ORB ACADEMIC PUBLISHER

Int. J. Comput. Sci. Theor. App., 2014, vol. 2, no. 1., p. 25-31.
Available online at www.orb-academic.org

International Journal of Computer Science: Theory and Application
ISSN: 2336-0984

Random Keying Technique for Security in Wireless Sensor Networks Based on Memetics

S.B. Suman[1], P.V. Ranjith Kumar[2], E. Sandeep Kumar[3]

[1] *Dept. of Computer Science & Engg, M S Ramaiah Institute of Technology, Bangalore, Karnataka, India.*
[2] *Dept. of Electronics & Communication Engg, M S Ramaiah Institute of Technology, Bangalore, Karnataka, India.*
[3] *Dept. of Telecommunication Engg, JNN College of Engineering, Shimoga, Karnataka, India.*
Email: sandeepe31@gmail.com

ABSTRACT
Wireless Sensor Networks (WSNs) are often prone to risk of security attacks and vulnerabilities. This is because of the less human intervention in their operations. Hence, novel security mechanisms and techniques are of a prime importance in these types of networks. In this context, we propose a unique security scheme, which coalesce the random keying technique with memetics. The application of these kinds of bio-inspired computation in WSNs provides robust security in the network with the obtained results supporting the security concerns of the network.

KEYWORDS
Random Keying Technique — Memetics — Bio-Inspired Computation.

1. Introduction

Wireless sensor networks is gaining lot of research interest in the present scenario because of its vast and versatile applications. These networks are of great requirements in remote monitoring and military applications where they exchange sensitive data. Security is an area that has been a challenge for the researchers. This is due to the versatility and complexity in attacks to which these networks are often prone. Hence, in this paper we propose a random keying technique merging with the concepts of memetics to combat against the spoofing attacks in the network. Spoofing is a type of attack where the adversary tries to impinge unwanted or false information packets into the network to hamper its normal operation. Few researches have been carried out in solving the issues of WSNs using memetics.

Chuan- Kang Ting et al. [1] propose a scheme for improving the network lifetime by enabling more coverage using memetic algorithm for WSNs. Konstantinos et al. [2] propose a method for improving network lifespan using memetic algorithm as an improvement on the genetic algorithm, taking into accounts of communication parameters and overheads of the sensor nodes. Sandeep et al. [3] propose a novel biologically inspired technique that uses random keying technique with the concepts of artificial immune system for identifying the spoofing attacks in the network. Sandeep et al. [4] propose a bio-inspired approach for addressing node capture attack, which is a combination of artificial neural networks with the game theory as a combat mechanism against malicious attacker. Kashif et al. [5] propose a bio-inspired approach that uses Ant Colony Optimization (ACO) for routing, and artificial immune system for securing from abnormalities and routing attacks. Rongrong Fu et al. [6] developed a bio- inspired security framework that adapts Artificial Immune System (AIS) with the fuzzy techniques for detecting anomalies in the network. Ranjith et al. [7] proposed a bio-inspired security technique, which is based on genetics as counter measure against spoofing attacks.

According our knowledge, very few research works has been carried out using memetics in solving issues of WSNs and with respect to applications of memetics concept is a novel approach towards security. In the proposed work, we use a combination of random key distribution scheme with memetic concepts for providing robust security for WSNs. The algorithm was simulated in MATLAB and the results prove that the method is energy efficient compared to the other widely used cryptographic techniques like ECC and RSA, while combating against spoofing attacks.

The rest of the paper is organized as follows: section 2 deals with memetics, section 3 with the proposed methodology, section 4 with radio model, section 5 discusses the attack scenario, section 6 deals with the simulations, section 7 deals with the results and discussions, section 8 with the concluding remarks of the paper and finally the paper ends with few references.

2. Memetics

Memetics is a theory based on Darwinian evolution, originating from the popularization of Richard Dawkins book ‘the selfish gene’. A ‘meme’ is same as ‘gene’ and but these are termed as ‘units of culture’, which are “hosted” in the minds of one or more

individuals, and which can reproduce itself, thereby jumping from mind to mind [8]. The concept of 'memetics' has been developed as 'memetic algorithm', for solving optimization problems.

2.1 Memetic algorithm

Memetic algorithms have elements of Metaheuristic and Computational Intelligence. Although they have principles of evolutionary algorithms, they may not strictly be considered an evolutionary technique. Using ideas of memes and memetic algorithms in optimization may be referred to as memetics computing [9]. Ideally, memetic algorithms embrace the duality of genetic and cultural evolution, allowing the transmission, selection, inheritance, and variation of memes as well as genes. The memetic algorithm can simply be considered as the improvement over the genetic algorithm in the notion that, the genes are transferred directly to the individual but the memes are processed locally and then transferred. Hence, adding local search to the genetic algorithm results in memetic algorithm.

The algorithm is given below:

1. **Start**: Randomly generate a population of N chromosomes.
2. **Fitness**: Calculate the fitness of all chromosomes.
3. Create a new population:
   - **Selection**: According to the selection criteria, select two chromosomes from the population that are best chromosomes.
   - **Crossover**: Perform crossover on the two chromosomes selected.
   - **Local search**: search for the best chromosomes.
   - **Mutation**: Perform mutation on the chromosomes obtained with small probability.
4. **Replace**: Replace the current population with the new population.
5. **Test**: Test whether the termination condition is satisfied. If so, stop. If not, go to Step 2.

This algorithm is modified for providing security in the WSNs.

3. Proposed Methodology

This section deals with the method introduced in the regard of providing security in the network.

3.1 Random Key range distribution

3.1.1 At the Base Station (BS) - Set up phase

i - Set the range with in which the keys have to be selected. The keys (integer numbers) between these ranges are the initial set of populations (memes). Let this be (A,B).

ii - From the range (A,B) a random set of keys will be selected for scaling down the range, this indicates the optimal set of the keys, which participate in the further process. Let this be (X,Y), where X is the lower limit and Y is the upper limit.

iii - Within (X,Y), randomly two numbers will be picked, and sent to the Cluster Head (CH). Step iii is repeated until all the CHs receive two random numbers from the BS. Let the number received at the CHs be (p,q), where, p is the lower limit and q is the upper limit.

3.1.2 At the Cluster Heads

The received range from the BS will be sent all the member nodes of its cluster. This is (p,q), which is dealt in the previous section.

3.2 Steady phase communication

1. Ordinary member node, if it wants to communicate with its CH, it randomly picks two numbers from within the range (p,q). The numbers (keys) in the range (p,q) is the pool of population of memes. The chosen numbers in this pool indicates the best locally picked memes for the further processes. Let this be (m,n).
2. These memes are allowed to crossover with each other. The procedure of the crossover is dealt in the later sections of the paper.
3. The crossovered numbers (memes) are now checked for fittest candidate, for the further mutation process. The result of crossover will be two numbers, let this be (k,h) and out of two, one candidate is picked based on the presence of number of ones. The candidate is now allowed for mutation, whose process is explained in the further sections of this paper, the other number is kept as it is without any change. Let the picked candidate be h, and the result of the mutation be v, the result after the process is (k,v).
4. (m,n) is placed in the header and (k,v) is substituted as the trailer and the packet is sent to the CH.
5. The process is repeated by all the ordinary nodes in a network that wants to communicate with the CH. The respective CHs wait until it receives the data from all the ordinary nodes and again follows same procedure as in step 1 to step 4 and places header and trailer information in the packet and sends to the BS.

3.3 Crossover

Let the range received by the higher hierarchy node be (p,q). Select two numbers randomly and let this be (m,n). The step involved is given below:

1. Initially, calculate intermediate number,
$$E = (m+1)+(n-1); \quad (1)$$
2. Find the smallest multiple of 3 between the range (m,E), let this be x, else use m. 3 is an example this can also be made random, and depends on the robustness required.
3. Find (x%8), which gives the point at which (m,n) has to be crossovered. Here 8 is chosen since the size of keys chosen for communication is 8 bits. The example is shown below.

Ex :
$(p,q) = (12,70)$,
$(m,n) = (15,56)$;
$E = (15+1)+(56-1) = 71$;
The smallest multiple between (15, 71) is x= 15;

4. $(15\%8) = 6$; hence 6 is the crossover point. The bits from 6^{th} position to the 8^{th} position of (m,n) is crossovered with one another.

$m = 15 = 00001111$
$n = 56 = 00111000$
After crossover $\rightarrow 00101111 \rightarrow 47$
$\rightarrow 00011000 \rightarrow 24$

The result of crossover is $(m,n) = (47,24)$.

3.4 Mutation

The two bytes obtained after the crossover is checked for number of 1's individually and the key with the highest number of ones is chosen as the best candidate for mutation. From the example dealt in the crossover section, the best candidate chosen is 47 because it has more number of 1's in it.

The mutation is carried out in such a way that all the bits in the number are complemented.

Ex :
$47 \rightarrow 00011111$
$11100000 \rightarrow 223$

The packet is put with (15, 56) as the header and (47, 223) as the trailer information and sent to the higher hierarchical sensor node.

3.5 Verification at the CH for the packet sent by ordinary node or verification at the BS for the packet sent by CH

1. Start
2. Receive the packet
3. Extract header
4. Check header, whether it is in the range that was sent by itself. **Let the received header be *m, n* and trailer be *k, v*.**

   /*(p,q) range received by the higher hierarchy node*/

   if $(m \geq p$ *and* $n \leq q)$ **then**

   /* packet cleared stage-1*/

   (g,h) = Crossover (m,n);

   /*h_1 and h_2 are results of crossover*/

   Select the best candidate for mutation. Let this be x**.**

   (g_1,h_1)= Mutation (x);

   if $(g_1 == k$ and $h_1 == v)$ **then**

   /* packet cleared stage-2*/

   else

   /* packet is malicious*/

   end

   else

   /* packet is malicious*/

   end

5. Stop

3.6 Packet Description

i. Packet sent from BS to CH/ CH to its member nodes

MAC	p	q

where, MAC→ address of the intended CH node and p, q→ keys randomly picked.

ii. Packet sent from ordinary node to CH/ CH to BS

This packet consists of the details regarding randomly picked keys by the node and the trailer.

m	n	CRITICAL INFO	k	v

where, $m,n \rightarrow$ keys randomly picked by the node for communication with its CH and g_1, h_1 are the trailers after crossover and mutation, CRITICAL INFO → consists of various fields including, preamble, sync bits, destination address, type, group identity, length of message, counter for message sent, source address, error checking bits and payload.

4. Radio Model

The proposed methodology uses a classical radio model [10]. The sensor node is a transceiver. Hence, this radio model gives the energy consumed for the transmission and reception. The block diagram representation is shown in fig. 1. The radio model consists of transmitter and receiver equivalent of the nodes separated by the distance'd'. Where E_{tx}, E_{rx} are the energy consumed in the transmitter and the receiver electronics. E_{amp} is the energy consumed in the transmitter amplifier in general, and it depends on the type of propagation model chosen either free space or multipath with the acceptable bit error rate. We consider E_{fs} for free space propagation and E_{mp} for multipath propagation as the energy consumed in the amplifier circuitry. The transmitter and the receiver electronics depends on digital coding, modulation, filtering and spreading of data. Additional to this there is an aggregation energy consumption of E_{agg} per bit if the node is cluster head.

4.1 Energy Consumption

This section describes the energy consumed for communication.

Packet transmission

$$E_t = (L_P * E_{tx}) + (L_P * E_{amp} * d^n); \quad (2)$$

where, $L_P \rightarrow$ is the packet length in bits, and $n \rightarrow$ is the path loss component which is 2 for free space and 4 for multipath

propagation.
Suppose a node transmits a packet. Each bit in a packet consumes E_{tx} amount of transmitter electronics energy, E_{amp} amount of amplifier energy. A packet of length L_P, consumes an overall energy of E_t.

Packet reception

$$E_r = (L_P * E_{rx}); \tag{3}$$

where, $L_P \rightarrow$ is the packet length in bits.

Suppose a node receives a packet. Each bit in a packet consumes E_{rx} amount of receiver electronics energy. A packet of length L_p, consumes an overall energy of E_r.

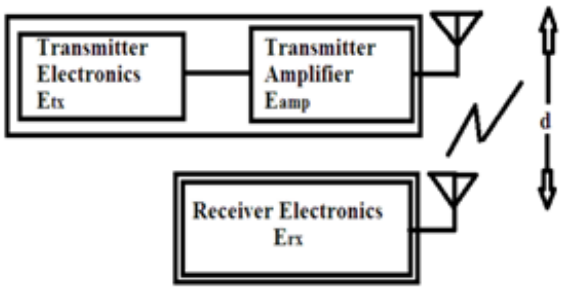

Figure 1. *Radio Model.*

5. Attack Scenario

The system relies on confusing the intruder by randomly varying the keys and ranges chosen for selecting the keys at the BS. The newly deployed malicious attacker may spoof unwanted packets to the CH or the BS. The attack scenarios are shown in the fig. 2 and fig. 3.

The new node carefully listens to the network paradigm and assigns its MAC address with that of another node, of which it may start disguising and spoofing packets to the higher hierarchical node. The packets follow the double verification steps and gets identified itself as either a legitimate or a spoofed packet. Suppose, the count of spoofed packets reaches above a pre-fixed threshold, an alarm is sent to the BS for preventing the further epidemic of the infected packet.

The spoofing can also be done at time by the legitimate nodes already deployed. The spoofing in this case can also be detected by the proposed methodology.

Since, the protocol protects the network using randomization concept, the attack not being identified is minimal. One scenario of attack was modeled in this paper, where a malicious node listens to the paradigm of the network and gets to know about the key ranges i.e. the keys are falling within the range (A,B), and puts header of the packet with those numbers and trailers with some random numbers. In this case, there are chances that the packet may pass the first verification stage, but the second stage clearance is difficult since the numbers in the headers has to undergo crossover, mutation and results has to match with the trailers. The results obtained for this scenario of attack is

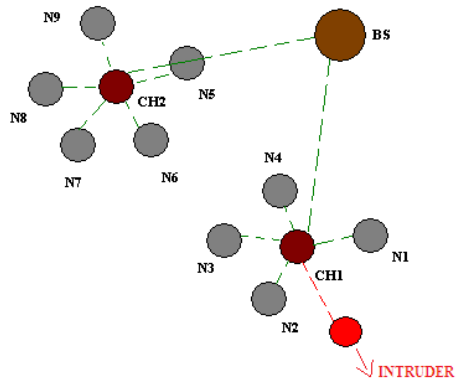

Figure 2. *Malicious ordinary node sending a false packet to CH.*

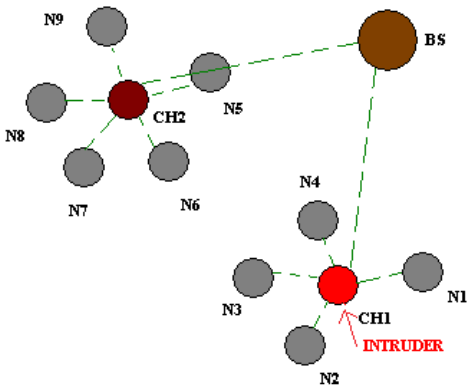

Figure 3. *Malicious CH sending a false packet to the Base Station.*

discussed in the fig. 4, fig. 5, fig. 6, fig. 7 and fig. 8. Apart from this case, if the malicious node has to successfully spoof the packet in every attack, then it has to get the algorithmic and mathematical details burnt in the node, which is the case of a node capture attack. The protocol fails if the node undergoes a capture attack and the security details are hacked.

6. Simulations

The algorithm was executed and tested using MATLAB 2013a on Intel core 5 Duo processor with windows operating system. CH requirement was set to 10% and the algorithm was verified on LEACH protocol till 1000 rounds. Table 1 contains the overhead in packet size due to the proposed security algorithm and table 2 depicts the various key sizes used for simulation. The parameters were set for modeling network environment is shown in table 3. The key sizes of ECC and RSA is shown in table 4, and of which the basic key size of 112 for ECC and 512 for RSA was considered for energy analysis.

Table 1. *Bits overhead due to cryptographic framework (per communication).*

Parameter	Value
Packet sent from BS to CHs	32 bits
Packet sent from CH to ordinary node	32 bits
Packet sent from end node to CH	32 bits
Packet sent from CH to BS	32 bits

Table 2. *Key sizes used in packets for communication.*

Parameter	Value
p, q	1 byte each
MAC	2 bytes
m, n	1 byte each
k, v	1 byte each

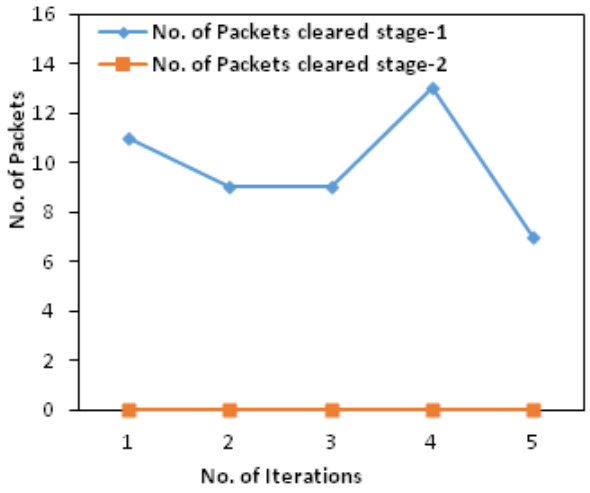

Figure 4. *Number of spoofed packets identified for five iterations (each for 100 rounds of LEACH).*

Table 3. *Radio characteristics and other parameters chosen for simulation.*

Parameter	Size
Number of nodes	100
Transmitter electronics, E_{tx}	$50nJ/bit$
Receiver electronics, E_{rx}	$50nJ/bit$
E_{mp}	$0.0013pJ/bit$
E_{fs}	$10pJ/bit$
E_{agg}	$5nJ/bit$
Length of plot	$100m$
Width of plot	$100m$
L_{pt}(packet sent from CH to Bs)	$6400bits$
L_{ctr}(packet sent from ordinary node to CH)	$200bits$
Initial energy of the node	$0.5J$

Table 4. *RSA and ECC key length comparison.*

RSA	ECC
512	112
1024	160
2048	224
3072	256
7680	384
15360	512

7. Results and Discussions

This section deals with the results obtained. The algorithm was tested on LEACH protocol. First five iterations are for analyzing the security, where number of rounds was limited to 100 in every iteration. Next, were five iterations each with 500 rounds. In both cases, after every fifth round a malicious packet was made to spoof into the network, the probability of being identified is checked, and the graph is plotted. It was observed that in both the cases, the accuracy in identifying the malicious packets was 100% as per fig.6 and fig. 7. In addition, in both the cases the number of packets clearing first stage and second stage of verification was plotted separately as per the fig. 4 and fig. 5.

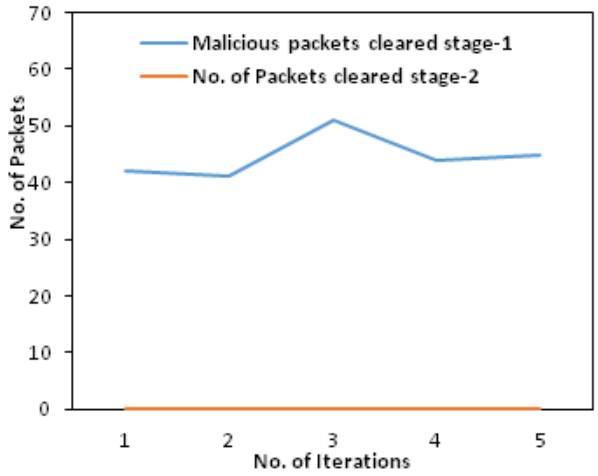

Figure 5. *Number of spoofed packets identified for five iterations (each for 500 rounds of LEACH).*

It was observed that even though the packets clear first stage, it was likely that they were caught in the second stage of verification; hence, the accuracy was always 100% and shows the robustness of the protocol in identifying the spoofed packets. The energy consumption analysis of our scheme with the existing cryptographic schemes like ECC and RSA was done for 100 rounds of LEACH for 10% of CH requirement, and the overhead in the energy consumption is plotted in fig. 8.

It was observed that the modeled memetics based algorithm (MA) is more energy efficient than the other widely used keying techniques and the results prove that the technique is robust in providing security in the network.

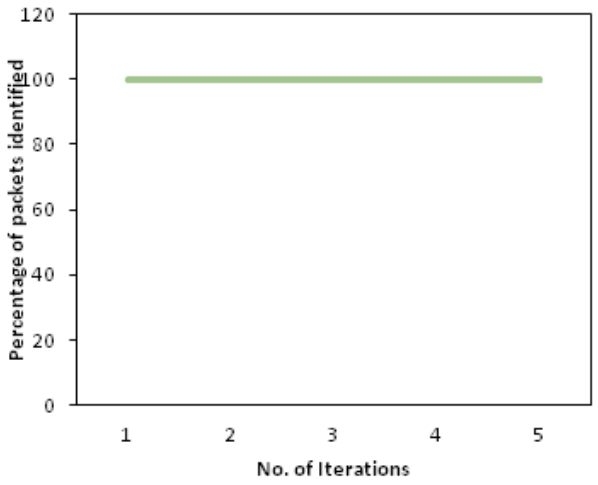

Figure 6. *Percentage of spoofed packets identified (each iteration with 100 rounds of LEACH).*

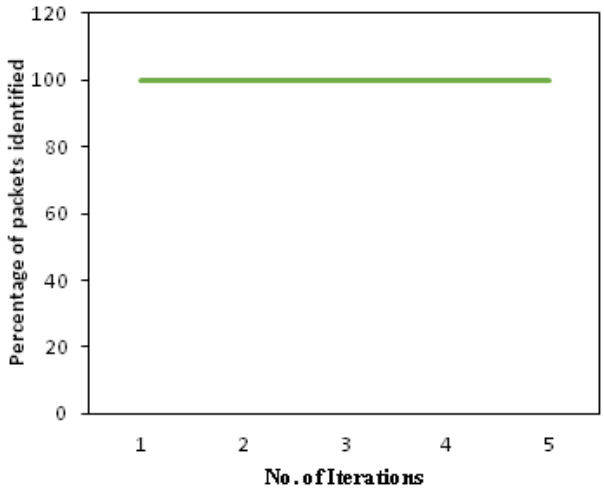

Figure 7. *Percentage of spoofed packets identified (each iteration with 500 rounds of LEACH).*

8. Conclusions

Security in WSNs has always been a difficult task in WSN to address. In this context, memetics based random keying approach is one of the contribution. It was observed a 100% accuracy in identifying the spoofed packets, since every packet has to undergo a double verification stage to get through into the network. In addition, the energy overhead calibration due to the use of the technique gives a promising result compared to other techniques. The developed algorithm uses confusion strategy in combination with memetics for identifying the spoofed packets. The obtained results prove that the algorithm can be implemented in the future networks with an ease.

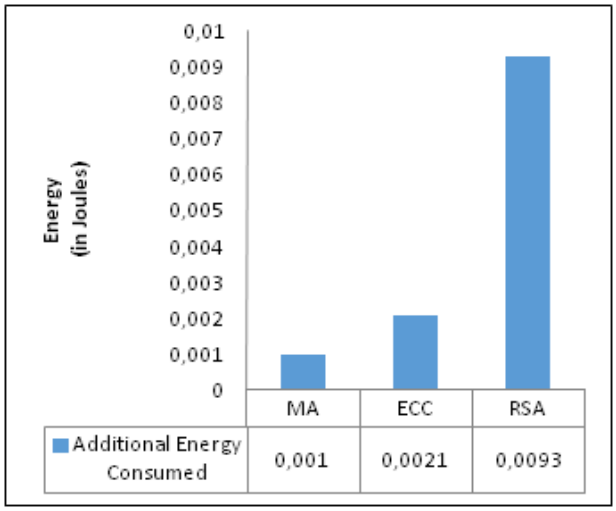

Figure 8. *Additional Energy consumed for 100 rounds between various cryptographic techniques.*

References

[1] TING, Chuan-Kang et LIAO, Chien-Chih. A memetic algorithm for extending wireless sensor network lifetime. *Information Sciences*, 2010, vol. 180, no 24, p. 4818-4833.

[2] FERENTINOS, Konstantinos P. et TSILIGIRIDIS, Theodore A. A memetic algorithm for optimal dynamic design of wireless sensor networks. *Computer Communications*, 2010, vol. 33, no 2, p. 250-258.

[3] KUMAR, E. Sandeep, KUSUMA, S.M., KUMAR, B.P. Vijaya, A Random Key Distribution based Artificial Immune System for Security in Wireless Sensor Networks, *Proc. of IEEE International Students' Conference on Electronics, Electrical and Computer Science (SCEECS)*-2014, 1-2 March, MANIT, Bhopal, Madhya Pradesh.

[4] KUMAR, E. Sandeep, KUSUMA, S.M., KUMAR, B.P. Vijaya, An Intelligent Defense Mechanism for Security in Wireless Sensor Networks, *Proc. of IEEE International Conference on Communications and Signal Processing (ICCSP)* - 2014, 3-5 April, APEC, Melmaruvattur, Tamil Nadu.

[5] SALEEM, Kashif, FISAL, Norsheila, HAFIZAH, Sharifah, et al. An intelligent information security mechanism for the network layer of WSN: BIOSARP. In : *Computational Intelligence in Security for Information Systems. Springer Berlin Heidelberg*, 2011. p. 118-126.

[6] FU, Rongrong, ZHENG, Kangfeng, LU, Tianliang, et al. Biologically Inspired Anomaly Detection for Hierarchical Wireless Sensor Networks. *Journal of Networks*, 2012, vol. 7, no 8, p. 1214-1219.

[7] V RANJITHKUMAR, P., P NEMAGOUD, Sandeep, KUMAR E, Sandeep, et al. A Novel Security Framework based on Genetics for Clustered Wireless Sensor Networks. *International Journal of Computer Applications*, 2014, vol. 96, no 5, p. 8-13.

[8] Concepts of memetics from, Memetics- http://en.wikipedia.org/wiki/Memetics

[9] Concepts of memetics from site nature- inspired algorithms: http://www.cleveralgorithms.com/natureinspired/physical/memetic_algorithm.html

[10] HEINZELMAN, Wendi Rabiner, CHANDRAKASAN, Anantha, et BALAKRISHNAN, Hari. Energy-efficient communication protocol for wireless microsensor networks. In : System Sciences, 2000. *Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2000. p. 10 pp. vol. 2.