SDLM: Source Detection Based Local Monitoring in Wireless Sensor Networks

Prathap U, Nisha K B, Deepa Shenoy P and Venugopal K R
Department of Computer Science and Engineering
University Visvesvaraya College of Engineering
Bangalore University, India
prathap.u@gmail.com

***Abstract*—Security in wireless sensor networks is critical due to its way of open communication. Local monitoring is one of the powerful technique to secure the data and detect various malicious activities. In local monitoring, neighbour nodes observe the communication between current sender, current receiver and next hop receiver to detect the malicious activity. To make sensors power efficient, sleep-wake scheduling algorithms along with local monitoring are suggested in literature. Solutions in the literature do not address the problem if source node is malicious and do not consider unnecessary wake up of the nodes as malicious activity. This paper tries to achieve without assuming source node as honest and considers unnecessary wake up of the node as a malicious activity. Simulated the algorithm in NS-2 and performance analysis is discussed. Even with additional checks applied to detect malicious activities, analytical results show no degradation in the performance.**

Keywords*—*WSN, local monitoring, malicious node

I. Introduction

A wireless sensor network (WSN) consists of spatially distributed autonomous devices having sensing and communication capability to cooperatively monitor physical or environmental conditions, such as temperature, pressure, sound, vibration, motion or pollutants. Wireless sensor networks are used in environmental conditions where information is difficult to access. Sensor node, also known as a 'mote', is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. Sensor network transmits the data from one node to another node in an adhoc way and finally to a base station where the data is stored, processed and displayed.

Sensor nodes can be deployed either in a controlled environment or in a uncontrolled environment. In the uncontrolled environments, the sensor networks are vulnerable to a wide range of security attacks. Attackers can eavesdrop on radio transmissions, inject data in the channel, replay previously heard packets, drop the packets which supposed to be transmitted and many more. Attackers may deploy a few malicious nodes with similar or better hardware capabilities as the legitimate nodes that might collude to attack and disrupt the system cooperatively. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. Also, in some cases colluding nodes might have good-quality communication channels available for coordinating their attack. Sensor nodes may not be tamper resistant and if an adversary compromises a node, one can extract all key information, data, and program stored on that node.

Cryptographic techniques alone are not sufficient to protect the data. Attacks such as wormhole, rushing attacks can be launched without the help of cryptographic keys. In order to provide security to wireless sensor networks a technique called local monitoring has been used in literature. In local monitoring, designated nodes observe part of the traffic going in and out of their neighbours. The designated nodes at each hop of packet transmission are also called guard nodes. Guard node does different types of checks locally on the observed traffic to make a consensus determination of malicious behaviour of a node. Though local monitoring can be used as powerful technique for enhancing security of WSNs, it consumes more energy since the monitoring node is continuously awake to oversee the network activity. So in order to optimize the energy overhead of monitoring and maintaining the effectiveness of monitoring service *Energy Aware Local Monitoring* (ELMO)[1] has been used. ELMO uses sleep wake scheduling technique for monitoring purpose. It provides the same level of security that was attained with local monitoring [2] and avoids the unnecessary awake of node.

In ELMO, the malicious behaviour of sending unwanted wake up signals to a node is not addressed and also source node is assumed to be honest. In ELMO, whenever a source node wants to transmit data packet to a destination node, it sends wake up signals to its neighbours and then transmit data to next hop without checking whether the monitoring (guard) nodes are actually activated or not. In the proposed SDLM system, i) the guard nodes send an acknowledgement to sender indicating that they are activated, ii) unnecessary wakeup of nodes without sending the actual data is treated as a malicious activity, iii) guard nodes also observes the source node for the malicious activity. The SDLM system is an improved method on the existing local monitoring technique called ELMO. We achieved SDLM with additional checks to detect the malicious behaviour and keeping the checks of ELMO intact.

We provide a theoretical performance analysis showing the comparison between the ELMO and our approach SDLM with various parameters. Even though we have added additional checks to detect the malicious activity, there is no degradation in the performance. The rest of the paper is organized as follows, section 2 discusses about the related work, section 3 describes the background of local monitoring and problem statement, section 4 presents the solution and

978-1-4799-8641-5/15/$31.00 ©2015 IEEE

algorithm, section 5 provides the performance analysis and results, and section 6 concludes the work and discusses the future challenges.

II. Related Work

Local monitoring is a popular technique to provide security in WSNs. LITEWORP [2] provided a lightweight countermeasure for wormhole attack and suitable for resource constrained multi-hop networks. SLAM [3] approach used local monitoring for detecting the data attacks and integrated the local monitoring with sleep wake protocol. SLAM partially addressed the problem of combining local monitoring and sleep-wake protocol under the assumption that malicious nodes does not have any ability to control the transmission power level. In [4] author proposed a protocol called DICAS that mitigates the data attacks by detecting, diagnosing, and isolating the malicious nodes. DICAS uses the ability of a node to oversee its neighboring nodes communication. In [5] author proposed a protocol called SECOS that mitigates the data attack problems in static sensor networks. SECOS divides the sensor nodes into control groups, each group with a control node. Data exchange between nodes within a control group happens through the control head which provides the common key. The keys are changed periodically and the control nodes are chosen periodically to enhance security. Lot of work carried out in the literature and shown that local monitoring is a feasible approach to counter data attacks [6], [7].

However, the application of local monitoring interferes with sleep-wake scheduling. In the ELMO [1] approach author addressed the problem of combining local monitoring (to support security) and sleep-wake scheduling (to conserve energy) under the no assumption that a malicious node have the ability to control its transmission power level. ELMO approach considered scheduling of the monitoring nodes with the goal of additional energy savings. ELMO provided the first methodology for enabling energy-efficient local monitoring in multi-hop wireless sensor networks.

III. Background and Problem Definition

In local monitoring neighbour nodes observe the data reception and forwarding from a node and decides the malicious behaviour based on certain conditions. Among the neighbour nodes, there are designated nodes which satisfy the guard selection criteria [1] participates in the observation and malicious activity detection.

In the figure 1, n_1 is the source node trying to transmit data to n_h node, where n_h is the destination node over *h-1* hops distance. Guard nodes are identified at every hop. Guard nodes satisfy the below conditions. 1. If n_1 transmitting data to n_2 , $GN(n_2) = \{n_1, g_2, g_3\}$ where $GN(n_2)$ is set of guard nodes participate in the observation on n_2's behaviour and $GN(n_2) = R(n_1) \Omega R(n_2) - n_2$. Where $R(n_1)$ is radio range of n_1 and $R(n_2)$ is the radio range of n_2. $R(n_1) = \{n_1, g_1, g_2, g_3\}$, $R(n_2) = \{n_2, g_2, g_3, g_4, g_5\}$. 2. Distance between n_2 & g_i should be greater than the distance between n_2 & n_3. This is to detect the attacks by adjusting the transmission power level control.

Each guard node has a watch buffer used to store the information for each packet sent from packets immediate sender to packets immediate receiver. For example, if n_1 is transmitting data to n_2, the guard nodes g_2 and g_3 stores information for each packet sent from n_1 to n_2 in its watch buffers. The information includes source, destination, packets immediate sender, packets immediate receiver, first hop of packets immediate receiver, data which is transmitted and the current time. Each entry in the watch buffer has a time threshold, by which the packets immediate receiver must forward the packet unless that one itself is the destination. Each packet forwarded by the packets immediate receiver is checked for the corresponding information in the watch buffer by guard nodes.

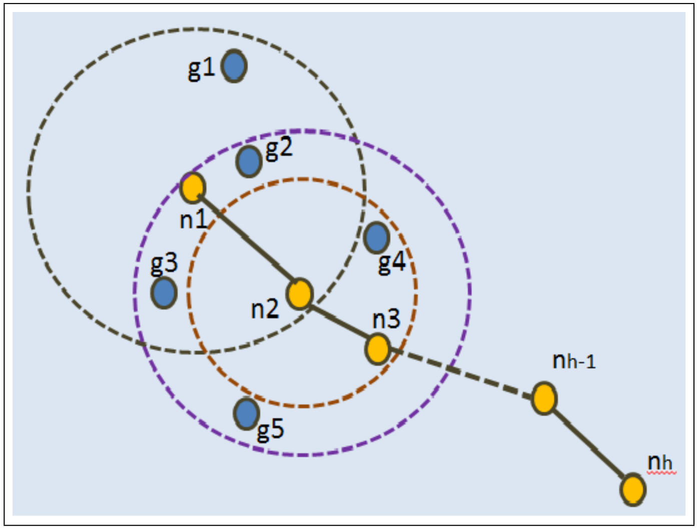

Fig. 1. network model

At each guard node g_i a malicious counter is maintained for a node n_j. Whenever the guard node g_i detects the malicious activity of node n_j the malicious counter is incremented. The increment of malicious counter depends on the nature of the malicious activity detected. A node is determined to be misbehaving only if the malicious counter value goes above a threshold value. Then guard node g_i removes n_j from its neighbour list and sends a warning message to each neighbour of n_j indicating that n_j is a suspected malicious node. Then all the neighbours of n_j, removes n_j from its neighbouring list, and isolates n_j by marking n_j's status as revoked in the neighbour list. After isolation the neighbour nodes does not send or accept any packet to a revoked node.

System Assumptions: SDLM assumes the network is static and the links are bidirectional. SDLM assumes that network nodes have pre-distributed keys for encryption and data communication as discussed in [8]. In SDLM each node knows the current *(X,Y)* location and also the location of the neighbour nodes. Malicious behaviour is manifested through delaying, dropping, fabricating, misrouting or modifying packets. In general most of the local monitoring algorithms assume that the nodes in the network are dense and suitable set of neighbouring nodes are available to become the guard nodes.

Goal of the SDLM is to detect the malicious behaviour of the source node such as unnecessary wake up of the neighbouring nodes without having the data to transmit or with the bad intention of draining the power of the neighbouring nodes with unnecessary wakeup. Current sending node also have the confirmation from guard nodes whether they actually woke up to observe the malicious behaviour of current receiver by receiving the acknowledgement from each guard node.

IV. SDLM

The main design goal of SDLM is to find out the malicious activity of sending unwanted wake up signals to a node with the purpose of wasting the energy of nodes in the network. SDLM also checks whether the guard nodes are actually woke up or not before sending data to a node. We use figure.1 to illustrate SDLM. Whenever source node n_1 wants to transmit data to a destination node n_h through an *h-1* hop route $n_1 \rightarrow n_2 \rightarrow ......n_{h-1} \rightarrow n_h$, SDLM uses the following technique.

Node n_1 wakes up by itself and broadcasts wake up beacon to all its neighbours. Each neighbour of n_1 after being awakened sends an acknowledgement back to the sender (n_1) indicating that each of them are now in active mode. Then the neighbour nodes determine whether to stay awake or go back to sleep mode depending on the role it plays in the data transmission. If the neighbour node is the next hop in the ongoing communication then it stays awake to receive data and to monitor the next communication link. If the neighbour node is a guard node then also it stays awake to monitor the communication link. To check whether a node is guard node or not it runs guard selection algorithm [1]. All other neighbour nodes go back to sleep mode.

SDLM checks whether a node sends unwanted wake up signal or not in addition to other malicious activities. In order to do that a node after being awakened wait for a particular time threshold (T_{th}) for getting a data packet from the node that sends wake up signal to it. If it does not get any data packets within T_{th} then it broadcasts a two hop warning message indicating that sender is doing a malicious activity. Then all the nodes that are in wake up mode will go back to sleep mode.

If the source node is honest it sends data packets to n_2. Node n_2, guards of n_2 (n_1, g_2 and g_3) have to continue listening the communication link for a maximum of time T_w (watch buffer time threshold). Each time a new packet is send from n_1 to n_2, T_w is initialized again. The nodes after being awakened go back to sleep mode whenever the time threshold T_w expires. The guard nodes g_2 and g_3 check whether there is any malicious activity carried out by the node n_2 as explained in section 3. Here the guard nodes identify four types of malicious activities: delay, misrouting, modification and no awakening of guard nodes [1]. If the node n_2 is honest it does the same steps as n_1 did to wake up its guards (g_4 and g_5) and the next hop (n_3). If node n_2 does not send wake up signal, the guards of node n_2 with lowest *ID* sends a two hop wake up signal. If that guard fails then next guard with lowest *ID* sends wake up signal and so on. This process continues until the packet reaches destination.

Notations

Ns: source node
Nd: destination node
NOH: method which provides the number of hops between a pair of nodes
NBi: neighbours of node i
sigWP: wake up signal
ACK: acknowledgement
NH(i): next hop from i while transmitting data packet
MODE(i) : represents whether the node is in SLEEP mode or WAKE UP mode
GUARDNH(i): represents guard nodes to monitor the behaviour of NH(i)
warMSG: warning message
Tth: threshold time

ALGORITHM

```
SDLM( Ns,Nd)
1. L= NOH(Ns,Nd)
2. If L>1 then
3. i=Ns
a.    Do
i.       For each j in NBi
1.           sigWP to j from i
2.           If MODE(j) = WAKE UP
a.               ACK to i from j
ii.       End for
iii.       If NBi=NH(i) then
1.           MODE(NBi)= WAKE UP
iv.       If NBi=GUARDNH(i) then
1.           MODE(NBi)= WAKE UP
v.       Else
1.           MODE(NBi)= SLEEP
1.           Two hop warMSG from GUARDNH(i)
2.           MODE(NBi,i) = SLEEP
vii.       Else
1.           NH(i) receive data from i
2.           If time !=Tw then
a.               MODE(i,GUARDNH(i)= WAKE UP
viii.       i=NH(i)
b.    While(i!=Nd)
4. End
```

V. Performance Analysis and Results

The effectiveness and efficiency of the proposed scheme are evaluated in the ns-2 simulator. we have compared SDLM the proposed approach with the ELMO [1]. 100 nodes are randomly deployed in a square area. Each node acts as a source. The malicious nodes are selected at random. Any pair of nodes can act as source and destination, if the source/destination pair is more than two hops apart. Even with additional malicious activity detection, SDLM achieved the results intact with ELMO. In order to find out the worst case end to end delay of SDLM, we compared it with ELMO. End to end delay is the time taken for a data packet to reach the final destination. Since ELMO and SDLM uses a sleep wake scheduling technique for local monitoring, the end-to-end delay of these two techniques is always higher compared to other monitoring techniques without sleep-wake scheduling. The end-to-end delay imposed by SDLM depends on the number of hops between the source and destination. As shown in Figure 2 the end-to-end delay of ELMO and SDLM increases linearly with the number of hops. In SDLM delay is little higher than that of ELMO since in this technique a node transmits data only after getting acknowledgement from guard nodes.

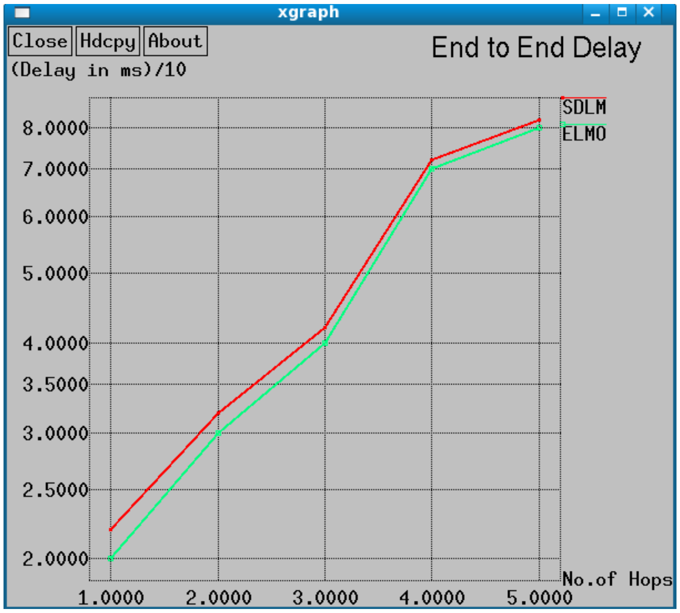

Fig. 2. end to end delay

A. *Average Energy Balance*

We also analyze the energy balance of sensor nodes with the proposed scheme in WSN. Since the energy consumption is mainly due to continuous monitoring of guard nodes to oversee the network activity, here we are using a sleep wake scheduling technique for monitoring. In SDLM nodes that are involved in current communication link (i.e. sender, receiver and the corresponding guard nodes) are awake and all other nodes are asleep. Figure 3 shows the average energy balance of nodes when we are conducting simulation experiments on ELMO and SDLM. In both cases the performance is almost same since both techniques uses the sleep wake scheduling technique. In the figure it is shown that the average energy balance of nodes slightly decreases as time increases. This makes SDLM more energy efficient as SDLM handles better security check compare to ELMO.

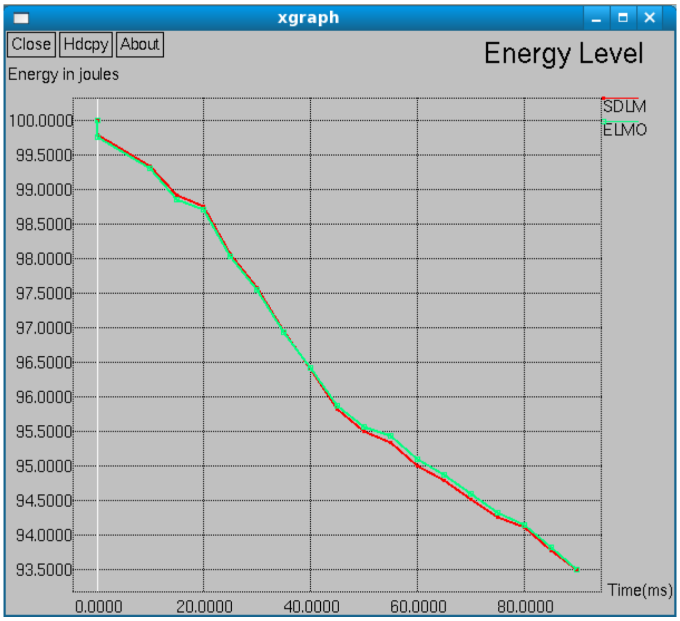

Fig. 3. Average Energy Balance

B. *Percent Delivery Ratio*

It indicates the percentage of the transmitted data packets that are successfully received. Firstly the total number of transmitted packets are counted, followed by the total number of received packets and the total number of dropped packets. The delivery ratio is calculated as the percent of packets received to the packets transmitted.

*% delivery ratio= (No. of packets received / No. of packets transmitted)*100*

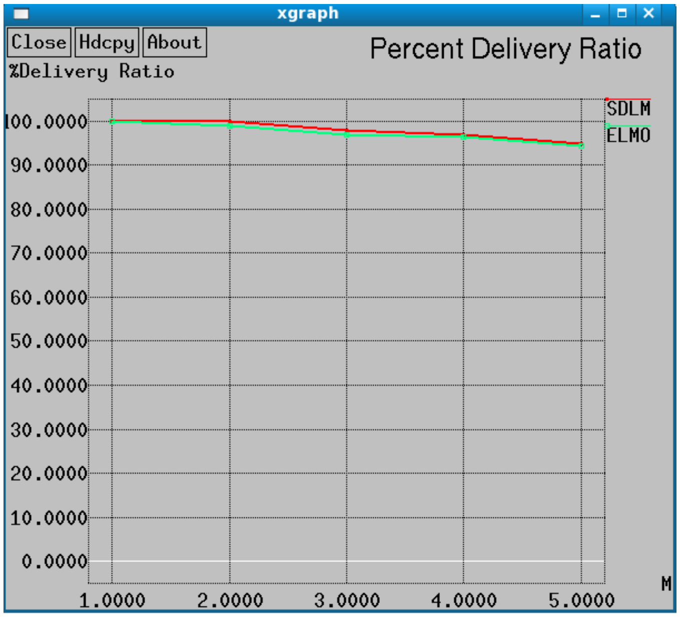

Fig. 4. Percent Delivery Ratio

Figure 4 shows the variations of percent delivery ratio as we vary the number of malicious nodes. Here we compare the percent delivery ratio of ELMO and SDLM. The performances of both techniques are almost same. It is shown that the percent delivery ratio slightly decreases as the number of malicious nodes increases. This is because some of the packets are dropped before the malicious nodes are detected and isolated. As the number of malicious nodes increases, the initial packets dropping also increases so the percent delivery ratio decreases.

C. *Percent Isolation*

It indicates the percentage of malicious nodes in the network that are isolated as a fraction of total number of malicious nodes.

*% isolation= (No. of malicious nodes isolated/ No. of malicious nodes) * 100*

Figure 5 shows the variations of percent isolation as we vary the number of malicious nodes. Here we compare the percent isolation of ELMO and SDLM. The performances of both techniques are almost same. It is shown that the percent

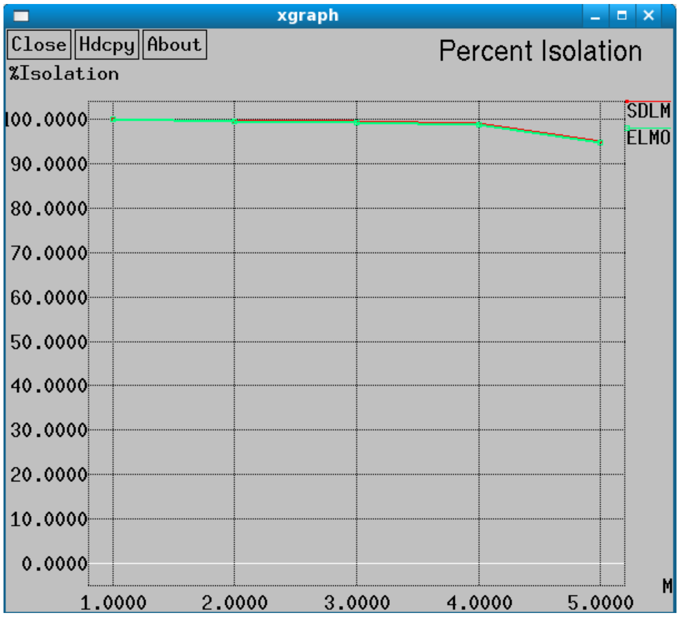

Fig. 5. Percent Isolation

isolation slightly decreases as the number of malicious nodes increases. This is because number of guard nodes decreases as the number of malicious nodes increases.

D. Percent Wakeup Time

It indicates the time a node has to be awake to do monitoring all the nodes as a percentage of simulation time. Here we compare the energy savings of SDLM with ELMO.

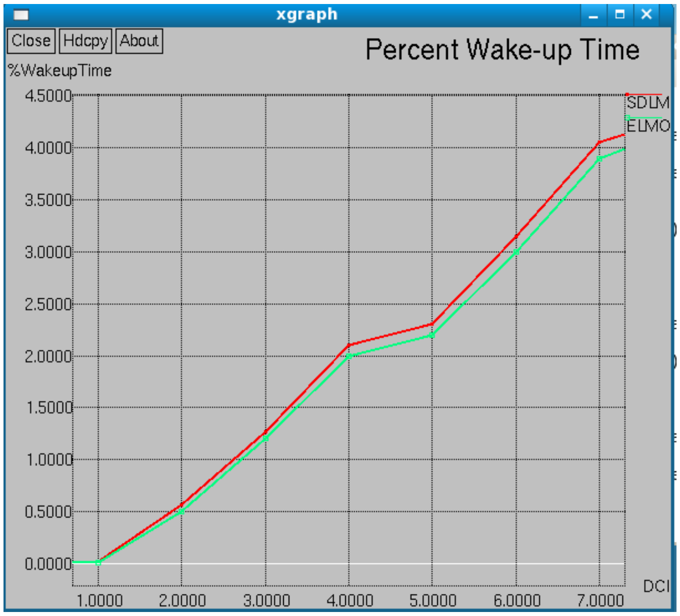

Fig. 6. Percent Wakeup Time

The performances of both techniques are almost same. Figure 6 shows the extra time a node needs to be awake for monitoring in SDLM and ELMO as we vary the detection confidence index. From the figure it is shown that the percent wake up time increases linearly as we increase detection confidence index. This is because if we increase detection confidence index [1] the number of guard nodes that are used for monitoring purpose also increases there by the extra time a node needs to be awake for monitoring purpose also increases.

VI. Conclusions and Future Directions

Local monitoring approach detects various kinds of malicious activities. SDLM achieved the detection of source node malicious activity and unwanted wake up malicious activities along with the malicious activities detected by ELMO. Based on the obtained performance results, SLDM performs better than ELMO. In Local Monitoring techniques, guard nodes themselves can inject many malicious activities. Need further analysis on identifying the malicious guard nodes and mitigating the security attacks from guard nodes along with local monitoring techniques.

References

[1] Issa M. Khalil. ”elmo: Energy aware local monitoring in sensor networks.”. In *IEEE Transactions on dependable and secure computing*, volume 8, pages 523–536, August 2011.

[2] I Khalil, S.Bagchi and N.B.Shroff. ”liteworp: Design and analysis of a protocol for detection and isolation of the wormhole attack in multihop wireless networks.”. In *Proc. Elsevier computer networks J.*, volume 51, pages 3750–3772, Sept 2007.

[3] I Khalil, S.Bagchi and N.B.Shroff. ”slam: Sleep-wake aware local monitoring in sensor networks.”. In *Proc. 37th IEEE Depndable Systems and Networks Conf. (DSN ’07)*, pages 565–574, June 2007.

[4] I Khalil, S.Bagchi and C. Nina-Rotaru. ”dicas: Detection, diagnosis and isolation of control attacks in sensor networks”. In *IEEE/CreateNet SecureComm*, pages 89–100, Sept 2005.

[5] I Khalil, S.Bagchi and N.B.Shroff. ”analysis and evolution of secos, a protocol for energy efficient and secure communication in sensor networks”. In *Ad Hoc Networks J.*, volume 5, pages 360–391, Apr 2007.

[6] A.Silva, M.Martins, B.Rocha, A.Loureiro, L.Ruiz and H.Wang. ”decentralized intrusion detection in wireless sensor networks”. In *Proc. first ACM workshop quality of service and security in wireless and mobile networks*, pages 16–23, 2005.

[7] I Khalil, and S.Bagchi. ”mispar: Mitigating stealthy packet dropping in locally-monitored mulit-hop wireless ad hoc networks”. In *Proc. Fourth ACM SecureComm*, pages 1–10, 2008.

[8] D.Liu, and P.Ning. ”establishing pair-wise keys in distributed sensor networks”. In *Proc. Conf. Computer and Comm. Security (CCS),*, pages 52–61, 2003.