LS-LEACH: A New Secure and Energy Efficient Routing Protocol for Wireless Sensor Networks

Muneer Alshowkan, Khaled Elleithy, Hussain AlHassan
Department of Computer Science and Engineering
University of Bridgeport
Bridgeport, USA
{malshowk, elleithy, halhassa} @bridgeport.edu

***Abstract*—Wireless sensor networks are facing many challenges such as the limited resource in processing power, storage and energy. The limited energy resource is one of the main challenges facing the security in such networks. This paper aims to improve the current security mechanisms in wireless sensor networks as well as reducing power consumption. LEACH protocol provides an energy routing protocol. However, it doesn't cover the security problems. Alternatively, this paper aims to provide an improved secure and more energy efficient routing protocol called LS-LEACH (Lightweight Secure LEACH). Authentication algorithm is integrated to assure data integrity, authenticity and availability. Furthermore, this paper shows the improvement over LEACH protocol that makes it secure and how to make it more energy efficient to reduce the effect of the overhead energy consumption from the added security measures.**

Keywords—attacks; wireless sensor; energy efficient; sensor security

I. INTRODUCTION

Wireless sensor networks form an infrastructureless wireless network where nodes are independent and self-organizing. Such networks provide an emerging technology that solve some of the environmental and social challenges by monitoring and collecting data related to the targeted applications [1, 2]. Notably, nodes perform a specific job by sensing the intended physical parameter. The key factor of using wireless sensors is because they have interesting characteristics such as low-power consumption and low cost.

Many applications of wireless sensor networks are designed to observe a variety of environments and collect data. The data is projected in different classifications based on their intended application. Applications of the aforementioned are related to nature, governments, and individuals. However, assuring data confidentiality, authenticity, availability, and integrity must be maintained. Security is one of the challenging aspects in wireless networks because it has effect on the sensors resource due to the very limited resources in the wireless sensors [3, 4]. Mobile and ad-hoc networks employ conventional security. Due to wireless sensor limitations it is hard to employ conventional security measures on wireless sensors networks. For example, it is inefficient to employ SSL protocol. SSL protocol requires a high amount of energy which is inefficient in wireless sensor networks. [5, 6].

In wireless sensor networks, there are many applications that require a high security level. For instance, military and health care applications. Such applications require maximum security. However, an increase in security consumes more resources [7]. When more resources are consumed it can negatively impact the lifespan of the network. Wireless sensors should have the maximum security with minimal power consumption to assure secure communication [8-10].

In the literature, many energy efficient protocols were proposed. LEACH protocol (Low-Energy Adaptive Clustering Hierarchy) [11] shown in Fig. 1 is a self-organizing and based on clustering hierarchy which is able to outperform the MTE (Minimum Transmission Energy) protocol by 8 times.

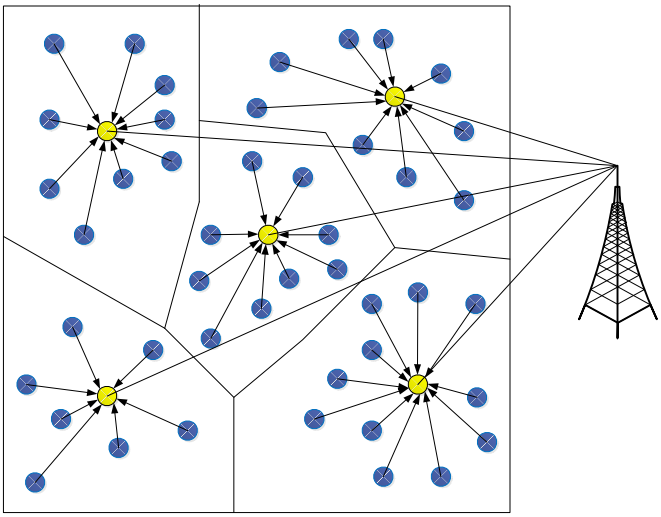

Fig. 1. LEAH Clustering Hierarchy

In this paper, we provide security measures to LEACH protocol after indicating the source and limitation of nodes. Also, we develop security measures to protect wireless sensors and the communications from possible attacks without compromising the network performance. For instance, securing LEACH protocol against denial of service attacks while maintaining its performance. Furthermore, the protocol assures that only the authenticated nodes are allowed to join and communicated in the network. At the other hand, we mitigate the overhead cost from the security measures applied to avoid compromising the network performance.

II. RELATED WORK

Security issues in wireless sensor networks are challenging especially the subject of network availability. Securing wireless sensor networks has been an active research field aiming to provide solutions to the different type of attacks that are related to confidentiality, integrity and availability. In [12], a dynamic solution was proposed to detect DoS attacks. This solution adopts the idea of LEACH protocol and adds a new node. Therefore, there are three types of nodes in the network which are sensing, analyzing and cluster head. Sensing nodes only performs sensing and the cluster head performs the necessary aggregation. However, the purpose of the new type of nodes analyzing nodes or controlling nodes are to analyze the traffic in each cluster. Once abnormal activity is detected, the controlling nodes make a report to the cluster head. Choosing the controlling nodes is based on the Multiplicative Linear - Congruential Generators to randomly choose nodes among the nodes with high energy remaining.

Li and Batten [13] proposed a solution to detect Path DoS where an attacker floods the communication paths with replayed or injected packets to disturb the communications medium. The proposed solution defines different types of nodes which are a member, aggregator, intermediate and sink nodes. Member nodes perform required sensing. An aggregator node collects data from the members and the intermediate nodes are the links between the aggregator nodes and the sink. However, the proposed solution has the some assumptions which might not be realistic. First, the mobile node has no power restrictions and it has a secure communication with the base station. Second, member nodes implement one-way hash function when sending data to the aggregator and their pre-distributed key is shared with the intermediate nodes. To detect network attacks, intermediate nodes verify the hash value before passing packets and report any abnormal behaviors.

A defensive framework against DoS attacks in wireless sensor networks was proposed in [14]. In order to detect and recover from several attacks such as network jamming, flooding and exhaustion the framework has two important stages which are the attack detection and the defense counter measurement. The framework consists of two networks; the sensor network and the defense network. The sensor network has four types of nodes which are sensor nodes, watch dog, cluster head and sink. Sensor nodes for data collection, watch dog sensors for communication monitoring, cluster head for data aggregation and the sink. There are several components responsible for communication, attack detection, defense and user control. In the attack detection, there are several detection modules to identify different types of DoS attacks. After the detection, it requests the countermeasure component to take the necessary action.

In [15] Stavrou and Pitsillides evaluated network recovery after different attacks, in order to develop a new protocol to improve the recovery in wireless sensor networks. The proposed protocol has two characteristics; detect malicious activity and isolate the infected node from network. Four procedures of evaluation were used; Blacklisting malicious nodes, Cryptographic keys revocation, Low duty cycle, and Channel hopping. The proposed protocol is able to speed up the intrusion detection process and fast recovery.

In [16], the authors observe the behavior of two protocols with specific algorithms. First protocol is TinySec with CBC-MAC algorithm. As a result, disadvantages are found such as the potential of message reply attack and network delay. The second protocol is Tiny ECC with Elliptic Curve Digital Signature Algorithm. This protocol proves to be more complex than the first protocol because it involves key distribution and management. At the other hand, processing time was faster in the first protocol than the second.

One important challenge that researchers extensively addressed is the limitation of energy in wireless sensor networks. In [17], the authors present a solution based on LEACH which they call E-LEACH. E-LEACH improves the lifespan of the network. When referencing the transmission of large amounts of data, E-LEACH is more efficient then LEACH. In regards to when the first node dies and half node dies, E-LEACH shows better performance. It changes the cluster head after every cycle to distribute cluster head energy consumption on all nodes. This results in minimizing the energy consumption so it can be used for security services.

III. PROPOSED PROTOCOL

The characteristics of the broadcast medium make the wireless sensor networks vulnerable to several attacks. An attacker could join the network and manage to intercept, eavesdrop, inject or transmit data. To solve most of the attacks, we have to successfully perform a number of tasks. First, deter the attackers from joining the network using light weight and energy-efficient authentication function where the cluster head verifies the authenticity of nodes requesting to join the clusters in an energy-efficient manner. Second, define a threshold for the normal node to node number of connections during time t. Thus, all the nodes in the network have to track the number of times any node initialized a connection with the corresponding node. This threshold is be used to detect any abnormal actively from a node trying to compromise the other nodes. Third, since LEACH utilizes a fixed TDMA schedule each node can only send data to the cluster head during that time. Another schedule should be used for each node specifying when the node is available to receive data from the cluster head.

A. Node Authentication

Initially, we assume that each node is equipped with two secret keys. One key shared with the base station and another key shared between all nodes. The private key shared with the base station is used when the node becomes a cluster head. However, the group key is used to join clusters. To deter the attacker from gaining access to the network, authentication should be done on both the cluster head when it elects itself and the nodes when they want to join the network. Nodes verify the authenticity of the node claiming to be the cluster head before they send their joining request. Once the nodes verify the cluster head authenticity, they can go ahead and make a joint request. The authenticity of the nodes requesting to join the cluster is verified by the cluster head before they become a member of that cluster.

Election for the next round cluster heads is done before the end of the current round and the wining node is authenticated by current cluster head to the base station afterward. Thus, nodes are notified by the base station and the current cluster head about the cluster heads elected for the next round.

B. *Detection Abnormal activity inside the network*

For further security and in case an attacker managed to join the network, we need to implement a detection process within the clusters. Considering the use of LEACH protocol, the communication takes place between the cluster head and the nodes and vice versa. Every node maintains a log to store the connection attempts from other nodes and a threshold should be defined for the number of possible connections with the node from any node during time t. When the connection attempts reach the threshold, the node should report those attempts to the cluster head. By the detection process, the attacker is detected before consuming the node's energy to avoid the attacker constantly initializing connection with the target node.

C. *Sending and Receiving TDMA*

Each node in the cluster has a specific time where it can transmit the data to the cluster head as in LEACH protocol. However another fixed schedule should be available between the node and the cluster head defining the time when the cluster head communicates with the node. The node starts to listen at specific times in case the cluster head has packets to be sent to the corresponding node [18].

D. *Deploying and Joining Clusters*

In this section we briefly discuss the required steps to form the network. These steps are necessary to prepare the sensors before the deployment and the required process after the deployment to elect, join and communicate between the base station, nodes and the cluster heads.

Step 1: All nodes are equipped with two keys. The first key will be shared with the base station and the second will be shared will all nodes for the initial phase to be used for cluster joining process.

Step 2: The cluster heads are elected and use their private key to communicate with the base station.

Step 3: Nodes use their group key to request joining the intended cluster as shown in Fig 2.

Step 4: After forming the clusters, the cluster head can update the cluster key providing a different key than the initial ones. Also, the base station can update the cluster head's private key if required.

Step 5: Electing a node for next round cluster head should be done before the end of the current round. The current cluster head verifies the authenticity of the new cluster to the base station and to the nodes.

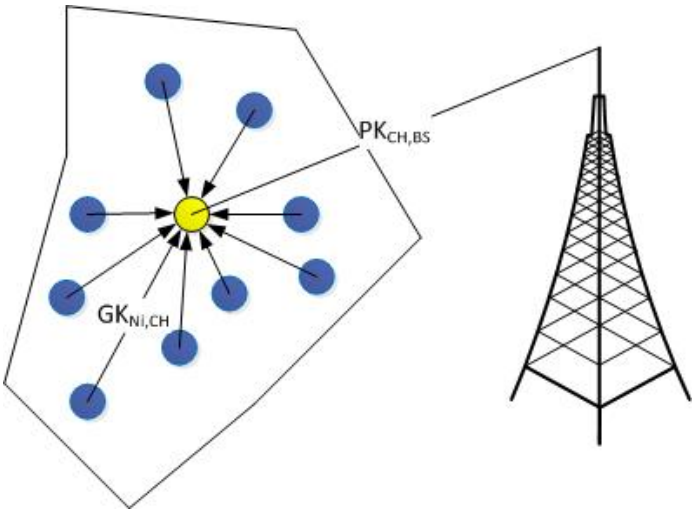

Fig. 2. Private and Group Key in one cluster

IV. LS-LEACH FOR WIRELESS SENSOR NETWORKS

In this protocol we assume that the cluster heads are 5% from the total number of the network nodes. The cluster heads will be elected after the network deployment and at the beginning of every cycle afterwards. The election of new cluster heads is from the nodes with the highest remaining energy. Then the current cluster head informs the base station about the authenticity of the elected cluster head. The message from the cluster head to the based station is encrypted by MAC algorithm with the shared key between the base station and the cluster head. After that, the base station broadcasts the list of the authenticated cluster heads to all nodes using uTESLA [19].

$$\mathbf{B.S.} \leftarrow {}_{\text{current}}\mathbf{CH}[\mathbf{MAC}[\mathbf{K_{N\text{-}BS}},\mathbf{M}]] \quad \mathbf{M}={}_{\text{new}}\mathbf{CH}$$

$$\mathbf{N_i} \leftarrow \mathbf{B.S.}[\ \boldsymbol{MAC}[\mathbf{K_{N\text{-}BS}},\mathbf{M},\ [\mathbf{K_{N\text{-}N}}]]]\ \mathbf{M}={}_{\text{new}}\mathbf{CH_i}$$

After broadcasting the list of the authenticated cluster heads, nodes can initiate a joint request to one of the cluster heads. The selection of cluster head should be based on the distance between the cluster and the node to reduce the energy required when communicating with each other.

A. *Election*

The election for the next round happens in advance in the current round. Nodes are elected as a cluster heads when they have more energy remaining than the other nodes ($N_i > {}_{\text{energy}}N_i$). In addition they must have a strong signal with base station ($N_i > {}_{\text{signal}}N_i$).

while [current round ≠ end]
  if $N_i > {}_{\text{energy}}N_i$ **and B.S** $\leftarrow N_i > {}_{\text{signal}}N_i$
    then ${}_{\text{new}}\mathbf{CH} \leftarrow \mathbf{N_i}$

B. Connection

After the election of the new cluster head and notifying the base station, the base station broadcast to all nodes the list of the new clusters head using uTESLA. Also it transmits the shared password to be used to join the new cluster heads ([K_{N-BS}, M, [K_{N-N}]]).

```
do
  if newCH = CH
      then B.S. ← currentCH[MAC[K_N-BS,M]]                4 bytes
          N_i ← B.S.[ MAC[K_N-BS,M, [K_N-N]]]             4 bytes
while [current round ≠ end]
```

At the beginning of a new round, the cluster head sends a verification message (verification [M]) with key (K_{N-N}) to neighbors' nodes. After receiving the message, nodes reply to the cluster head's request by a verification message encrypted by the shared key ([K_{N-N}, authentication [M]]) requesting to join the cluster. However, the cluster head needs to make sure that it doesn't allow the number of nodes to exceed the allowed number in cluster ($N_i \leq 20\ N_i$). On the other hand, nodes must request to join the clusters closer to them to reduce the energy consumption in receiving and transmitting ($_{new}$CH > signal $_{new}$CH).

```
while [current round ≠ end]
    then N_i ← newCH[MAC[K_N-N, M]]                       4 bytes
    while newCH ← N_i ≤ 20 N_i
       and newCH > signal newCH ← N_i
         then newCH ← N_i[MAC[K_N-N, M]]                  4 bytes
```

C. Transmission

The network nodes have three stases; sensing, listening/transmitting and sleeping. Sensing takes place when the nodes are sensing the environment. Listening/Transmitting happens when nodes are expecting to have communication with the cluster head or base station. Sleeping takes place when the nodes are node in sensing or listening/Transmitting modes. This requires the nodes to be in sleep mode to avoid the overhearing which consume nodes energy.

```
while [current round ≠ end]
   then if N_i ≠ sensing data or CH ← N_i ≠ sending data
            do sleep ← CHStatus
               sleep ← N_iStatus
        else
           Listen ← CHStatus
           Listen ← N_iStatus
           CH ← N_i[MAC[K_N-N, M]]                        4 bytes
```

Nodes are required to have a log for the connections attempts that are initialed with them. When the attempts reach a predefined threshold, a flag is raised to the cluster head and the base station. The base station has to perform the necessary actions in case the sensor is under attack.

```
while [current round ≠ end]
   then while N_i = sleep or CH = sleep
          if(CH ← newCH[MAC[K_N-N, [M]])
            CH ← report
          if(CH ← N_i[MAC[K_N-N, M]])
            B.S ← report
```

V. Simulation

The implementation software of LS-LEACH was carried using network simulator NS-2. NS-2.34 version was used in this implementation and simulation. NS2 is open source software under GPL (general public license). Moreover, NS2 is built in C++ and the interface is in OTcl language which is an object oriented extension of TCL language. Further, the operating system environment of the simulation was Linux Ubuntu 10.04 LTS installed on system with 2.5 GHz Intel Core 2 Duo and 4 GB memory.

The implementation of LS-LEACH was executed by adding new parameters and functions to the existing LEACH in NS2. Most of the changes were done in the source files of LEACH which are located in the ns-2.34 (as in version 2.34) directory in folder 'mit'. Other changes were also done in different files for the purpose of the linking of the TCL and C++ as TCL language is used as the interface for NS2 and OTcl is liking between TCL and C++.

A. Simulation Parameters

In simulating LS-LEACH, we have used the following parameters shown in Table 1.

TABLE I. Simulation Parameters

Parameter	Value
NS-2 Version	2.34
MAC Protocol	Sensors
Channel Type	WirelessChannel
Propagation	TwoRayGround
Queue	DropTail
Queue Length	100
Antenna	OmniAntenna
Area	1000 x 1000
Routing Protocol	LEACH
Number of Nodes	100
Number of Cluster	5
Nodes in Cluster	20
Simulation Time	3600 sec or CH < 5
Node Initial Energy	2j
Equal Energy (Start Up)	YES
CSThresh	1 nW
RXThresh	6 nW
Round Period	Each 20 sec

VI. Simulation Results

The performance of the system was measured using the system throughput, network life time and the total energy consumption.

A. System throughput

Fig. 3 shows the system throughput comparing between the classical LEACH protocol, and the proposed LS-LEACH. The proposed protocol has better performance because it tries to mitigate the ideal listening by putting the nodes in sleeping state which reduce the power consumption allowing the nodes to live longer and to reduce the collisions. The normal Leach protocol stopped performing because all the nodes died at time 360 which is after 19 rounds. However, the proposed protocol kept performing until time 475 which is after 24 rounds.

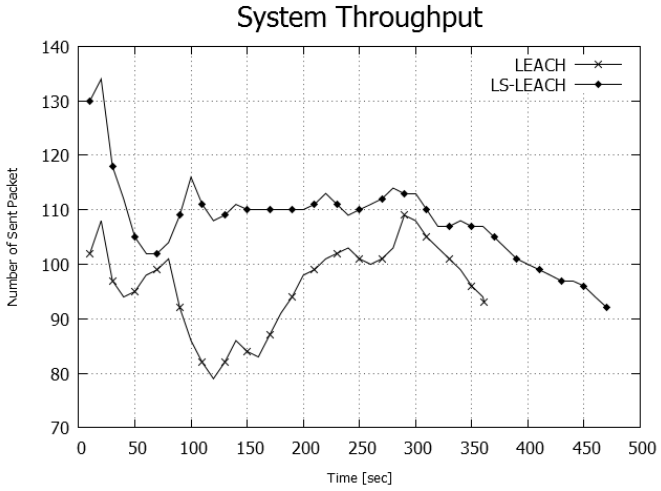

Fig. 3. System Throughput

B. Network Life time

Fig. 4 shows the comparison between the normal LEACH protocol and the proposed protocol in terms of network life time. At time 210 sec, normal LEACH protocol started to lose nodes, and by time 368, most nodes run out of power (when CH < 5).

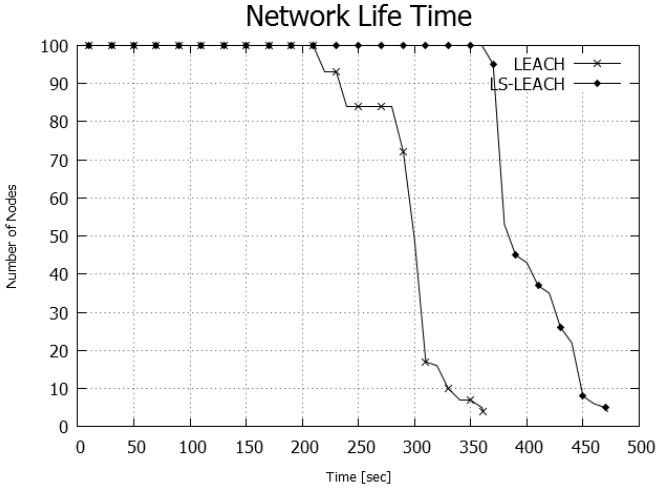

Fig. 4. Network Life Time

On the other hand, LEACH with security lost the first node at time 360 and lost most of the nodes (when CH <5) at time 471.

C. Energy Consumption

Comparing the energy consumption between normal LEACH and leach with security in Fig. 5, we find the proposed protocol has less power consumption. As a result, normal LEACH lasted until time 364 and the proposed protocol lasted until time 371. The increase of power consumption in LEACH protocol started at time 210 with the loss of the first node. As a result, the other nodes faced more load due to the increase of power consumption which reduced the network life. On the other hand, the proposed protocol lost the first node at time 360. This reduced the power consumption and increased the network life time.

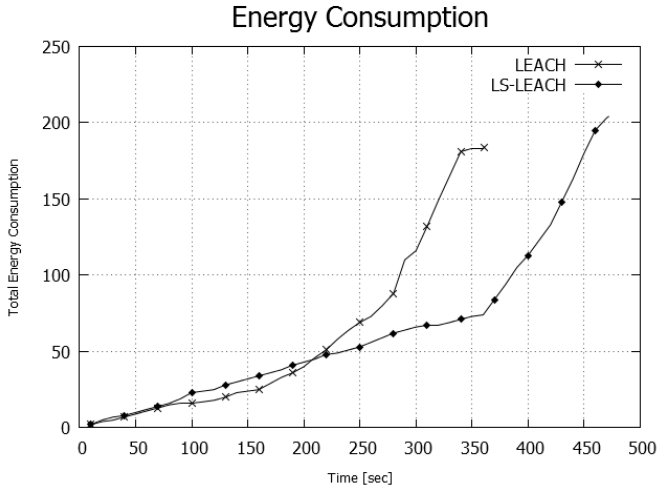

Fig. 5. Total Energy Consumption

VII. Comparision with other protocols

By mitigating the idle listening and overhearing, our protocol saved the network energy and prolonged the network life time. On the other hand, the other proposed protocols based on LEACH tried to only add overhead cost on the network without trying to save the network energy which reduced the network life time [4, 7, 20]. Furthermore, we used CBC MAC with 4 bytes (2^{32}) which is a reasonable authentication protocol as the sensor wouldn't stand a brute-force attack with 2^{32} attempts. As a result, it performs better than a public key cryptography or 2^{64} or 2^{128} CBC MAC.

VIII. Conclusions

In this paper we have introduced and implemented LS-LEACH which is an improvement of LEACH protocol. After improving LEACH protocol power consumption and adding the security measures, the protocol performed better in terms of the system throughput, network life time and the total energy consumption. The proposed protocol provided a secure authentication protocol for the network where the new nodes requesting to join the network must be authenticated in order to join the network.

REFERENCES

[1] [M. V. Ramesh, A. B. Raj, and T. Hemalatha, "Wireless Sensor Network Security: Real-Time Detection and Prevention of Attacks," in Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on, Mathura, 2012, pp. 783-787.

[2] L. Gheorghe, R. Rughinis, R. Deaconescu, and N. Tapus, "Authentication and Anti-replay Security Protocol for Wireless Sensor Networks," in Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on, Nice, France, 2010, pp. 7-13.

[3] M. Rahman, S. Sampalli, and S. Hussain, "A robust pair-wise and group key management protocol for wireless sensor network," in GLOBECOM Workshops (GC Wkshps), 2010 IEEE, Miami, FL, 2010, pp. 1528-1532.

[4] M. El-Saadawy and E. Shaaban, "Enhancing S-LEACH security for wireless sensor networks," in Electro/Information Technology (EIT), 2012 IEEE International Conference on, 2012, pp. 1-6.

[5] H. Soroush, M. Salajegheh, and T. Dimitriou, "Providing transparent security services to sensor networks," in Communications, 2007. ICC'07. IEEE International Conference on, Glasgow, 2007, pp. 3431-3436.

[6] D. Martynov, J. Roman, S. Vaidya, and H. Fu, "Design and implementation of an intrusion detection system for wireless sensor networks," in Electro/Information Technology, 2007 IEEE International Conference on, Chicago, IL, 2007, pp. 507-512.

[7] L. Sang Hyuk, L. Soobin, S. Heecheol, and L. Hwang-Soo, "Wireless sensor network design for tactical military applications : Remote large-scale environments," in Military Communications Conference, 2009. MILCOM 2009. IEEE, 2009, pp. 1-7.

[8] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of Security Issues in Wireless Sensor Networks," in Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on, 2011, pp. 308-311.

[9] D. E. Burgner and L. A. Wahsheh, "Security of Wireless Sensor Networks," in Information Technology: New Generations (ITNG), 2011 Eighth International Conference on, 2011, pp. 315-320.

[10] A. Blilat, A. Bouayad, N. El Houda Chaoui, and M. E. Ghazi, "Wireless sensor network: Security challenges," in Network Security and Systems (JNS2), 2012 National Days of, 2012, pp. 68-72.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in International Conference on System Sciences, Maui, Hawaii, 2000, pp. 1-10.

[12] M. Guechari, L. Mokdad, and S. Tan, "Dynamic solution for detecting denial of service attacks in wireless sensor networks," in IEEE ICC Ad-hoc and Sensor Networking Symposium, Ottawa, ON, Canada, 2012, pp. 173-177.

[13] L. Bai and L. Batten, "Using Mobile Agents to Detect Node Compromise in Path-Based DoS Attacks on Wireless Sensor Networks," in Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on, Shanghai, China, 2007, pp. 2507-2510.

[14] Y. Xin, B. Tian, Q. Li, J.-y. Zhang, Z.-M. Hu, and Y. Xin, "A Novel Framework of Defense System Against DoS Attacks in Wireless Sensor Networks," in Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on, Wuhan, 2011, pp. 1-5.

[15] E. Stavrou and A. Pitsillides, "Vulnerability assessment of intrusion recovery countermeasures in wireless sensor networks," in Computers and Communications (ISCC), 2011 IEEE Symposium on, Kerkyra, 2011, pp. 706-712.

[16] V. Cionca, T. Newe, and V. Dadarlat, "On the (im) possibility of denial of service attacks exploiting authentication overhead in WSNs," in Sensors Applications Symposium, 2009. SAS 2009. IEEE, 2009, pp. 74-79.

[17] J. Xu, N. Jin, X. Lou, T. Peng, Q. Zhou, and Y. Chen, "Improvement of LEACH protocol for WSN," in Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, 2012, pp. 2174-2177.

[18] Y. Wei, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE/ACM Transactions on Networking, vol. 12, pp. 493-506, 2004.

[19] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," Wireless networks, vol. 8, pp. 521-534, 2002.

[20] L. B. Oliveira, H. C. Wong, M. Bern, R. Dahab, and A. A. Loureiro, "SecLEACH-A random key distribution solution for securing clustered sensor networks," in Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on, 2006, pp. 145-154.