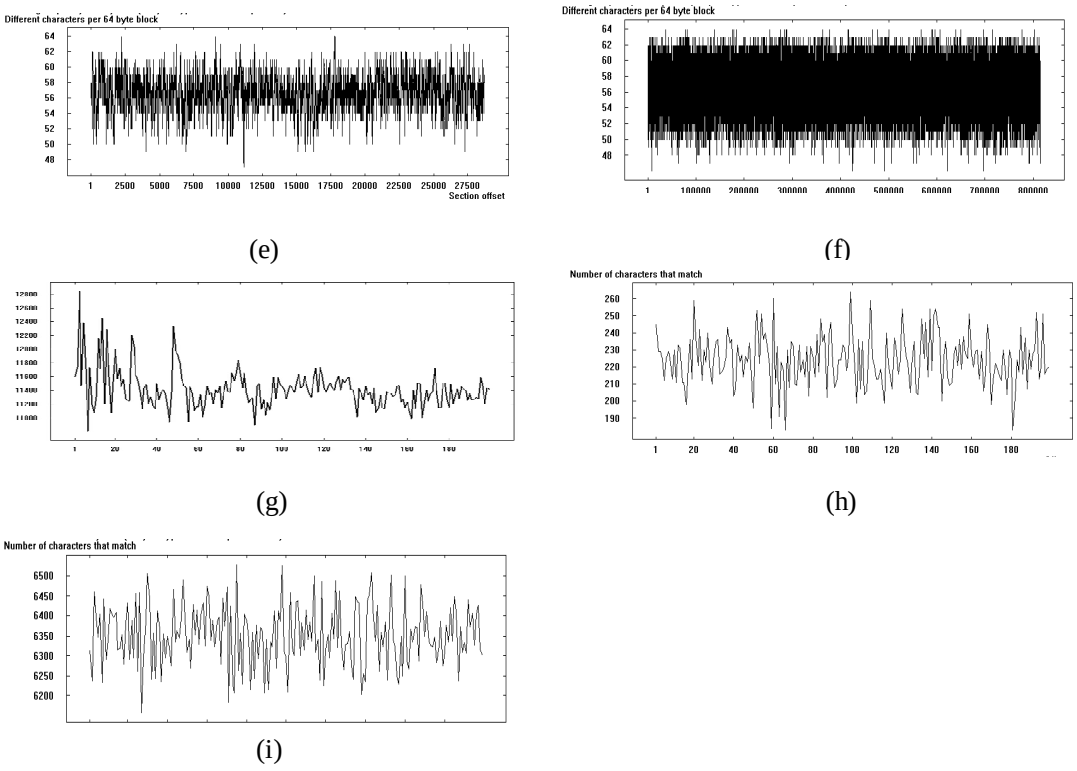


Figure 23: (a), (b), (c) shows the frequency distribution of plain text, TPM based encrypted text, proposed chaos KSOFM based encrypted text, figure: (d), (e), (f) shows the floating frequency of plain text, TPM based encrypted text, proposed chaos KSOFM based encrypted text, figure: (g), (h), (i) shows the autocorrelation of plain text, TPM based encrypted text, proposed chaos KSOFM based encrypted text

5. CONCLUSION

Proposed technique is very simple and easy to implement in various high level language. The test results also show that the performance and security provided by the proposed technique is good and comparable to standard technique. The security provided by the proposed technique is comparable with other techniques. To enhance the security of the technique, proposed technique offers changes of some parameters randomly in each session. To generate the secret session key index mask get exchanged between sender and receiver. This technique has a unique ability to construct the secret key at both sides using this exchanged information. Since the encryption and decryption times are much lower, so processing speed is very high. Proposed method takes minimum amount of resources which is greatly handle the resource constraints criteria of wireless communication. This method generates a large number of keys which is the same number of neurons in the map. For ensuring the randomness in every session, some of the parameters get change randomly at each session. proposed methods outperform than existing TPM, PPM and does not suffers from Brute Force or Man-In-The-Middle (MITM) attack. No platform specific optimizations were done in the actual implementation, thus performance should be similar over varied implementation platform. The whole procedure is randomized, thus resulting in a unique process for a unique session, which makes it harder for a cryptanalyst to find a base to start with. This technique is applicable to ensure security in message transmission in any form and in any size in wireless communication.

Some of the salient features of proposed technique can be summarized as follows:

a) Session key generation and exchange – Identical session key can be generate after the tuning of KSOFM in both sender and receiver side. So, no need to transfer the whole session key via vulnerable public channel.
b) Degree of security – Proposed technique does not suffers from cipher text only Attack, known plaintext attack, chosen plaintext attack, Chosen cipher text only attack, brute force attack.
c) Variable size key –128/192/256 bit session key with high key space can be used in different session. Since the session key is used only once for each transmission, so there is a minimum time stamp which expires automatically at the end of each transmission of information. Thus the cryptanalyst will not be able guess the session key for that particular session.
d) Complexity – Proposed technique has the flexibility to adopt the complexity based on infrastructure, resource and energy available for computing in a node or mesh through wireless communication. So, the proposed technique is very much suitable in wireless communication.
e) Key sensitivity – Proposed method generates an entirely different cipher stream with a small change in the key and technique totally fails to decrypt the cipher stream with a slightly different secret session key.
f) Trade-off between security and performance – The proposed technique may be ideal for trade-off between security and performance of light weight devices having very low processing capabilities or limited computing power in wireless communication.

In future, some other soft computing based approach can be used to generate the session key.

ACKNOWLEDGEMENT

The author expresses deep sense of gratitude to the DST, Govt. of India, for financial assistance through INSPIRE Fellowship leading for a PhD work under which this work has been carried out.

REFERENCES

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

[2] 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.

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

[4] Wolfgang Kinzel and ldo Kanter, "Neural cryptography" proceedings of the 9th international conference on Neural Information processing(ICONIP 03).

[5] Charles Pfleeger, Shari Lawrence Pfleeger, Security in computing, Third Edition 2003, pp 48, Prentice Hall of India Pvt Ltd, New Delhi.

[6] Biham, E. and Seberry, J."Py (Roo): A Fast and Secure Stream Cipher". EUROCRYPT'05 Rump Session, at the Symmetric Key Encryption Workshop (SKEW 2005), 26-27 May 2005.

[7] Chung-Ping Wu, C.C. Jay Kuo, "Design of Integrated Multimedia Compression and Encryption Systems", *IEEE Transactions on Multimedia, Volume 7, Issue 5*, Oct. 2005 Page(s): 828 – 839.

[8] HongGeun Kim, JungKyu Han and Seongje Cho."An efficient implementation of RC4 cipher for encrypting multimedia files on mobile devices". SAC '07 Proceedings of the ACM symposium on Applied computing, 2007, pp 1171--1175, NewYork, USA.

[9] Mantin and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4", Lecture Notes in Computer Science, Vol. 2259, Revised Papers from the *8th Annual International Workshop on Selected Areas in Cryptography, pp: 1 - 24, 2007.*

[10] Sarkar Arindam, Mandal J. K., "Artificial Neural Network Guided Secured Communication Techniques: A Practical Approach", Paperback: 128 pages, Publisher: LAP LAMBERT Academic Publishing (June 4, 2012), Language: English, ISBN-10: 3659119911, ISBN-13: 978-3659119910.

Authors

Arindam Sarkar

INSPIRE FELLOW (DST, Govt. of India), MCA (VISVA BHARATI, Santiniketan, University First Class First Rank Holder), M.Tech (CSE, K.U, University First Class First Rank Holder).

Jyotsna Kumar Mandal

M. Tech.(Computer Science, University of Calcutta), Ph.D.(Engg., Jadavpur University) in the field of Data Compression and Error Correction Techniques, Professor in Computer Science and Engineering, University of Kalyani, India. Life Member of Computer Society of India since 1992 and life member of cryptology Research Society of India. Dean Faculty of Engineering, Technology & Management, working in the field of Network Security, Steganography, Remote Sensing & GIS Application, Image Processing. 25 years of teaching and research experiences. Eight Scholars awarded Ph.D. and 8 are pursuing.